

Estación Meteorológica móvil con posicionamiento GPS

Miguel Ángel Iglesias Jiménez

Grado Ingeniería Informática, especialidad Ingeniería de Computadores

Jordi Bécares Ferrés

24 Enero 2014



Esta obra esta sujeta a una licencia de
[Reconocimiento-No comercial- Sin Obra Derivada 3.0
España de Creative commos](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

A MI PADRES Y PAREJA POR EL APOYO

DADO A LO LARGO DE ESTOS AÑOS QUE HA HECHO

POSIBLE LLEGAR HASTA AQUI.

*Agradecer este proyecto a profesores de la UOC y en especial a
mi consultor en este proyecto, Jordi por su ayuda, paciencia y
respuestas siempre que lo he necesitado.*

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Estación Meteorológica Móvil con indicador de posición por GPS</i>
Nombre del Autor:	<i>Miguel Ángel Iglesias Jiménez</i>
Nombre del consultor:	<i>Jordi Bécares Ferrés</i>
Fecha del libramiento (mm/aaaa):	<i>Enero/2014</i>
Area del trabajo Final:	<i>Diseño Sistemas Embebidos</i>
Titulación:	<i>Grado Ingeniería Informática. (Ingeniería .de computadores)</i>

Resumen

El objetivo principal del presente proyecto es desarrollar una plataforma que permita la medición de parámetros climáticos ambientales característicos de una estación meteorológica tales como presión atmosférica, temperatura, luminosidad, precipitación y viento. Esta tendrá la capacidad de poder integrarse y/o anexionarse para su uso, a distintos tipos de sensores por cualquiera de las plataformas de comunicación que incorpora, Bluetooth, Wifi (IEEE 802.11), Bus I2C, o Ethernet permitiendo así que pueda ser operativa con las diversas redes de meteorológica existentes.

La plataforma descrita tendrá además la posibilidad de movilidad facilitando al mismo tiempo datos de posicionamiento tales como altitud, latitud y longitud. Latitud, mediante GPS y por último y lo que la hace un sistema de mayor interés, tendrá la posibilidad de toma de decisiones y actuaciones externas lo que lo dota de un especial atractivo ya que al mismo tiempo que puede dar un servicio a las citadas redes meteorológicas puede desarrollar las tareas de control en domótica tales como sistema de riego de jardín, actuaciones sobre toldos en función de luz, control de calefacción o incluso de intrusión pues dispone de sensor de proximidad e infrarrojos.

El sistema se compone de una plataforma central que actuara como nodo coordinador al que se pueden conectar nodos sensores encargados de tomar medidas como puede ser precipitación dirección del viento entre otros.

Los nodos sensores externos deberán de tener unas determinadas características como recoger información de sensores y ser capaces de enviarla al nodo central el cual está desarrollado sobre la plataforma STM32L-CORTEX-M3, concretamente LPCXpresso1769.

Así con estos objetivos se efectúa un estudio de diversos sensores en los que principalmente buscamos precisión, bajo consumo y que sean standards a ser posible sin requerimiento de adaptaciones de señal sino que lleven integrados alguno de los sistemas de comunicación que implementa el nodo coordinador. Dada la movilidad que se pretende del sistema se efectúa un estudio para que su consumo sea bajo y se implementa con diversos métodos en alimentación desde la red eléctrica, a batería con placas solares. Así si su montaje implica acción con actuadores externos dispondrá también de sistema de comunicación con el usuario mediante LCDG, esto no se fijó en el objetivo pero sin embargo dados los buses empleados y las posibilidades se ha considerado apropiada su inclusión.

Overview

The main objective of this project is to develop a platform for measuring climatic environment parameters characteristic of a weather station such as atmospheric pressure, temperature, light , precipitation and wind. This will have the ability to integrate and / or annex for use , different types of sensors by either incorporating communication platforms , Bluetooth, Wi-Fi (IEEE 802.11) , I2c Bus or Ethernet , allowing it to be operative with weather various existing networks .

The platform will also have described the possibility of mobility while facilitating positioning data such as altitude , latitude and longitude Latitude, using GPS and last and what makes a system of interest, the possibility of decision making and the external actions that gives it a special appeal and at the same time you can provide a service to those schemes can develop control automation tasks such as garden irrigation system , awnings actions depending on light , heating control or even intrusion because it has proximity sensor and infrared.

The system consists of a central platform to act as coordinator node that can connect sensor nodes responsible for taking steps such as precipitations, and wind direction, etc.

The external sensor nodes must have certain characteristics as gather information sampling sensors and be able to send it to the central node which was developed on the STM32L-CORTEX-M3 platform, specifically LPCXpresso1769..

So these objectives a survey of various sensors in which they primarily seek precision , low standards and are ideally without requiring adaptations signal but with any of the integrated communication systems that implements the coordinator node takes place. Given the mobility of the system is to be performed to study its consumption is low and is implemented with various methods from the mains power , battery with solar panels. So if the action involves mounting external actuators also have communication system with the user by LCDG , this is not fixed on the target but nevertheless given the buses employees and possibilities have been considered appropriate.

Palabras Clave

Microcontroladores, sistemas embebidos o empotrados, sensores, red de sensores sin hilos (WSN), sistemas de tiempo real, LPCXpresso LPC1769, WiFly RN-XV, FreeRTOS

Índice de Contenidos

1. INTRODUCCION.....	12
1.1. Justificación y contexto	12
1.2. Descripción del proyecto	13
1.2.1. Objetivos del sistema	13
1.3. Objetivos del proyecto	14
1.3.1. Lista de los objetivos	14
1.4. Enfoque y método seguido	14
1.5. Planificación del proyecto	15
1.5.1. Planteamiento inicial y seguimiento	15
1.5.2. Metodología de Memoria	15
1.5.3. Metodología de aplicación.....	16
1.6. Recursos empleados	18
1.7. Productos obtenidos	19
1.7.1. Breve descripción módulos	20
1.8. Breve descripción de los capítulos de la memoria	21
2. ANTECEDENTES	21
2.1. Estado del arte	22
2.2. Estudio de Mercado	22
3. DESCRIPCION FUNCIONAL.....	24
3.1. Estación meteorológica móvil con indicador de posicionamiento GPS	24
3.1.1. Diagrama de Bloques de la aplicación	24
3.1.2. Posicionamiento y Comunicación en la Red	24
3.1.3. Interacción entre los diferentes objetos del Sistema	24
3.2. Aplicación control y monitorización (PC / WEB / TouchPanel/Leds-Buzzer)	25
3.2.1. Diseños interfaces usuarios	25
3.2.2. Diagrama de Bloques Aplicación WEB	25
3.3. Diseño aplicación sistema embebido	27
3.3.1. Diagrama de Bloques del Sistema Embebido	25

4. DESCRIPCION DETALLADA HARDWARE.....	34
4.1. Descripción del Sistema	34
4.2. Descripción de Sensores	35
4.2.1. Sensor Medición presión Atmosférica	35
Definición, métodos para su medición y selección del sensor.	
4.2.2. Sensor Medición Temperatura y Humedad Relativa	36
Definición, métodos para su medición, selección del sensor.	
4.2.3. Sensor Medición Iluminación	37
Definición, métodos para su medición y selección del sensor.	
4.2.4. Sensor Medición Precipitaciones.....	36
Definición, métodos para su medición y selección del sensor.	
4.2.5. Sensor Medición dirección y Velocidad del viento.....	38
Definición, métodos para su medición y selección del sensor.	
4.3. Acondicionamiento de señales	40
4.3.1. Pluviómetro.....	40
4.3.2. Sensor Ultrasónico para medir velocidad y dirección del viento.....	40
4.4. Alimentación del sistema	41
5. DESCRIPCION DETALLADA SOFTWARE.....	44
5.1. Sistema Operativo seleccionado para el Sistema Embebido	44
5.2. Entorno de desarrollo de la aplicación	44
5.3. Programación de la aplicación	45
5.4. Desarrollo de la aplicación	46
5.5. Casos de Uso	47
5.6. Diagrama de Flujo aplicación	53
6. VIABILIDAD TECNICA.....	54
6.1. Estudio Viabilidad Proyecto	54
7. CAPÍTULO COSTES	55
7.1. Presupuesto prototipo	55

8. CAPÍTULO 7. CONCLUSIONES	57
8.1. Objetivos alcanzados	57
8.2. Ampliación y opciones de mejora	57
8.3. Impacto ambiental	58
8.4. Autoevaluación	58
9. GLOSARIO.....	59
10. BIBLIOGRAFIA	59
11. ANEXOS.....	60
Anexo A (Estructura de las tramas de datos)	61
Anexo B Manual de Usuario	64
Parámetros Configuración Sistema (Fichero <i>Config_sys.h</i>)	65
Compilación y Grabación de la aplicación	66
Manual Aplicación WEB	69
Anexo C Documentación API Software aplicación	75
Anexo D (esquemas Eléctricos Sistema)	158
Anexo E (<i>datasheet</i> Sensores y Componentes)	168

Índice de Figuras

Figura 1 Diagrama de Gantt	17
Figura 2 Controlador LCDG	20
Figura 3 Ensamblaje LCDG	20
Figura 4 LCDG Ensamblado	20
Figura 5 Placa Sensores	21
Figura 6 Placa Base Sistema Central	21
Figura 7 Placa Salidas	21
Figura 8 Sistema Central Ensamblado	22
Figura 9 Placa simulador nodo sensor externo conectada	22
Figura 10 Placa simulador nodo sensor externo	22
Figura 11 Tabla comparativa CPU,s	23
Figura 12 Estado de las tareas en FreeRTOS	25
Figura 13 Esquema Sistema Embebido FreeRTOS+IO	26
Figura 14 Estación meteorológica WMR200	27
Figura 15 Estación meteorológica WatchDog WD2700	27
Figura 16 Estación meteorológica MatPrakPro	28
Figura 17 Diagrama bloques aplicación	29
Figura 18 Captura WEB Local	30
Figura 19 Captura WEB Aplicativo	30
Figura 20 Diagrama Bloques Interacción Objetos del Sistema	31
Figura 21 Captura comunicación del sistema con PC	32
Figura 22 Interfaz Configuración Red Wifi	33
Figura 23 Interfaz Configuración Estado Sistema	33
Figura 24 Interfaz Localización Estación	33
Figura 25 Captura pantalla inicialización Touch Panel	34
Figura 26 Captura pantalla Menú LCDG	34
Figura 27 Diagrama bloques aplicación WEB	34
Figura 28 Diagrama bloques del Sistema Embebido	35
Figura 29 Placa LPCXpresso	36
Figura 30 Modulo UART-USB CP2102	36

Figura 31 Modulo WiFly RN.-171	36
Figura 32 Modulo Bluetooth H-05	36
Figura 33 Sensor Presión Atmosférica y Tabla Características	37
Figura 34 Sensor Temperatura-Humedad Relativa y Tabla Características	38
Figura 35 Sensor Iluminación-Proximidad y Tabla Características	39
Figura 36 Tabla de Precipitaciones	39
Figura 37 Pluviómetro 7852M-HOBO	40
Figura 38 Tabla Direcciones de Viento	41
Figura 39 Anemómetro WindSonic	41
Figura 40 Medidas consumos Sistema en configuración como Control Domótica	44
Figura 41 Montaje Paneles Solares	44
Figura 42 Circuito Monitor de Tensión	45
Figura 43 Batería Li-Ion Enix Energi	45
Figura 44 Entorno de desarrollo de la Aplicación	46
Figura 45 WorkSpace Programación	47
Figura 46 Diagrama de Ejecución	48
Figura 47 Diagrama Casos de Uso	49
Figura 48 Interfaz caso de Uso valores consigna	50
Figura 49 Interfaz caso de uso Control salidas	50
Figura 50 Interfaz caso de uso histórico por Fecha	50
Figura 51 Interfaz caso de uso listado de muestreos	51
Figura 52 Interfaz caso de uso Scan Punto Acceso (PA)	51
Figura 53 Interfaz caso de uso conexión a P.A.	51
Figura 54 Interfaz caso de uso terminal consola WiFly	51
Figura 55 Interfaz caso de uso Calibrar Touch Panel	52
Figura 56 Interfaz caso de uso Monitorizar Datos GPS	52
Figura 57 Diagrama Flujo de la Aplicación Sistema	53

1.1 Justificación

Debido a diversas razones como pueden ser los cambios climáticos, el incremento de riesgo de incendio en determinadas zonas, etc., es frecuente el interés por realizar estudios meteorológicos, estos son llevados a cabo mediante estaciones meteorológicas, lugares que permiten la observación de fenómenos atmosféricos y donde se instalan determinados sistemas que efectúan medidas de las variables atmosféricas. De otro lado cada día es más utilizado el control domótica tanto en domicilios como fuera de ellos hasta el punto de encontrarnos con las *smart cities* donde en muchos casos las decisiones a tomar son causa-efecto directo de esos fenómenos atmosféricos.

En este contexto el presente proyecto pretende desarrollar un sistema de medidas atmosféricas y control domótica quedando por tanto englobado dentro del marco de los sistemas de domótica pero con algunas peculiaridades que lo hacen diferente a los existentes en el mercado, ya que pretende la conjunción de los dos sectores anteriormente mencionados.

Así el presente proyecto nace de la idea de poder disponer de una estación meteorológica capaz de ofrecer unas prestaciones aceptables para llevar a cabo dichos estudios, pero que a su vez tenga unas características que la diferencien como es la capacidad de movilidad, pudiendo así ser instalada de una forma rápida y sin requerimiento de una infraestructura en zonas donde en determinadas temporadas puedan requerir de un especial control, (p.e. bosques en verano), disponiendo de la capacidad alarmar.

Otra diferencia es dotar al sistema de medios para que pueda ser utilizado como sistema de control en diferentes ámbitos, como puede ser agricultura o control en domótica si la estación se instala en domicilio. Así a la vez que un particular hace su uso como control domótica del sistema este podrá dar ofreciendo los datos atmosféricos de la zona donde se encuentre instalado a través de la red a alguna de las diversas redes existentes de meteorología.

Los dispositivos sensores nos permiten medir variables físicas del entorno, estos cada vez mas evolucionan hacia dispositivos embebidos con un menor tamaño, menor consumo y una mayor capacidad de proceso y mayores prestaciones en comunicación. Por otro lado se avanza hacia los sistemas inteligentes los cuales son capaces de recibir datos de su entorno, tomar decisiones y llevar a cabo actuaciones a través de sus correspondientes actuadores de salidas.

De igual forma cada día existe una mayor conciencia en tener en cuenta nuestro entorno llevándonos a una mejor conservación del mismo y un mejor uso de los recursos, esta tendencia que parece consolidada y en expansión esta llevando a implantar cada vez mas sistemas que permiten conocer esos fenómenos y en función de ellos tomar una serie de decisiones, o prevenciones como pudiera ser en riesgos de incendios, previsiones de temporales, apoyo a la agricultura, etc...

Así, este proyecto permitirá aunque de manera modesta introducirse, explorar y aprender aspectos de estos campos relacionando las redes de sensores, con las comunicaciones y la integración mediante los actuadores externos que llevan a obtener una causa-efecto directa de los fenómenos observados.

1.2 Descripción.

[\[Ir al índice\]](#)

El sistema desarrollado es un sistema embebido con MCU, memoria RAM y Memoria EEprom o Flash, el núcleo principal está basado en tecnología CORTEX-M3 entorno al cual se efectúa la instalación de diversos sensores para la adquisición de medidas consta de diversos interfaces de comunicación dotando al mismo de la posibilidad de incorporar diversos nodos sensores, bien con la finalidad de obtener un mayor número de medidas o bien para incorporar dispositivos sensores con fines de control de circuitos externos como pueden ser: Control de riego en función de luz solar, humedad, control de calefacción, control de subida o bajada de toldos en función de la velocidad de viento etc.

El sistema será capaz de capturar fenómenos atmosféricos concretamente temperatura, presión, luminosidad, humedad, proximidad, dirección-velocidad del viento y pluviosidad, estos últimos mediante un nodo externo independiente inalámbrico. Implementa una monitorización de los diferentes factores atmosféricos y dada su movilidad nos facilitará la localización de la zona donde se están dando dichos factores mediante GPS.

Estos muestreos serán efectuados de forma periódica y los resultados obtenidos enviados a una base de datos mediante una conexión sin hilos a la red Internet, a partir de dichos datos el sistema realiza un registro de aquellos que pudieran ser útiles como estación meteorológica quedando estos disponibles en la red, y por otro lado tomara las actuaciones correspondientes como sistema de control en función de los requerimientos que el usuario predetermine para sus necesidades.

En el alcance de este proyecto quedan excluidos los circuitos a controlar, estos deberán tener causa-efecto mediante los actuadores, así para la verificación de los mimos se simularán dichos circuitos a controlar mediante una batería de diodos Led los cuales simularán el circuito sobre el que debe actuar el sistema central.

De igual forma quedan excluidos nodos anexos al sistema siendo simulado el nodo exterior de velocidad-dirección del viento y pluviómetro para ello se simulara la trama de datos que debe enviar el supuesto sensor externo, verificando así el correcto funcionamiento del sistema.

Se efectúa a la finalización entrega de la correspondiente documentación acompañando a esta memoria, esquemáticos detallados del sistema, Manuales y documentación técnica que queda detallada en anexos (*datasheet* componentes), programación del sistema, y aplicación Web de control del mismo.

1.2.1 Objetivos del sistema.

Las características mínimas que se buscan en el sistema se buscan en este sistema son:

- Bajo consumo para poder ser alimentado mediante baterías o placa solar.
- Dimensiones reducidas para su movilidad
- Alta precisión (dependerá de los sensores)
- Fácil manejo en la programación de sus actuadores (si se le requieren).
- Bajo coste
- Monitorizar parámetros de temperatura, posicionamiento,...
- Flexibilidad, para poder conectar diferentes elementos
- Conexión a Internet vía inalámbrica y/o cableada
- Puertos de comunicación

Se pretende el diseño de un sistema de medida de fenómenos atmosféricos con indicador de localización via GPS y toma de decisiones el cual estará desarrollado en torno a un sistema embebido que se nos ha proporcionado en la asignatura concretamente, el dispositivo MCU LPC1769 - CORTEX-M3, a partir del cual y teniendo en cuenta criterios de costos, despliegues, precisión requerida, calidad y mantenimiento del sistema, utilizando herramientas de las que disponemos y software en su mayoría libre de coste, este debe de presentarnos las siguientes prestaciones.

Sensores y adquisición de datos:

- Lectura de temperaturas (externa e interna)
- Lectura de luminosidad mediante sensor de precisión compensado incorporado al sistema
- Lectura de presión mediante micro sensor de precisión incorporado al sistema
- Lectura de Humedad mediante sensor por hilos incorporado al sistema
- Lectura (Simulación) Velocidad del viento mediante nodo externo
- Lectura (Simulación) Dirección del viento mediante nodo externo
- Lectura (Simulación) Pluviosidad mediante nodo externo
- Lectura información GPS, con datos como Posicionamiento, altitud

Interfaces de comunicación:

- Conexión a Red Inalámbrica Wifi para salida de Red Internet del sistema empujado
- Conexión con dispositivos Bluetooth
- Conexión via USB
- Conexión Ethernet LAN 100/1000

Interfaces con el Usuario y funcionalidades:

- Interfaz Usuario vía LCD (Texto o Grafico)
- Interfaz Usuario vía WEB
- Definición por usuario de valores umbrales
- Visualización de medidas, histórico de medidas, posicionamiento de la estación
- Control y monitorización del estado de los actuadores(salidas).

1.4 Enfoque y metodología.

El presente proyecto esta desarrollado con un enfoque didáctico, se lleva a cabo en diferentes ciclos, los cuales se realizan de forma secuencial si bien en todo momento se ha tenido una visión global lo cual permite poder realizar trabajos de forma transversal cuyos resultados han sido utilizados en ciclos posteriores facilitando la solución requerida.

Así en una primera *fase de búsqueda e investigación*, llevada a cabo mediante las cuatro primeras pruebas de seguimiento (PAC,s), se realiza el estudio de la tecnología y herramientas requeridas tanto en el ámbito del hardware (microcontroladores, sensores, componentes electrónicos y herramientas) como en software (sistemas operativos, lenguajes y entornos de programación). Por ultimo esta fase permite la identificación de los datos a tratar y la determinación de las metodologías utilizadas para la obtención del objetivo final.

Las fases *de montaje, de codificación y de testeo y verificación* son llevadas a cabo de una forma transversal, mediante pruebas empíricas para ir obteniendo los resultados deseados en cada uno de los dispositivos sensores instalados.

Por ultimo la fase *de documentación* es llevada a cabo durante toda la duración del proyecto, desde su inicio recabando documentación e información y efectuando documentos tales como diagramas de bloques, diagramas de flujos, esquemas eléctricos, capturas de señales y pruebas efectuadas, que se aportan como anexos a esta memoria.

Se ha tenido presente la utilización de Código Abierto, así el proyecto se presenta en *Open Source* y de igual forma se beneficia de librerías de código de otros autores el cual ha sido convenientemente modificado conseguir nuestros objetivos.

1.5 Planificación del proyecto

[\[Ir al índice\]](#)

1.5.1 Planteamiento inicial y seguimiento

El proyecto queda dividido en tres fases bien definidas: Fase de estudio, fase de implementación y desarrollo y fase de pruebas y conclusiones. Estas fases en su mayoría son secuenciales aunque manteniendo una vision global del proyecto en ocasión se llevan a cabo de forma transversal interactuando entre ellas.

Fase de estudio.

En esta fase se realiza el estudio de la tecnología necesaria que se utilizara así como de las herramientas de las que haremos uso. Se identifican los dispositivos necesarios para la implementación efectuando la selección de estos para lo que se tiene en cuenta diversas consideraciones como calidad y precisión componentes sensores para que cumplan con el mínimo requerido en las condiciones del proyecto, estandarización de los mismos, y costo. De igual forma se determina el sistema operativo a utilizar, lenguajes de programación y aplicaciones, quedando definida la metodología utilizada presentando todo el itinerario a seguir en una planificación inicial (PAC 5) para la obtención del objetivo final.

Fase de implementación y desarrollo.

Una vez determinada la tecnología a utilizar y seleccionados los componentes que se considerados como adecuados para el cumplimiento de requisitos se inicia la fase de montaje de un prototipo, que si bien se realiza el estudio y un primer esquemático de todo el sistema completo, el montaje no es efectuado en su totalidad en un única vez una única vez, sino que aplicamos metodología modular compaginándola con la fase de codificación del sistema y en su inicio con la parte final de la fase de estudio.

Así se inicia el montaje con el diseño y montaje de la Fuente de alimentación, para dar autonomía al sistema, se anexan los módulos de comunicaciones Wifly, Bluetooth y USB. Para verificar su correcto funcionamiento se aprovecha la fase de estudio de la aplicación consumidor-productor la cual nos permite verificar correctamente el funcionamiento de los módulos de comunicaciones Wifly y USB.

A continuación se pasa a efectuar su correspondiente programación para depuración y control mas exhaustivo de posibles errores por considerar que este es el punto que puede resultar más vulnerable e importante en el proyecto.

Una vez obtenidos los resultados óptimos y realizadas todas las pruebas de comunicaciones, tanto vía USB, vía *Wifi*, y *Bluetooth* con GPS, y conseguido un código de comunicaciones estable con un optimo

control de errores, se han ido anexando al montaje los distintos sensores, efectuando en cada uno de ellos su correspondiente programación.

Fase de pruebas y conclusiones.

En esta fase se llevan a cabo los test de pruebas para verificación de todas las funciones descritas para este proyecto, se incluye en ella una parte de subsanación de posibles errores o desviaciones de los objetivos que puedan darse, asegurando así que las pruebas finales sean correctas.

Se termina con la verificación de que se cumplen todos los requisitos marcados y llevando a cabo una toma de datos para dejar constancia de ello y posterior aportación a la Memoria del proyecto.

La planificación puede verse en el siguiente diagrama de Gantt.

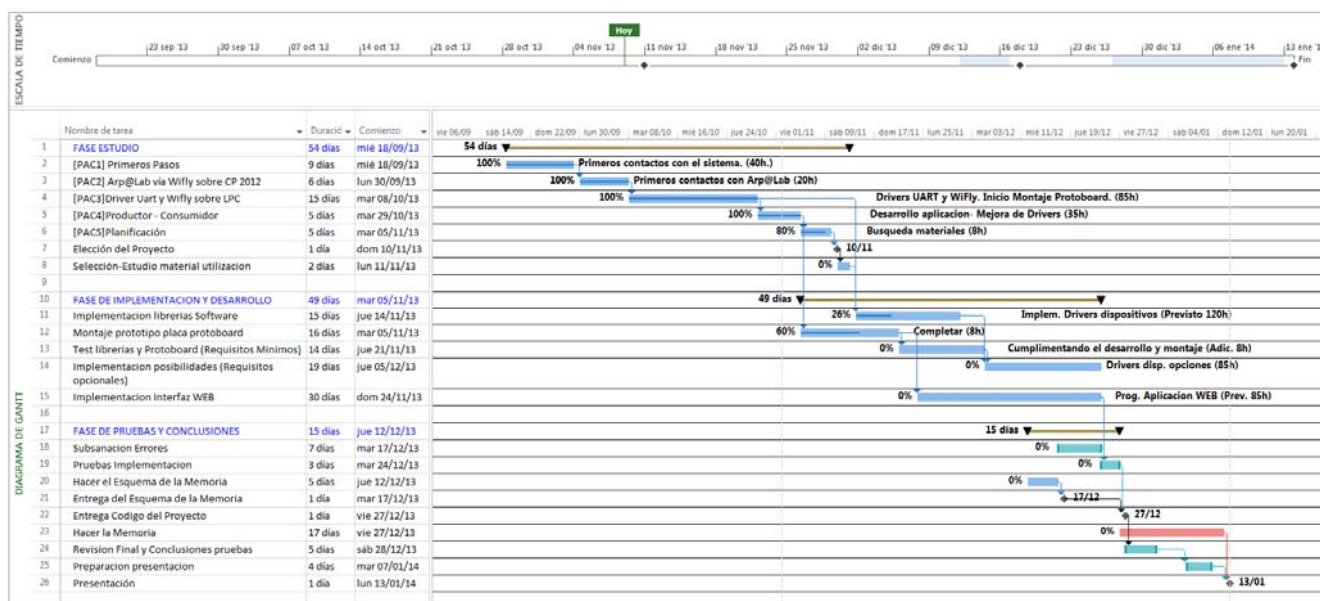


Fig. 1 Diagrama de Gantt

Esta planificación no ha sufrido cambios importantes, **con la salvedad de retraso en el hitos de entrega de programación y documentación final la cual ha tenido un retraso de 10 días**, debido a la carga de trabajo que ha supuesto poder implementar prácticamente en su totalidad todas las opciones consideradas extras que se proponían.

Las dificultades que se han tenido con proveedores, los cuales nos indicaban plazos de entrega que no permitían llevar a cabo la planificación propuesta, se han solventado utilizando otros componentes de características similares subsanando así el problema si necesidad de modificación en la planificación inicial.

1.5.2. Metodología de la memoria

Para la realización de la memoria del proyecto esta se realiza mediante la fase de documentación la cual es llevada a cabo a lo largo de todo el proyecto. Así en la fase de estudio se recopila documentación, *datasheet* de componentes, referencias de bibliografía de igual forma en la fase de desarrollo se capturan tomas de señales, imágenes correspondientes al montaje de prototipo y capturas de pantalla de las pruebas

de verificación efectuadas y en la fase de codificación se implementa la documentación correspondiente a la aplicación del sistema la cual se aporta anexa e incluida en *Workspace* del código fuente.

1.5.3. Metodología de aplicación

La implementación de la aplicación es realizada en cuatro fases que definimos como fase de análisis, fase de programación, fase de codificación, fase de depuración y control de errores.

a) Fase de análisis

En esta fase se efectúa el planteamiento del problema de una forma técnica, se determinan los lenguajes y estructuras mas adecuadas, obteniendo tras el análisis un primer diagrama que aporta la información general del sistema y los dispositivos hardware del sistema que se utilizaran y un diagrama de flujo general de la aplicación.

Se pueden diferenciar dos aplicativos uno para la estación, en adelante sistema central, y otro WEB para usuario.

Aplicación del Sistema Central

En el sistema se requiere que el mismo en su inicialización efectúe un auto chequeo y detección de los recursos disponibles (interfaces con usuario, interfaces de comunicación, sensores del sistema), desactivando los que encuentre con error y enviando un comunicado al usuario de la incidencia. Efectuado este primer chequeo efectúa la conexión de red, conexión con GPS y presentación de resultados obtenidos en dispositivo interfaz con el usuario (si dispone de el ya que este es opcional según el sistema sea instalado como estación meteorológica únicamente o bien sea instalado como sistema de control). Si la inicialización es correcta comienza la recepción de datos de los distintos sensores, adquiridos estos efectúa el envío a la base de datos para su archivo y posterior tratamiento por parte del usuario. Esta labor es periódica y el usuario podrá determinar el intervalo de la misma.

Cuando el sistema no tenga trabajo pasara a estado de bajo consumo, especialmente si se encuentra configurado como estación meteorológica y alimentado por baterías o paneles solares.

Aplicación WEB

Esta aplicación Web monitoriza los datos obtenidos por el sistema central y los presenta al usuario en un formato comprensible. De otro lado realiza las funciones de interfaz de comunicación *usuario-sistema* cuando se requiera configurar parámetros del sistema como puede ser valores umbrales para toma de actuaciones, des-alarmar al sistema si este se ha alarmado, activar o desactivar determinadas salidas si el sistema esta configurador como Control Domótica etc.

b) Fase de programación

En esta fase se instalan los sistemas y entornos de programación a utilizar, se determinan finalmente las estructuras y se realiza un diagrama de flujo detallado de las aplicaciones que será la base para la fase de codificación de las mismas. En este proyecto se utilizara:

- Sistema FreeRTOS para sistemas Embebidos
- Lenguaje HTML
- Lenguaje Java
- Lenguaje php para programación de aplicación WEB
- Lenguaje MySql para tratamiento de base de datos

1.7 Recursos empleados

[\[Ir al índice\]](#)

En el desarrollo del proyecto se requieren diversos dispositivos y componentes electrónicos así como herramientas tanto de electrónica como de informática.

El trabajo se ha desarrollado mediante un ordenador de sobremesa concretamente equipo Intel(R) Core(TM) i7 CPU 939 @ 2.80 GHz con 12Gb de memoria.

Los programas de soporte general utilizados, estos han sido obtenidos bien mediante código abierto o bien a través de las versiones para estudio que nos facilita la propia UOC en su programa Microsoft DreamSpark para uso académico, así se han utilizado:

- **Sistema Operativo Windows 7 64bits**
- **Microsoft Office** (incluyendo FrontPage)
- **Microsoft Project**
- **Visual Studio 12**
- **Microsoft Studio Mx** (para creación animación WEB)
- **MySql** (base de datos)
- **Apache** (para montaje Servidor WEB)
- **DoxyGen** (para creación documentación aplicaciones software):

Los programas específicos para la creación del proyecto han sido:

- **LPCXPRESSO v6.1.0_164**: Entorno de Desarrollo Integral basado en Eclipse, para la programación y grabación de la MCU LPC1769.
- **Tera Term**: Como terminal de comunicación serie (USB).
- **Sistema Operativo Tiempo Real FreeRTOS**
- **CMSISv2p00_LPC17xx**: Librerías Lenguaje prog. C para sistemas embebidos LPC17xx

Los programas específicos para Diseño Electrónico han sido:

- **OrCad Capture**: Para creación de esquemas eléctricos
- **FreeView**: Software Captura Osciloscopio PROMAX OD-571 y Multímetros PROMAX
- **Eagle 5.6**: Para diseño placa PCB

Como componentes electrónicos se indica a continuación los dispositivos principales utilizados, anexando un listado junto con esquemático eléctrico de todos los componentes electrónicos, (resistencias, condensadores, transistores, etc. utilizados) que se omite por su extensión.

- **Placa LPCXpresso1769**
- **Adaptador USB-Serie CP2102**
- **Modulo WiFly RN-171**
- **Modulo Bluetooth H-05**
- **Sensor temperatura TMP 101**
- **Sensor Temperatura-Humedad DTH-22**
- **Sensor Luminosidad-Proximidad VCNL4010**
- **Sensor Presión MPL115A**
- **Acelerómetro MMA7361**

- LCDG Windstar 128x64 con TouchPad
- PCF8575 (Para diseño Control LCDG 128x64 por bus I2C)
- PCF8475 (Como driver controlador Placa de salidas)
- 4 - Paneles Solares
- Placa de salidas 8 Relés 5V/ 10A/250Vcc con opto acopladores
- Buzzer, porta pilas y pila CR2132
- Placas Soldables para montaje protoboard Aristón CEB31

Como herramientas específicas para electrónica:

- Osciloscopio Promax OD-571 (150Mhz)
- Multímetro Mateman 38XR
- LCR Meter PROMAX MZ-505C
- Estación Soldadora-Desoldadora Ersa Icon
- Corta hilos, destornilladores, pela hilos, pinzas sujeción placas

1.7 Productos obtenidos

[\[Ir al índice\]](#)

1.7.1 Breve descripción de los módulos.

En el diseño se ha seguido un criterio modular, por razones de posibles cambios bien por avería en sensores o bien por cambio del tipo de sensor. Así el producto obtenido está compuesto por una placa controladora LCDG (opcional según instalación sistema), placa de Sensores, placa base sistema central, placa de salidas y un Servidor Web con las funcionalidades de monitorización e interfaz usuario-sistema que permitirá la verificación del funcionamiento del sistema a través de la red Internet. Para simulación de los nodos sensores externos se efectúa montaje de una placa simuladora de nodo externo.

La placa controladora LCDG (LCD): Esta compuesta por un LCDG 128x64 con Touch Panel, el cual es controlado por un dispositivo cuyo diseño se basa en el C.I. PCF8575, permitiendo así el control del display mediante bus I2C. El objetivo de esta placa es servir como interfaz de usuario si el sistema es instalado como control domótica, pudiendo ser suprimido si el sistema se instala únicamente como Estación Meteorológica. En las siguientes imágenes puede verse el diseño, montaje de la placa y su adaptación al LCDG 128x64 con TouchPad.

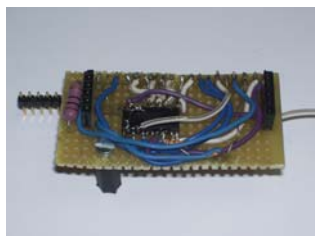


Fig. 2 Controlador LCDG



Fig. 3 Ensamblaje LCDG



Fig. 4 LCDG Ensamblado

Placa Sensores: Esta incorpora un sensor de luminosidad estándar mediante fotorresistencia LDR, sensor de luminosidad y proximidad de precisión VCNL4010, sensor de temperatura TMP101, y sensor de presión MPL115A. El montaje puede observarse en la siguiente figura, esta placa es ensamblada mediante conectores tipo pin Header a la placa base del sistema central.

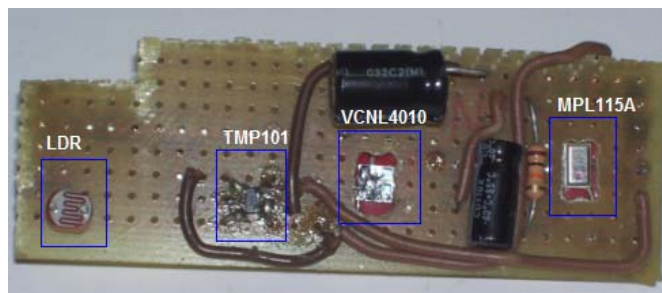


Fig. 5 Placa de Sensores

Placa base Sistema Central: Esta constituye la placa base sobre la que se monta la MCU LPC1769, y los distintos periféricos, incluyendo en ella de forma fija únicamente fuente de alimentación, batería para dispositivo LPC1769, driver de salida PCF8475, y sensor humedad-Temperatura DHT22. El resto de dispositivos descritos anteriormente son instalados en los diversos zócalos realizados con conectores Header como puede verse en la siguiente figura.

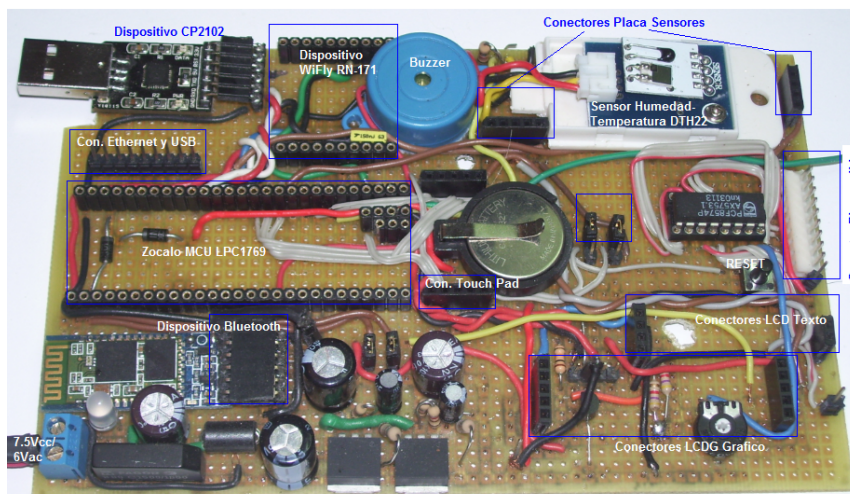


Fig. 6 Placa base Sistema Central

Placa Salidas: Esta es adquirida mediante proveedor y se encarga de efectuar la adaptación de salidas mediante opto acopladores y relés, a los circuitos externos sobre los que actuar.



Fig. 7 Placa de Salidas

En la siguiente figura podemos ver todo el conjunto ensamblado, una vez conectado:

[\[Ir al índice\]](#)



Fig. 8 Sistema Central Ensamblado

Placa simuladora nodo externo: Los nodos externos no están incluidos en el ámbito del proyecto, como se indico en descripción, estos deben de ser capaces de efectuar captura de medidas físicas, para los que estén diseñados tales como: pluviosidad, dirección del viento, velocidad del viento, temperatura etc.. y enviarlos al sistema central. Para la verificación del correcto funcionamiento del sistema se efectúa montaje de un simulador de nodo externo mediante dispositivo CP2102 y modulo Wifly el cual emula el envío de los datos correspondientes a pluviosidad, velocidad del viento y Dirección del viento. Es una vez programado y conectado a cualquier USB se comportara como si de un sensor externo se tratara enviando datos periódicamente al sistema. En la siguiente figuras podemos ver dicho montaje y su conexión a un HUB.

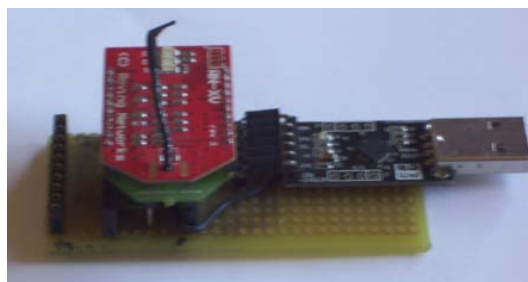


Fig. 9 Simulador Nodo Externo Conectada Fig. 10 Montaje placa simulador Nodo externo

1.8 Breve descripción de los capítulos de la memoria

En los siguientes capitulos se realiza una breve descripción conceptual del sistema diseñado, analizando la situación actual del mercado y diferentes alternativas existentes a la seleccionada para la realización del presente proyecto.

En el capitulo tercero se realiza una descripción funcional de todo el sistema

En el capitulo cuarto se realiza una descripción detallada de las soluciones propuestas y se presenta una justificación de la implementación llevada a cabo.

2.1 Estado del Arte

Un sistema embebido se puede definir como un sistema dedicado, diseñado exclusivamente para la realización de unas tareas específicas y que frecuentemente es un sistema de computación en tiempo real, por lo que son desarrollados para la realización de tareas muy concretas y que por lo general requieren de una respuesta en tiempo real.

Están compuestos por una CPU ó MCU, memoria , buses de datos y puertos de entrada y salida. Suelen incorporar en su estructura dispositivos conversores analógico-digitales (ADC), digital-analógico (DAC), moduladores de pulsos (PWM) ,y diversos interfaces como pueden ser Ethernet, UART, I2C, SSP, CAN, que le permiten toda una interconexión con el mundo real y poder interactuar con dispositivos periféricos como pueden ser sensores, motores, redes, u otros sistemas computacionales de mayor envergadura.

Las características del sistema embebido principalmente están determinadas por el tipo de CPU ó MCU integrada, siendo las principales de estas características: La velocidad de procesamiento, longitud de palabra MCU, cantidad de memoria RAM ó Flash y numero de dispositivos interfaces.

En la actualidad existen gran diversidad de CPU,s, MCU,s en el mercado en la siguiente tabla vemos algunas de las mas utilizadas y sus características principales.

	ATMega1280	MK11DX256VMC5	MSP430F5438	LPC1769	PIC18F8720
CPU	MCU AVR	ARM Cortex M4	MSP RISC	ARM Cortex M3	RISC
Longitud Palabra	8 bit	32bit	16bit	32bit	8bit
RAM	8 Kb	32 Kb	32 Kb	64kb SRAM	3Kb SRAM
ROM	Flash 128Kb EEPROM 4Kb	Flash & Flex Memory 256Kb EEPROM 4Kb	Flash 256Kb	Flash 512Kb	Flash 128Kb EEPROM 1Kb
F.CLOCK	16Mhz	50Mhz	25Mhz	120Mhz	25Mhz
Nº Perif.	SPI : 1 USART: 4 I ² C : 1 ADC: 16	SPI : 2 UART: 4 I ² C : 1 ADC: 16	USCI A : 4 (UART, IrDA, SPI) USCI B: 4 (I ² C, SPI) ADC: 14 ext, 2 int	SPI :4 UART/USART I ² C: 3 ADC: 8 CCP(PWM): 6 Ethernet:1 USB: 1	SPI : 1 USART: 2 (RS232 y RS485) I ² C: 1 CCP(PWM): 5 ADC: 16
Coste	13,61€	7,52€€	9,86€	14,04€	10,75€

Fig. 11 Tabla comparativa CPU,s

Como puede verse existe una extensa oferta disponible. En este proyecto la CPU se nos aporta en la asignatura por lo que no hay razón para justificar la elección, aunque si cabe indicar que se considera una buena elección pues esta se encuentra entre las mas avanzadas ofreciendo grandes posibilidades.

A esta selección de CPU, el sistema tiene que unir las red de sensores sin hilos o WSN (Wireles Sensor Network) la cual esta compuesta, de forma genérica por un conjunto de sensores distribuidos. Cada sensor consta de un sistema empotrado o embebido y de la sensorización específica necesaria para cada proyecto (LOS SANTOS, 2004).

Así la sensorización puede ser parte del sistema embebido o puede acoplarse a través de las correspondientes interfaces analógicas o digitales, siendo en este caso tratadas por el sistema embebido como periféricos.

En este proyecto convivirán ambas, ya que el sistema central incorpora sensores mediante bus I2C, mediante SPI, y ADC, por otro lado las medidas correspondientes a los sensores de pluviosidad, dirección del viento y velocidad del viento son obtenidas mediante nodo secundario el cual debe constar de su propio sistema embebido que le permita el envío y comunicación de los datos recabados al sistema central, dicho nodo no entra en el ámbito de este proyecto por lo que se han simulado dichas tramas de información.

Visto este breve resumen de los componentes de Hardware que conforma un sistema embebido, cabe indicar que este hardware tiene fuertes limitaciones como se ha visto, alguna de estas son tamaño de memoria, capacidad de computo, comunicación con su entorno, consumo de energía etc. de igual forma debe adaptarse a condiciones externas debidas a ubicación, movilidad, condiciones de trabajo, que lo hacen especial. Debido a estas diferencias los Sistemas Operativos (S.O.) para este tipo de hardware presentan también diferencias con respecto a los sistemas convencionales como son:

- Suelen ser de Micro núcleo o modulares haciéndolos fácilmente adaptables a diferentes requerimientos de aplicaciones.
- Generalmente son de diseño con propósito específico, y por tanto más estáticos que un sistema operativo de propósito general.
- En su mayoría estos no disponen de MMU, por lo que no soportan paginación de memoria no utilizando memoria virtual, por tanto no existe protección a nivel de memoria.
- Consolidan un todo con las aplicaciones que ejecuten, es decir por lo general no es posible instalar estos sin ningún tipo de software adicional.
- Mayor fiabilidad y seguridad: Dado que un fallo en un sistema puede hacer que el sistema que controla se comporte de forma peligrosa y/o traiga consecuencias graves, es importante asegurar que si falla lo haga de forma controlada quedando en un estado seguro.
- Mayor eficiencia, ya que deben responder con gran rapidez a cambios en el sistema controlado
- Forma de Interacción con dispositivos físicos: Debido a que la interacción con el entorno suele ser por dispositivos no convencionales como pueden ser ADC,s DAC,s, PWM, I/O series y paralelas, sensores, los componentes del software que controlan estos dispositivos son generalmente dependientes del sistema concreto.
- Mayor Robustez: Dado que no siempre trabajan en condiciones optimas de temperatura, humedad, limpieza, movimientos, etc...

En función de las prioridades y atendiendo a estas diferencias existen por tanto una gran diversidad de Sistemas Operativos (S.O.) especialmente diseñados para sistemas embebidos, algunos de estos son:

- **Windows Embebed CE:** Desarrollado por Microsoft, para CPU,s Intel x86, MIPS y ARM
- **RTLinux:** S.O. en tiempo real ejecutando Linux como un thread (hilo de ejecución).
- **ChorusOs:S.O.** para aplicaciones empotradas, en tiempo real, desarrollado por Sun Microsystems, actualmente bajo código abierto. Altamente escalable y estable. pertenece a la quinta generación.
- **QNX: S.O.** de tiempo real basado en Unix, cumple la norma POSIX, desarrollado por QNX Software System, empresa Canadiense.

- **ulTRON**: Sistema operativo en tiempo real para dispositivos móviles de próxima generación.
- **FreeRTOS**: S.O. de tiempo real para sistemas embebidos pequeños, incluye programas preconfigurados de demostración para diversas arquitecturas como ARM-7, CORTEX-M3, AVR, x86, PIC18, H8S, entre otros.

En este proyecto se hará uso del S.O. FreeRTOS que como se ha indicado es Open Source y que se describe brevemente a continuación.

Filosofía de funcionamiento del S.O. FreeRTOS

FreeRTOS es un sistema operativo en tiempo real Open Source para sistemas embebidos escrito en lenguaje C. Su filosofía consiste en conseguir tiempos reducidos de reacción y tiempos de respuesta predecibles, a la vez que proporciona un acceso rápido al hardware y capacidad de priorización. Para ello el sistema consta de una única imagen binaria, que incluye: el kernel, las aplicaciones y dominio del algoritmo de *Scheduling*.

Parte de la configuración del funcionamiento del S.O se produce en tiempo de compilación así mediante la indicación de los parámetros en el fichero cabecera *FreeRTOSConfig.h* se puede configurar parámetros como los tipos de multitareas, Ticks del sistema (propio o externo), velocidad CPU, N° de prioridades posibles de las tareas, tipos semáforos, tamaño pila, entre otros.

Los diferentes elementos en una aplicación

Los diferentes elementos de una aplicación FreeRTOS son las tareas, estas son la unidad básica de ejecución. Las tareas son gestionadas en tiempo de ejecución mediante el manejador de tareas (*Scheduling*), pudiendo ser instanciadas con un contexto propio e independiente para cada llamada. Con limitación y solo en modo cooperativo, entre instancias de tareas podemos utilizar las denominadas *Corrutinas*, las cuales comparten el mismo contexto entre diferentes instancias.

Por otro lado las tareas no instanciadas no consumen tiempo CPU, ni ocupan espacio en memoria, las tareas en espera si bien no consumen tiempo de CPU si consumen memoria, por lo que hay que tener presente el uso de mecanismos que el propio sistema FreeRTOS proporciona para la sincronización de tareas.

Estas tareas son interrumpidas por IRQ de hardware o cambios de prioridades o contexto de FreeRTOS. Vemos en la siguiente figura los diferentes estados en los que puede encontrarse una tarea.



Fig.12 Estado de las tareas en FreeRTOS

La tareas pueden interactúan entre ellas. Así para sincronizar, proteger y compartir procesos y datos se dispone de semáforos binarios MUTEX y semáforos contadores.

El intercambio de información entre tareas puede hacerse mediante el uso de variables globales con la adecuada sincronización, sin embargo además de este método mediante la API de FreeRTOS proporciona otro método de intercambio de mensajes entre tareas y entre tareas e interrupciones denominado Colas (QUEUES) las cuales tienen elementos de tamaño fijo, pudiendo contener un numero máximo de elementos que se define en el proceso de la creación de la cola.

Existen otros elementos de la API de FreeRTOS como son gestión de sincronización, de captura y liberación de recursos compartidos, colas de mensajes, pipes, gestión de memoria compartida entre tareas, etc..

Por ultimo indicar que FreeRTOS dispone de una capa adicional que hace transparente los detalles de llamadas a periféricos, incluyendo posibles interrupciones que se requieren para su funcionamiento es la denominada FreeRTOS+IO.

A continuación se puede ver el esquema de los que seria el sistema embebido incluida la capa FreeRTOS+IO.

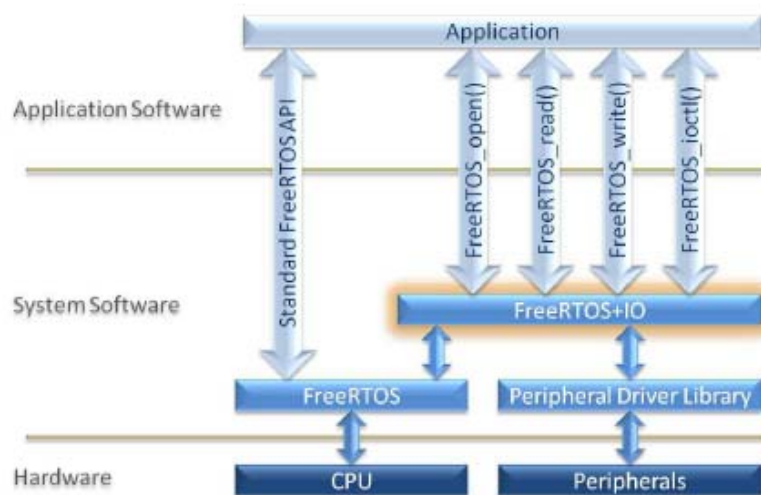


Fig.13 Esquema sistema embebido FreeRTOS+IO

Se ha podido comprobar que existe gran diversidad en el mercado de equipos de estaciones meteorológicas, desde las mas simples hasta las profesionales, de las cuales mostramos a continuación algunas opciones las cuales pueden ser las más parecidas al sistema propuesto en este proyecto.

Si bien cabe indicar que no se ha localizado la conjunción de los dos ámbitos propuestos en este proyecto es decir un sistema que de las prestaciones de estación meteorológica pero que a su vez pueda ser utilizado como control domótica.

Estación meteorológica WMR200

Inalámbrica, captura datos meteorológicos a distancias de hasta 100metros. Indicador de tiempo teledirigido, temperatura interior y exterior, humedad, velocidad del viento y dirección, frialdad del viento, punto de rocío, índice de calor, presión barométrica y datos de precipitación. Permite integración de hasta diez sensores remotos.

En la siguiente imagen puede verse su central y accesorios que incorpora. Costo 950€



Fig. 14 Estación meteorológica WMR200

WatchDog WD 2700 (PCE-IBERICA)

Cinco sensores incluidos, dirección de viento, velocidad de viento, temperatura, humedad relativa del aire, pluviosidad siendo extensible a 6 sensores, logger de datos interno para 8800 valores, puerto RS232 y opcionalmente kit de software. Con un costo de 1295€



Fig. 15 Estación meteorológica WatchDog WD2700

Estación meteorológica profesional base MetPrakPro (SENSOVANT)

[\[Ir al índice\]](#)

Cuenta con un sensor de presión barométrico integrado y una sonda de temperatura/ humedad, sensor de temperatura TMP100, pluviómetro de cazoletas y dos sensores analógicos (sensor nivel de agua, piranometro, sensor de presión, sensor temperatura suelo, etc.) .Coste 2800€

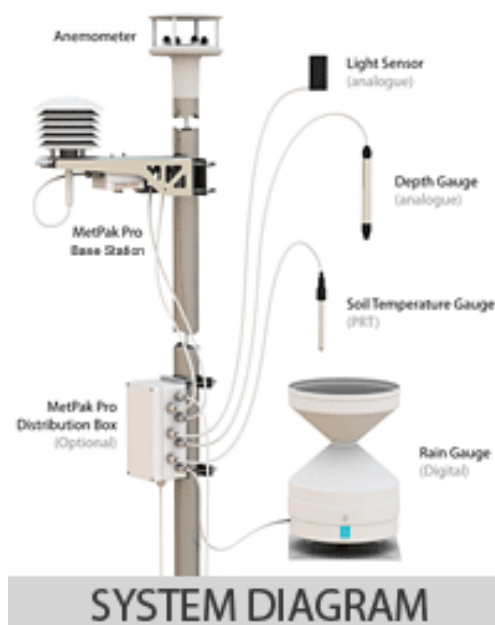


Fig. 16 Estación meteorológica MatPrakPro

Por ultimo indicar que igualmente tampoco se ha localizado ninguna estación que a su vez utilice la información GPS, para dar su localización entre otros datos de interés que puede aportar la red de satélites existente y de uso libre.

3.1 Estación Meteorológica Móvil con indicador de posicionamiento GPS

El sistema esta compuesto por:

- Una Unidad Central la cual incorpora un conjunto de sensores de luminosidad, presión, temperatura, y Humedad, también incorpora los dispositivos interfaces Bluetooth que posibilita la conexión con GPS y el dispositivo Wireles que permite al sistema la conexión a la red Internet de forma inalámbrica, dotando al dispositivo de movilidad. Por ultimo incorpora el conjunto de actuadores de salida.
- Conjunto de Nodos Externos estos componen la red de sensores inalámbricos (WSN), los cuales deben de ser capaces de recabar datos y enviarlos vía inalámbrica a la aplicación. (No incluidos en el ámbito de este proyecto por su extensión), si bien se simulan sus tramas de datos.
- Una aplicación de Monitorización y Control por el usuario, dicha aplicación esta ubicada en servidor WEB remoto.

Su funcionamiento es el siguiente: La MCU solicita los datos al conjunto de sensores incorporado a la unidad central a través de sus correspondientes interfaces, igualmente recibe los datos de los nodos externos vía Internet, efectuando el tratamiento de datos correspondiente toma las decisiones de actuación según los datos recabados, y envía los datos y estado de salidas a la aplicación WEB para su monitorización a la vez que efectúa el registro de los datos correspondientes factores climáticos y de posicionamiento GPS en la base de datos para su posterior tratamiento, si se requiere, por parte del usuario.

Los parámetros de consigna para actuación como sistema de control domótica, son introducidos por el usuario a través del aplicativo WEB o mediante dispositivo TouchPanel instalado en el sistema.

3.1.1. Diagrama de Bloques de la aplicación

En la siguiente figura se represente un diagrama de bloque de la aplicación.

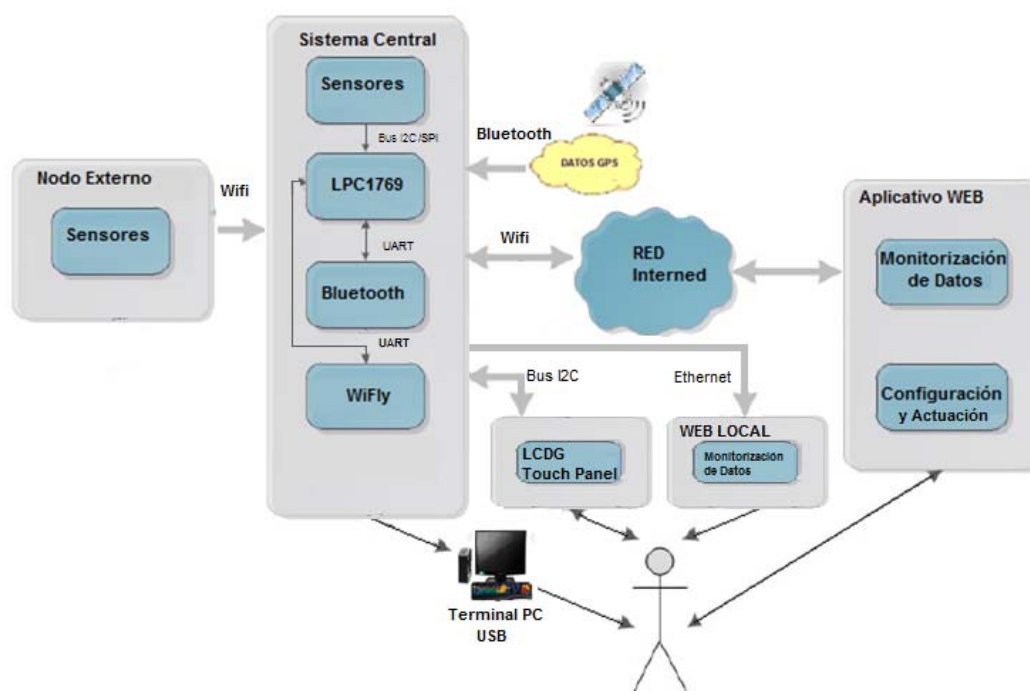


Fig. 17 Diagrama bloque Aplicación

3.1.2. Posicionamiento y comunicación a través de la Red.

[\[Ir al índice\]](#)

El sistema queda posicionado en la Red de dos formas:

- Una de manera independiente mediante la implementación de un servidor WEB (Local) propio, el cual puede ser configurado en sus valores de IP, y Puerto de uso HTTP. A través de este servidor el sistema monitoriza: Los datos atmosféricos obtenidos en cada muestreo que realiza, el posicionamiento GPS de la estación, el estado de las redes de que dispone (*Bluetooth* y *Wifly*), así como el estado de los actuadores de salidas y de alarma del sistema. (En la figura13 se puede ver captura de esta monitorización).
- Otra mediante el uso de un servidor WEB de aplicación(requiere bien un Equipo servidor WEB o bien un hosting en la Red). Este debe ser instalado en la red, a través de este el sistema además de monitorizar los datos anteriormente indicados, permite la interacción del usuario con el sistema. (En la figura 17 puede observarse una captura de la pagina principal de este Servidor WEB).

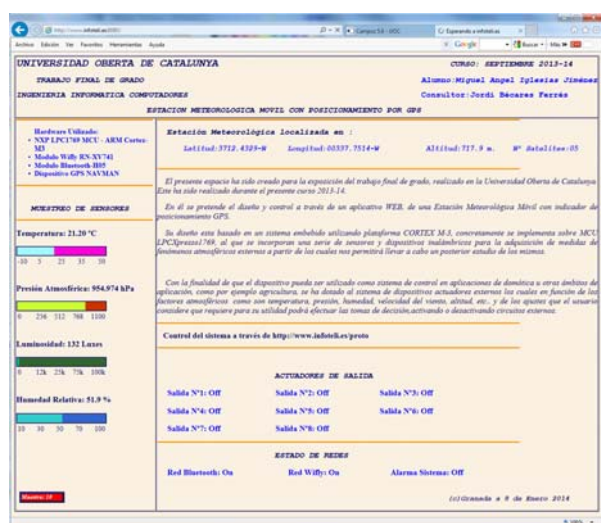


Fig. 18 WEB Local



Fig. 19 Aplicativo WEB

En ambos casos el sistema efectúa el registro de datos a través de la Red, mediante el envío de tramas que se detallan en anexo A.

Estas Web se han posicionado y configurado en las siguientes direcciones:

- La aplicación WEB estará alojada en servidor instalado expresamente para el desarrollo de este proyecto en la URL: <http://infoteli.es/proto>. Incluyendo WEB CAM en tiempo Real.
- El servidor WEB Local se instala en URL: <http://www.infoteli.es:8080>.

Por ultimo indicar que el servidor WEB Local puede ser desactivado si no se requiere su uso como puede ser en el caso de que el sistema se instale como Control Domótica ya que se dispondría del Servidor WEB de aplicación.

3.1.3 Interacción entre los diferentes objetos del sistema.

Como puede verse en el diagrama de bloques aplicación (Fig.12), existe interconexión con diferentes dispositivos y aplicaciones, realizándose estos mediante diversos medios de comunicación así :

- Entre Nodos Sensores externos y sistema se utiliza la comunicación inalámbrica *Wifly*, que a su vez comunica con el sistema mediante UART.

- Entre dispositivo GPS y Sistema se utiliza comunicación *Bluetooth*, cuyo dispositivo *Bluetooth* a su vez utiliza para interactuar con el sistema mediante un puerto UART
- Sistemas de comunicación con Usuario LCDG y TouchPanel hace uso del bus I2C
- Sistema de Monitorización WEB Local utiliza dispositivo LAN Ethernet.
- Sistema comunicación con usuario mediante PC hace uso de dispositivo USB
- Los dispositivos sensores instalados en el sistema efectúan comunicación a su vez por distintos buses esto son:
 - Sensor Temperatura-Humedad DTH22 utiliza BUS SPI (de un único hilo)
 - Sensores de Temperatura TMP101, Presión, y Luminosidad mediante Bus I2C
 - Dispositivo Acelerómetro (opcional) mediante ADC

En la siguiente figura puede verse detallados los distintos interfaces de comunicación de los que hace uso cada dispositivo y aplicativos.

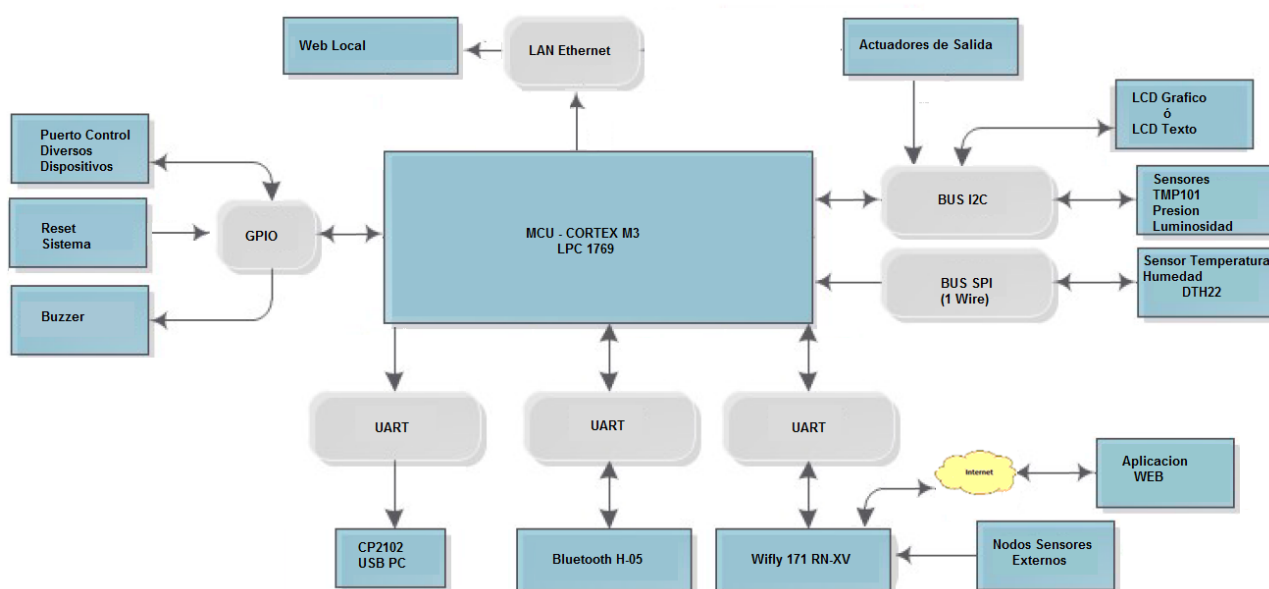


Fig. 20 Diagrama Bloques Interacción Objetos del Sistema

Así el sistema interactúa con cada dispositivo mediante el protocolo correspondiente a cada uno de los buses de interconexión al que se conectan, los cuales han sido programados en las correspondientes librerías como (bus I2C, UART), mediante estas librerías se puede llevar a cabo operaciones de solicitud datos, o enviar actuaciones a cada uno de los dispositivos.

Para la programación de dichas librerías se han seguido los comandos y métodos descritos para cada uno de los dispositivos en su correspondiente hojas técnicas (*Datasheet de componente*), estos se incluyen anexos a esta memoria.(Anexo E.).

3.2 Aplicación Control y Monitorización (PC / WEB / TouchPanel)

[\[Ir al índice\]](#)

El sistema interactúa con el usuario de cuatro formas diferentes: Conexiona a PC por USB, a través de la RED mediante los aplicativos WEB,s descritos anteriormente, de forma directa a través de LCD gráfico y TouchPanel, así como mediante indicadores luminosos (*Led,s*) y avisos sonoros (*buzzer*).

- **Monitorización Mediante PC:** Este se realiza únicamente en dirección sistema-usuario. A través del puerto USB de que dispone, el sistema envía las actuaciones que esta llevando a cabo, así como los parametros diversos de configuración y chequeo en su inicialización o si existe algún error el tipo del mismo y las actuaciones tomadas. Para ello bastara con conectar el sistema mediante su dispositivo USB (CP2102) al PC y mediante una aplicación de comunicaciones serie, como puede ser *TeraTerm* abrir una consola de comunicación con el mismo. En la siguiente figura puede verse una captura de dicha comunicación desde el inicio del sistema hasta que comienza la ejecución de las tareas entrando en un bucle sin fin de la aplicación.



Fig. 21 Comunicación con PC (USB)

- **Monitorización y Control mediante aplicación WEB:** Este se realiza en comunicación bidireccional es decir sistema-usuario y usuario-sistema,. Así a través de la aplicación WEB diseñada el sistema monitoriza históricos de datos recibidos y posicionamiento sobre mapa de la posición de la estación. De igual forma presenta estado de salidas, estado de redes, y estado de alarma del sistema. El usuario interactúa con el sistema mediante las diversas paginas diseñadas en la aplicación WEB entre las que se encuentra: Localización, Estado Sistema, Configuración Red. El usuario puede llevar a cabo la configuración del sistema en los siguientes parámetros:
 - Valores consigna para toma de decisiones tales como limites de temperatura, presión, luminosidad, humedad, etc.
 - Activación / Desactivación de actuadores del sistema.
 - Alarmar al sistema.
 - Intervalos de tiempo entre muestreos y registro de los mismos en base de datos.
 - Configuración parámetros de conexión de la Red *Wifi*.

En la figura 14 mostrada anteriormente podemos ver la interfaz inicial de la aplicación WEB, en las siguientes figuras podemos ver los distintos diseños correspondientes a las interfaces de Configuración Red, Estado del sistema y Localización de la estación.

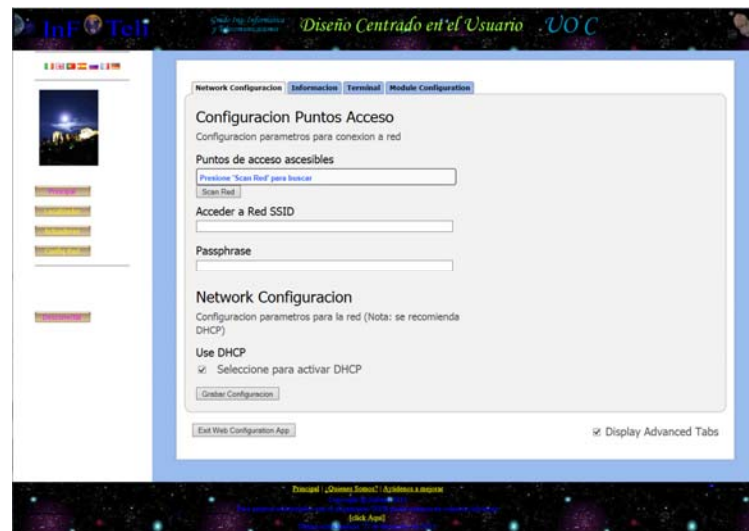


Fig. 22 Interfaz Opción Menú Configuración Red

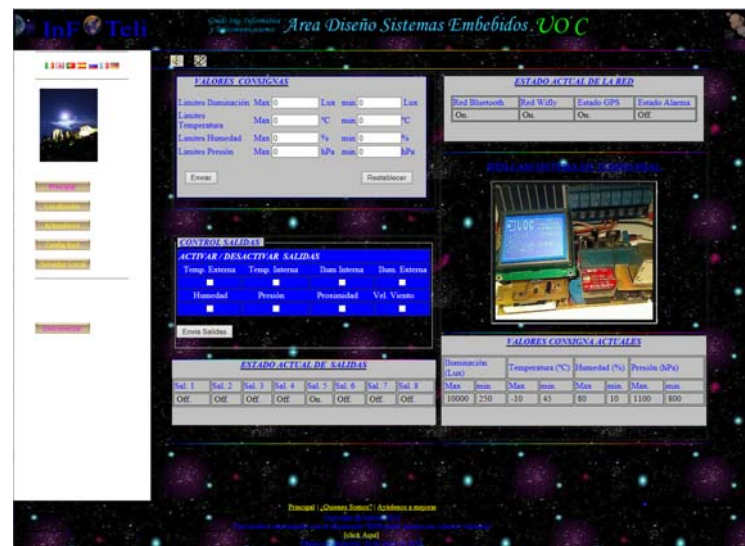


Fig. 23 Interfaz Opción Menú Actuadores

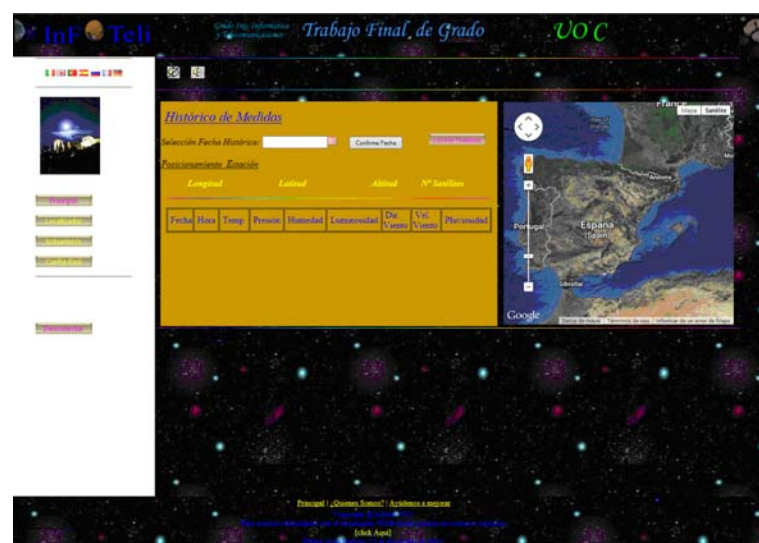


Fig. 24 Interfaz Opción Menú Localización Estación

- **Monitorización y Control mediante TouchPanel y LCD Grafico:** El sistema interactúa con el usuario de forma directa mediante el dispositivo LCD Grafico y a través de su *Touch Panel*. Así mediante el LCDG el dispositivo presenta distinta información como: Estados de redes (mediante representación grafica), fecha, hora del sistema, Coordenadas de posicionamiento de la estación y altitud de la estación, Nº de satélites en uso, etc. En la siguiente captura vemos la pantalla de presentación de inicial y presentación de datos.



Fig. 25 Interfaz inicio y configuración Touch Panel



Fig. 26 Interfaz LCDG

**** (Anotación)** la interfaz de menús para poder llevar a cabo la interacción mediante el TouchPanel únicamente se ha efectuado la programación del drivers TouchPanel para su instalación y calibrado, quedando así verificado su correcto funcionamiento y ajuste, el cual es efectuado en el inicio del sistema y guardado en memoria *Eeprom*. Sin embargo por falta de tiempo no se han podido implementar los diversos menús, como serian teclado para comandos e introducción de valores consigna, menú de actuación sobre salidas etc.. Queda esto por tanto incluido como ampliación y mejoras del sistema.

- **Monitorización por indicaciones Luminosas y Avisos sonoros:** El sistema en todo momento informa al usuario del estado de las salidas mediante los indicadores luminoso (*leds*) de que dispone cada una de las salidas instalados en placa de salidas. Así el diodo correspondiente a una salida si esta se encuentra activa se mostrará encendido y apagado en caso contrario.

Igualmente el sistema esta dotado de un Buzzer el cual envía avisos sonoros, en casos como por ejemplo cuando se requiere en su inicio la atención del LCDG para ajuste del TouchPanel si este no ha sido efectuado antes, o si el sistema esta alarmado emitiendo avisos sonoros de forma intermitente.

A continuación se presenta el diagrama de bloques de la aplicación WEB.

3.2.2 Diagrama de Estructura aplicación WEB

[\[Ir al índice\]](#)

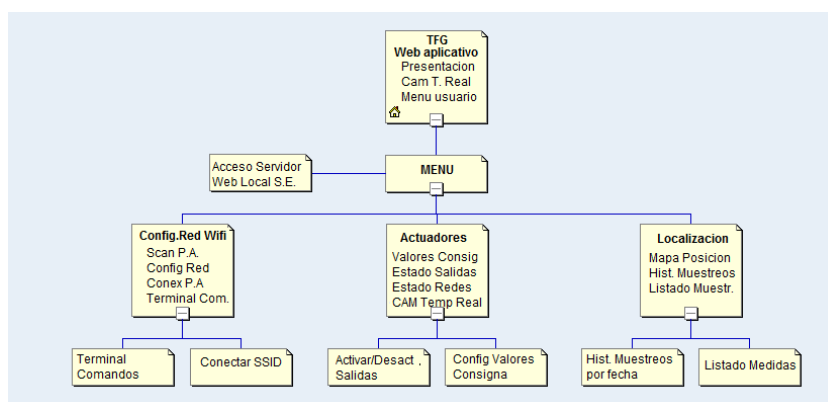


Fig. 27 Diagrama Estructura Aplicación WEB

La aplicación desarrollada en el sistema se ha llevado a cabo de forma totalmente modular. Por un lado se dispone de la librería de controladores de todos los dispositivos hardware implicados (*TFC_LIBRARY*) tanto los incluidos en la propia MCU como pueden ser ADC, Buses, Ethernet, etc.. como los dispositivos externos a esta tales como los sensores, LCDG, Wifly, Bluetooth. Así estos dispositivos pueden interactuar entre ellos a través de su correspondiente *drivers*.

De otro lado se dispone de la aplicación del sistema central, la cual es la encargada de efectuar la inicialización del hardware, la creación de semáforos, colas y tareas dando paso al administrador de estas últimas.

Mediante esta programación modular se consigue hacer el sistema escalable y la reusabilidad del código implementado, a la vez que se elimina la interdependencia entre ellos.

A continuación puede observarse el diagrama de bloques del sistema diseñado.

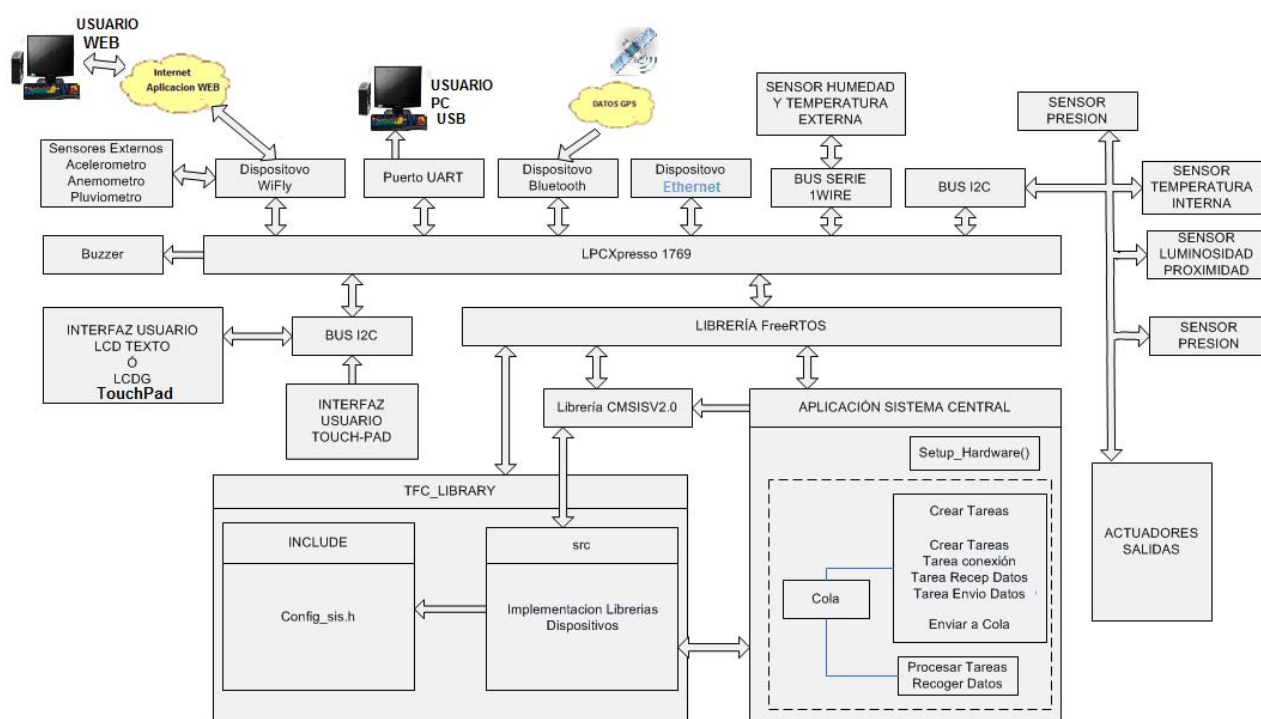


Fig. 28 Diagrama Bloque del Sistema Embebido

4.1 Descripción del sistema

El sistema se ha diseñado realizando una placa a la que se le ha denominado placa base. En esta placa base como hemos podido ver anteriormente (Fig. 6) se han efectuado el montaje de la fuente de alimentación, batería y los distintos zócalos para el ensamblaje de los diferentes componentes hardware que componen finalmente el sistema y que se describen a continuación:

LPCXpresso LPC1769

Esta placa incluye la MCU LPC1769, con los diferentes dispositivos e interfaces descritos anteriormente y cuya información detallada se incluye en anexos de datasheet de componentes.

Este dispositivo va previsto del grabador LPC-link de la MCU. Mediante este grabador y su correspondiente software nos permite la grabación del código de programación en el dispositivo.

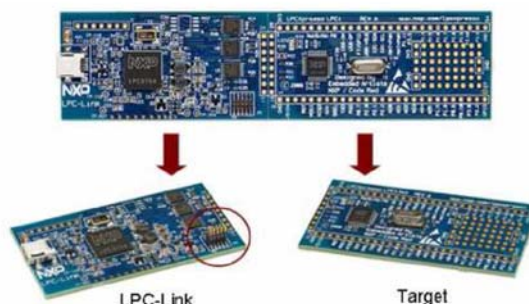


Fig. 29 placa LPCXpresso

Conversor UART-USB

Dispositivo utilizado como interfaz Sistema-PC a través del cual el sistema efectúa la comunicación con el usuario para informar del estado del sistema, así como de los posibles errores que se puedan suceder en el mismo.



Fig. 30 Módulo UART-USB 2102

Módulo Wifly RN-171

Módulo de comunicación inalámbrica utilizando el protocolo IEEE802.11b,g,n dispone de un microcontrolador de 32 bits, su consumo es bajo del orden de los 4uA en modo *sleep*, 38mA en modo activo, dispone de GPIO, y ADC, lo que lo hace especialmente atractivo para uso de sensores externos.



Fig. 31 Módulo Wifly RN-171

Módulo Bluetooth H-05

Módulo de comunicación inalámbrica *Bluetooth* utilizando el protocolo IEEE802.15 este dispositivo puede ser configurado bien como maestro o bien como esclavo con un alcance de 100metros.



Fig. 32 Módulo Bluetooth H-05

4.2 Descripción de sensores

[\[Ir al índice\]](#)

Un sensor, a diferencia de un transductor, esta siempre en contacto con la variable a medir. Así mientras que un transductor es un instrumento que transforma un tipo de energía en otra, el sensor es un dispositivo con capacidad para transformar un fenómeno físico o químico en energía eléctrica.

En función del tipo de sensor se llevara a cabo la medida de los siguientes parámetros: temperatura, humedad relativa, iluminación solar, presión, precipitación y velocidad y dirección del viento.

Las características mínimas que deben de reunir los sensores escogidos para este proyecto son:

- Bajo consumo
- Alta precisión
- Pequeñas dimensiones
- Ser estándar
- Bajo Coste

4.2.1 Sensor Medidor presión atmosférica

La presión se define como la fuerza que se ejerce sobre un área o superficie. Existen cinco tipos de presión: Presión absoluta, presión atmosférica o barométrica, presión diferencia, presión positiva relativa y presión negativa relativa, este proyecto se centra en la presión atmosférica o barométrica.

La presión barométrica es la que ejerce el aire en cualquier punto de la atmósfera, de ahí que se hable de forma general de presión atmosférica terrestre. Esta presión varia ligeramente con las condiciones meteorológicas y disminuye con la altitud.

Existen diversos métodos para su medición como son mecánicos, neumáticos, electromagnéticos y elementos electrónicos. Dentro de estos se conocen cuatro tipos de sensores: Capacitivos, Sensor de Hall, sensor piezoresistivo y sensor monolítico.

Sensor seleccionado: Efectuando algunas comparativas de los sensores vistos en el mercado se ha seleccionado el sensor MPL115A, principalmente por su bus de comunicación I2C, por su bajo consumo, alta precisión y reducido tamaño. El dispositivo esta calibrado en fabrica y se corresponde con el tipo de sensor monolítico.



PRESION	Unidades	min.	Tip.	Max
Resolución	kPa	-	0.15	-
	Bit	16	16	16
Rango	KPa	50	-	115
Conversión Time	ms		1.6	3
VDD	Voltios	2.3	3.3	5.5

Fig. 33 Sensor Presión Atmosférica y Tabla Características

Sensor MPL115A

4.2.2 Sensor Medidor Temperatura y Humedad relativa

[\[Ir al índice\]](#)

La temperatura es una magnitud que hace referencia a la noción de calor o frío. Esta puede expresarse en el S.I de Unidades por grados Kelvin, aunque también fuera del ámbito científico se hace uso de otras escalas como la escala centígrado o Celsius, y la escala Fahrenheit, En este proyecto utilizaremos la escala Centígrada o Celsius. Para llevar a cabo la conversión la ecuación siguiente muestra como realizarla:

$$T(^{\circ}C) = T(^{\circ}K) - 273.16$$

La Humedad Relativa esta definida como el porcentaje de vapor de agua que contiene el aire, este se mide en porcentajes, así a nivel del mar si nos encontramos con un porcentaje del 90% nos esta indicando que el aire contiene ese porcentaje del vapor de agua que puede admitir. En zonas secas puede llegar a ser del 20%. Este factor esta directamente relacionado con la temperatura y nivel de saturación del aire.

Sensor seleccionado: El sensor seleccionado para este proyecto ha sido el SHT 11, sin embargo debido a problemas con proveedores en plazos de entrega, los cuales superaban los plazos para poder cumplir la planificación nos vemos obligados a utilizar uno similar el DHT22.

Ambos son capaces de efectuar medidas de los dos parámetros indicados, temperatura y Humedad relativa, estando calibrados, se trata de un sensor de la familia CMOS el cual posee un converso A/D de 14 bit, bajo consumo de energía y pequeñas dimensiones .

En la siguiente tabla podemos ver sus características principales, tanto para la medición de Humedad Relativa como para la de Temperatura.



Sensor DHT22

HUMEDAD	Unidades	min.	Tip.	Max
Resolución	%HR	0.5	0.03	0.03
	Bit	8	12	12
Repetitividad	%HR	-	± 0.1	-
Rango	%HR	0	-	100
TEMPERATURA				
Resolución	°C	0.04	0.01	0.01
	°F	0.07	0.02	0.02
	Bit	12	14	14
Repetitividad	°C		± 0.1	-
	°F	-	± 0.2	-
Rango	°C	-40	-	123.8
	°F	-40		254.9

Fig 34 Sensor Temperatura Humedad Relativa y Tabla Características

4.2.3 **Sensor Medidor Iluminación y proximidad**

[\[Ir al índice\]](#)

La radiación solar se define como el conjunto de radiaciones electromagnéticas emitidas por el Sol, las cuales para alcanzar la superficie terrestre han de pasar la capa de ozono la cual absorbe la mayor parte de radiación.

Su magnitud de medida es la irradiancia, la cual nos indica la energía por unidad de tiempo y área cuando alcanza la tierra. Su unidad son los W/m^2 aunque existe otra conocida como Lux. Esta es la luminosidad (lumen) por metro cuadrado. Esta será la que se utilizara a lo largo de este proyecto. Aunque no existe conversión directa entre Lux y W/m^2 si existe una equivalencia en función de la longitud de Onda.

Aquí hemos de indicar que en principio seleccionamos dos sensores independientes el S1087, pero que debido a plazos de entrega de proveedores se ha tenido que sustituir por uno mas estándar el cual si fue recibido dentro de los plazos.

Sensor seleccionado: El sensor seleccionado para medición de Iluminación es el VCNL4010, debido a sus reducidas dimensiones, bajo consumo y una buena precisión.



Sensor VCNL4010

ILUMINACION	Unidades	min.	Tip.	Max
Resolución	Luxes	-	0.25	-
	Bit	16	16	16
Rango	%HR	0.25	-	16383
PROXIMIDAD				
Resolución	mm		Up 200mm	
	Bit	16	16	16

Fig. 35 Sensor Iluminación-Proximidad y Tabla Características

4.2.4 **Sensor Medidor Precipitaciones**

La precipitación se define como la caída de hidrometeoros los cuales llegan a la tierra de en forma de lluvia, nieve, granizo, etc. Si esta se presenta en estado liquido se unidad de medida se expresa en milímetros o litros por metro cuadrado (l/m^2). Esta se trata de una unidad de volumen por ejemplo 1mm. de precipitación indica que en una base de $1 m^2$ han caído 1 litro de agua.

En función de la cantidad de lluvia caída la tasa de precipitación (cantidad de lluvia por unidad de tiempo) se clasifica en:

Tipo de Lluvia	Tasa de precipitación en (mm/hora)
Débiles	2
Moderadas	2-15
Fuertes	15-30
Muy Fuertes	30-60
Torrenciales	Superiores a 60

Fig. 36 Tabla de precipitaciones

Los métodos para la medición de la precipitación se pueden clasificar en tres:

[\[Ir al índice\]](#)

- Punto de medición.
- Medida de precipitación por radar.
- Medida de precipitación por Satélite.

Aquí únicamente haremos referencia al primer método, es decir puntos de medición de precipitación ya que será el seleccionado. Este es simple, se trata de un contenedor capaz de recoger la lluvia, nieve o granizo, ahora bien para mejorar las mediciones ya que estas pueden verse afectadas por los efectos del viento existe diversos tipos de puntos de medición: Indicador de precipitación por acumulación, pluviómetro de balancín, y pluviógrafo de sifón.

Sensor seleccionado: De los sensores indicados la selección efectuada sería el pluviómetro de balancín debido a su fácil adaptación y su reducido coste. Además de permitir la evacuación automática del recipiente, este modelo es el más utilizado en la mayoría de las estaciones meteorológicas existentes.



Fig. 37 Pluviómetro 7852M-HOBO

Este sensor en su balancín tiene un imán el cual cada vez que el balancín actúa hace que este interruptor magnético cierre los contactos. Por tanto fijando su salida a nivel alto y conectando a un contador asíncrono podremos conocer la cantidad de pulsos producidos en un determinado tiempo. De otro lado existe una ecuación para el cálculo del volumen de agua ya que las palas del balancín están calibradas, existiendo varios modelos que son 0.1mm, 0.2mm ó 0.25mm de profundidad de pala. Así en función de esta medida característica del tipo de sensor instalado se podrá conocer el volumen de agua en cada pulso siendo este igual a:

$$V = A_c \Delta h$$

$$A_c = \text{Area de pala (mm}^2\text{)} \quad \Delta h = \text{profundidad de la pala en mm}$$

4.2.5 Sensor Medidor dirección y velocidad del viento

[\[Ir al índice\]](#)

El viento se define como un fenómeno que consiste en el movimiento del aire por la atmósfera ocasionado por fenómenos naturales. Estos son debidos a las acciones de la energía solar sobre la superficie y a las diferencias de presión entre las capas atmosféricas provocadas por las diferentes temperaturas de la Tierra.

Existen tres tipos de vientos: Planetarios los cuales circulan por todo el planeta manteniendo su dirección todo el año se clasifican en alisios, contraalisios y circumpolares. Regionales o continentales cambian con paso de los días o con el cambio de estaciones clasificándose en brisas, Monzones, Ciclones y anticiclones. Vientos locales presentan un desplazamiento desde zonas de alta presión a baja presión y se clasifican en Brisa marina y terrestre, Brisa de valle, Brisa de Montaña, Viento catabático y Viento anabático.

La unidad de medida internacional para la medición de su velocidad es Km./h y también Nudos (millas náuticas/hora), así un 1 nudo=1 milla náutica/h=1.852Km/h.

Para medir su dirección esta se lleva a cabo en grados y según los puntos cardinales, norte, sur, este y oeste.

Dirección Viento	Grados(°)	Siglas
Norte	360	N
Norte Nordeste	22.5	NNE
Norte Noroeste	337.5	NNW
Noroeste	315	NW
Sur	180	S
Sur Sudeste	157	SSE
Sur Suroeste	202,5	SW
Este	90	E
Este Nordeste	67.5	ENE
Este Sudeste	112.5	W
Oeste	270	W
Oeste Noroeste	292.5	WNW
Oeste Sudeste	247.5	WSW

Fig. 38 Tabla de Direcciones Vientos

Los dispositivos para la medición de la velocidad del viento son conocidos como anemómetros y para la dirección se les denomina veletas de forma general estos suelen ir en el mismo dispositivo hasta el punto que en la actualidad existen anemómetros capaces de efectuar ambas medidas, los mas estandarizados son: Anemómetros de cazoleta o molinillo, y los Anemómetros ultrasónicos.

Sensor seleccionado: Se han visto diversos tipos de sensores, seleccionando el de Gill Instruments WindSonic, ya que se trata de un sensor de bajo consumo y su voltaje de operación es de 3 Voltios. Por otro lado no tiene partes móviles lo que evita el mantenimiento y un menor desgaste al de los sensores tradicionales.



Fig. 39 Anemómetro WindSonic

Si relación de precio/calidad es relativamente bajo comparado con los sensores tradicionales (anemómetro-veleta). Por otro lado no requiere de calibrado, pudiéndose configurar las necesidades de muestreo y velocidad de envío de datos a través de interfaz RS232.

Otras características son la no existencia de ángulos muertos, estable a bajas temperaturas hasta -35°C, rango de direcciones de 0-360° y velocidad de 0-60m/s

El software que acompaña a este sensor permite realizar un test que verifica el correcto funcionamiento descartando posibles fallos electrónicos.

Una vez vistos los dispositivos sensores que se consideran los apropiados, se pasa a ver el acondicionamiento de señales y la alimentación del sistema.

4.3 Acondicionamiento de señales.

[\[Ir al índice\]](#)

Los sensores de forma general para que puedan interconectarse con el modulo requieren de circuitos que adapten su señal haciendo esta entendible al circuito correspondiente a la MCU donde son acoplados.

En este sentido este proyecto se ha decantado por el uso de los buses de que dispone, por lo que el circuito de acondicionamiento de señales en la mayoría de los sensores queda simplificado a la simple interconexión y a lo sumo la instalación de los correspondientes condensadores de desacoplo de ruido que pueden necesitar en sus circuitos de alimentación, los cuales son recomendados y están indicados sus valores en los datasheet de dichos dispositivos sensores.

Por lo tanto con la salvedad del dispositivo anemómetro y pluviómetro, no han sido requeridos circuitos de acondicionamiento de señal, ya que los sensores seleccionados interactúan con la MCU mediante el bus I2C y el bus SPI (1 hilo).

Indicar que el dispositivo anemómetro y pluviómetro compondrían un nodo sensor externo, lo que aunque su montaje queda fuera del alcance de este proyecto, por cuestiones de tiempo si se ha llevado a cabo su estudio y como debería de efectuarse dicho montaje. Se incluye en anexos los esquemas eléctricos y características de las el nodo externo debe enviar al sistema.

4.3.1 Pluviómetro

Este requiere de un circuito simple compuesto por una un condensador de desacople para impedir los rebotes (100nf) el cual seria instalado con una resistencia en serie de 470KOhmios. Esta salida seria instalada un timer el cual configurado como contador asíncrono nos Daria los pulsos cada vez que el balancín del pluviómetro actúe abriendo y cerrando sus contactos.

4.3.2 Sensor ultrasónico velocidad y dirección del viento.

El sensor *WindSonic* como se ha indicado se comunica mediante el protocolo RS232, por lo que el único acondicionamiento que hay que realizar es respecto a su alimentación ya que este utiliza una tensión de entrada, que aunque en sus características indica 2.5 a 3V observando otras referencias indica que su valor correcto de alimentación es de 5V.

Esto puede efectuarse mediante un conversor DC/DC el cual se incorporaría a la fuente de alimentación del nodo externo el cual es alimentado mediante pack de baterías recargable y panel solar.

Como se ha indicado el montaje de este nodo externo queda fuera del alcance del proyecto, este supone el montaje de un nodo exterior el cual estaría compuesto por el dispositivo sensor pluviómetro, sensor ultrasónico velocidad y dirección del viento, una MCU LPC1769, un modulo Wifi, todo ello con su fuente de alimentación la cual estaría compuesta por un pack de batería recargable y un panel solar.

Se aportan esquemáticos un posible montaje propuesto, asi como las tramas de datos que debe enviar al sistema, las cuales han sido emuladas implementando un pequeño simulador como se ha descrito en el punto 1.7 Productos obtenidos.

El sistema se ha dotado de una alimentación la cual permite al mismo poder llevarla a cabo de dos formas diferentes.

- Mediante la red eléctrica a través de un alimentador con una salida de 7.5Vcc
- Mediante baterías y Panel Solar

En este apartado hay que diferenciar el tipo de instalación y configuración que se haga del sistema, debido a que si este es instalado como sistema de control domótica con la posibilidad de actuación externa el consumo es mucho mas elevado por lo que obligatoriamente deberá disponer de alimentación mediante red eléctrica.

Por el contrario si el sistema es instalado como Estación meteorológica móvil, para recabar datos de factores atmosféricos, no requerirá por tanto del dispositivo de actuadores incluso puede ser suprimido la interfaz grafica haciendo uso de un simple LCD de texto, siendo así los consumos mucho menores y pudiendo el sistema se alimentado mediante batería y panel solar.

Alimentación mediante Red Eléctrica

Esta puede llevarse a cabo mediante un transformador/adaptador AC/DC convencional el cual sea capaz de suministrar un mínimo de 7.5Vcc/1A, recomendable 7.5Vcc/1.5A, o bien mediante un transformador AC reductor 220V/7.5Vac, el circuito esta diseñado para ambas posibilidades.

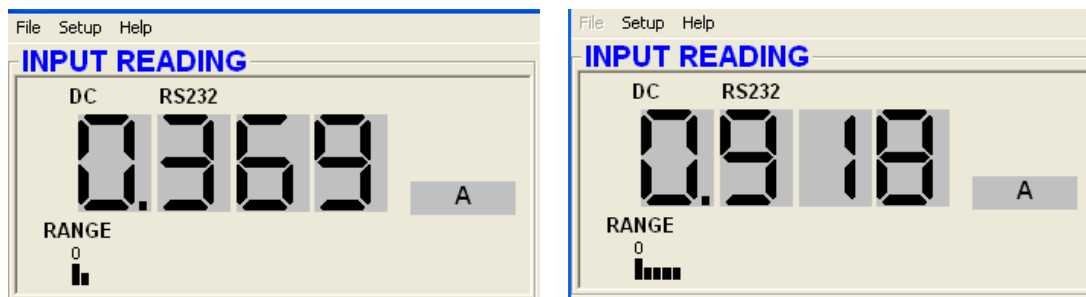
El diseño de la fuente de alimentación se ha llevado a cabo mediante circuitos LM317-T no utilizando los reductores típicos LM78xx por los motivos que se indican a continuación:

- Los reductores 78xx no ofrecen protección contra sobrecargas, por lo que si esta se produce suelen presentar en la salida la totalidad de la tensión de entrada lo que lleva a la destrucción total del circuito que alimentan.
- El consumo de los C.I. LM78xx es mucho mas elevado que el consumo de los reductores LM317 ya que estos pueden ser debidamente ajustados, para llevar a cabo una deducción minima y por tanto un menor consumo.
- El amperaje que admiten es mas bajo (1A) que lo que puede suministrar el LM317 el cual puede llegar a dar 1.5A min. y típico 2.2A (según *datasheet*).
- Mediante el calculo adecuado los C.I. LM317 nos permite ajustar el voltaje que se requiere a nivel de décimas de voltios así podemos obtener los 3.3V exactos que se requieren.

Por lo tanto la fuente de alimentación se lleva a cabo mediante dos reguladores de tensión LM317-T con su correspondiente y adecuado filtraje capaces de suministrar al circuito una corriente de hasta 2Amperios, mas que suficiente incluso en su máximo consumo siendo este cuando:

- Módulos GPS, Wifi, y Ethernet activados y en funcionamiento
- LCGD Activado y monitorizando datos
- Los ocho actuadores de salidas activos.

En esta situación se toman las correspondientes medidas empíricas y vemos que el consumo no llega a 1A. concretamente 918mA, como puede verse en las siguientes tomas de datos donde queda reflejada el consumo mínimo y el consumo máximo del sistema configurado como Sistema de control Domótica.



Consumo con salidas a en Off

Consumo máximo con salidas en ON

Fig. 40 Medidas consumos configuración como Control Domótica

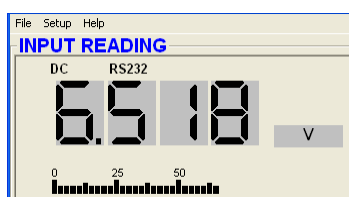
Alimentación mediante batería y Panel Solar

Se realiza un diseño dando la opción al sistema de poder ser alimentado mediante baterías y panel solar, este es utilizable tanto para el nodo externo correspondiente al pluviómetro y anemómetro como para el sistema central si este es instalado únicamente como Estación Meteorológica con posicionamiento.

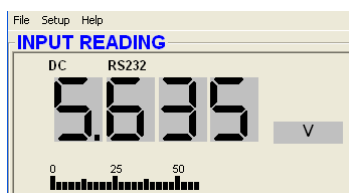
Para llevar a cabo el diseño de este circuito se requiere de un monitor de tensión el cual detecta cuando la tensión de alimentación esta por debajo de 6V, valor umbral por debajo del cual no se asegura una correcta salida de la estabilización de salidas en la fuente de alimentación.

Para ello se utilizara el circuito integrado de Maxim ICL7665 el cual es el encargado de que cuando la tensión baje de los 6.0 V dejar de alimentar el hardware para pasar a efectuar una carga rápida de batería. Por otro lado las baterías no se mantienen en una carga permanente debido a que ello acortaría su vida de uso considerablemente.

El panel solar seleccionado es el MSX-005F este panel solar en cantidad de cuatro paneles solares de este tipo efectuamos un acople de los mismos el cual nos proporcionara con una iluminación minima de aproximadamente 1200Lux, un voltaje de 8Vcc con un suministro de corriente de 450mA.



Nivel de voltaje obtenido a 1300Lux



Nivel de voltaje obtenido a 400Lux

Características Panel MSX-005F

Potencia Máxima: 0,8W

Voltaje máxima: 4.8V

Corriente máxima: 230mA

Dimensiones:

Largo:140mm

Ancho 100mm

Grosor:4.9mm

Fig. 41 Montaje Paneles Solares

En la siguiente figura podemos ver el circuito monitor de tensión, extraído de los esquemas los cuales se adjuntan en anexos.

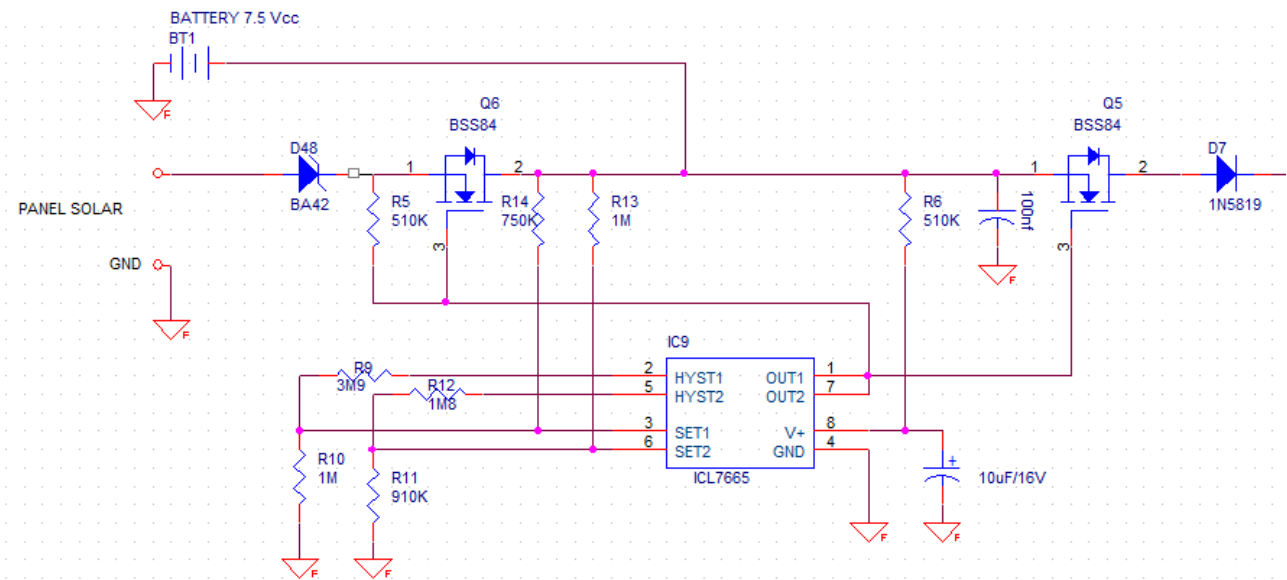


Fig.42 Circuito monitor de tensión

El circuito ICL7665 como puede observarse en la figura, efectúa el control mediante dos ciclos de histéresis sobre los transistores PMOS los cuales actúan como conmutadores, en este circuito se pueden dar cuatro estados según Q5 y Q6 estén en estado abierto o cerrado los cuales describimos a continuación en la siguiente tabla.

Q6	Q5	Comportamiento circuito
0	0	No se da nunca este estado
0	1	Batería por encima de la tensión umbral, el circuito se alimenta únicamente de baterías
1	0	Panel alimentando baterías
1	1	Alimentación por parte de baterías y de panel.

Por ultimo se presenta a continuación la batería propuesta para la alimentación del sistema de la que se indican algunas de sus características.



Fig. 43 Batería Li-Ion Enix Energi

Battery LI-ION 7.5V, 2.2Ah Pack

Capacidad 2.2 Ah

Voltaje nominal: 7.5V

Tecnología Lithium-Ion

Dimensiones: 70mm x 37.5mm x 19mm

Peso: 100gr.

Corriente Maxima de carga 3A.

Terminales: Hilo

5.1 Sistema Operativo para el sistema embebido

Para que el Hardware descrito sea operativo se requiere de un sistema operativo el cual permita al sistema entenderse con el mundo exterior y recabar información a partir de la cual podrá efectuar toma de decisiones y en función de ellas llevar a cabo determinadas actuaciones. De aquí que se requiera de un sistema operativo en Tiempo Real ya que una de las características de este tipo de S.O. es la capacidad de respuesta ante eventos internos o externos.

Como se ha explicado en antecedentes (Capítulo 2), el sistema operativo que se utilizara en este proyecto es FreeRTOS, por ser Open Source y además aportarnos código de aplicaciones reutilizable para este proyecto.

5.2 Entorno de desarrollo de la aplicación

[\[Ir al índice\]](#)

El entorno de desarrollo utilizado recibe el nombre de *LPCXpresso IDE*, el cual es desarrollado por *CodeRed* y *NXP*, esta basado en Eclipse, siendo su aspecto y manejo similar. Este IDE se instala sin problema alguno con la simple descarga de un ejecutable desde la WEB de CodeRed, debiendo efectuar el registro e identificación en dicha pagina WEB.

El IDE se basa en los conceptos de *WorkSpace* y proyecto a partir de los cuales se puede realizar la ordenación de la programación. Los proyectos pueden a su vez ser bibliotecas estáticas o aplicaciones ejecutables, contienen archivos de código fuente, encabezados y cualquier tipo de archivos necesarios. Se puede efectuar la compilación de cualquier proyecto pero únicamente serán las aplicaciones ejecutables, aquellas que incluyen el procedimiento *main()*.

El espacio de trabajo o *Workspace* es un contenedor de proyectos, bien sean aplicaciones o bibliotecas, incluyendo a su vez las configuraciones de entorno, esto permite efectuar copia o mover el contenido de una maquina a otra .

Por ultimo indicar que este entorno nos proporciona un herramienta de gran utilidad que nos permite interactuar mediante el JTAG y el Target pudiendo llevar a cabo depuración y grabación de código en el dispositivo MCU.

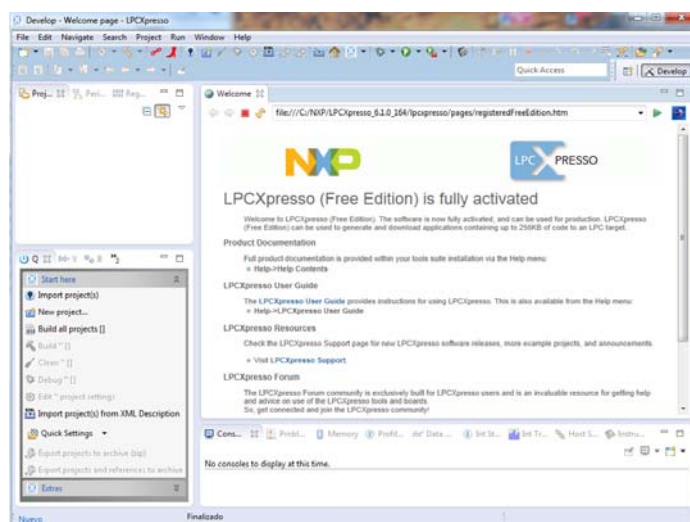


Fig. 44 Entorno de desarrollo de la aplicación

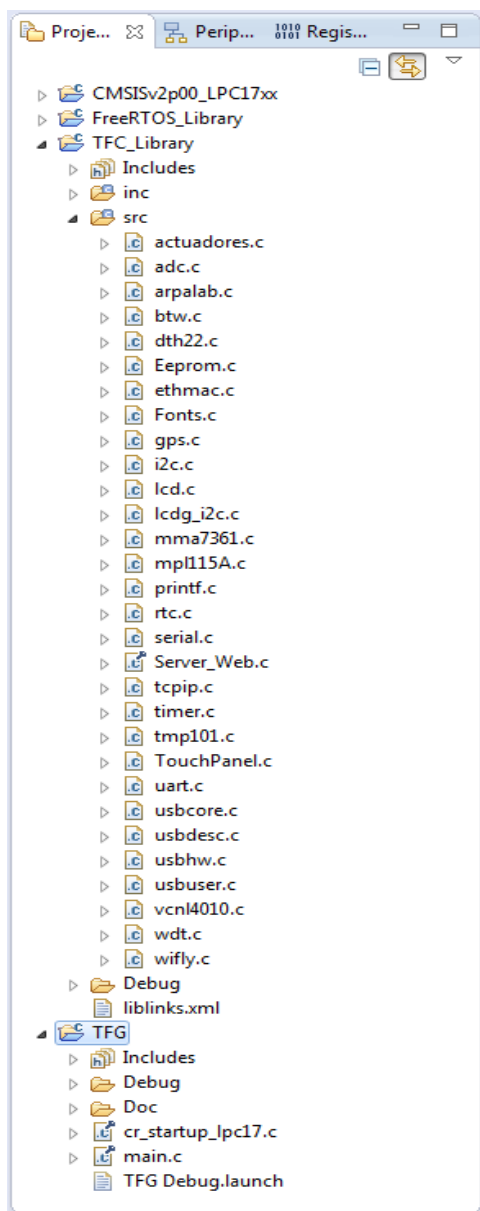
En primer lugar hay que indicar que nos ayudamos para llevar a cabo la programación de mucho código que ya esta implementado y que se recibe de la pagina *CodeRed* junto con el IDE anteriormente indicado. A partir de este código Open Source se lleva a cabo su modificación y se efectúa la creación de la librería de la que se hará uso en este proyecto para la programación de la aplicación.

Así el entorno de trabajo creado (*Workspace*) queda finalmente organizado de la siguiente manera:

- Por un lado los proyectos de sistema (librerías de *FreeRTOS* y *CMSISv2p00_LPC17xx*).
- Por otro el proyecto Librería de Drivers de la aplicación (*TFC_Library*) y el proyecto TFG que es la aplicación en si misma.

Programación Librería Drivers aplicación (*TFC_Library*).

Una vez que se ha instalado el entorno y disponiendo las librerías de los sistemas indicados anteriormente, se lleva a cabo la creación de un proyecto librería el cual hemos denominado *TFC_Library*,



este incluye todo el código implementado específicamente para este proyecto, incluyendo todos los ficheros fuentes y cabeceras (.c y .h) así como ficheros de imágenes, fuentes tipos de letra etc. que pudieran requerirse por la aplicación final que se desarrolla.

En la figura de la izquierda puede verse a modo de breve resumen los ficheros fuentes (.c) de los dispositivos utilizados por el sistema tales como: *actuadores*, *ADC*, *Bluetooth*, *Wifly*, *GPS*, *UART*, *timer*, etc. y otros de aplicaciones como *Arpalab* para envío de datos mediante HTTP, o *Server_Web* para monitorización de datos mediante el servidor WEB local del que se dispondrá en la propia placa MCU del sistema, independientemente del aplicativo WEB.

Programación aplicación TFG

Una vez que disponemos la librería que permitirá al sistema interactuar con los diversos dispositivos se lleva a cabo la creación de un proyecto de aplicación *FreeRTOS*, el cual se ha denominado (TFG). En este como puede observarse en la imagen Fig.(45) contiene el fichero ejecutable *main()*.

Este fichero contiene el código con las tareas y código específico de la aplicación que se desarrolla y implementara las distintas funcionalidades de la aplicación. Las cuales pasamos a explicar mas detalladamente su desarrollo, mostrando su diagrama de ejecución y los casos de uso a continuación.

Fig.45 WorkSpace Programación.

El desarrollo y la implementación de la aplicación se lleva a cabo mediante la programación de las librerías indicadas anteriormente llevándose a cabo en lenguaje de programación C, igualmente se codifica un determinado numero de tareas (concretamente cinco tareas) que junto con otros procedimientos de apoyo a estas tareas, y el uso de componentes del sistema, como son colas, semaforos, etc, permiten llevar a cabo la interacción entre los dispositivos del sistema y llevar a cabo las funcionalidades solicitadas a la aplicación, en definitiva, efectuar los muestreos de los dispositivos, recibir datos del GPS, enviar y recibir datos de la aplicación WEB, monitorizar los a través del servidor WEB Local, y efectuar el control y activación sobre las salidas, avisando mediante alarma si existe determinados datos fuera de los limites. Esta codificación es la incluida en fichero *main.c*.

La programación como puede desprenderse de lo indicado en el anterior punto, ha sido llevada a cabo de una forma totalmente modular.

Por ultimo indicar que el sistema incluye un fichero cabecera (*Config_sys.h*) el cual tiene unos fines especiales de configuración de la aplicación. En este fichero ,antes de llevar a cabo la compilación del programa, se pueden configurar parámetros de la aplicación tales como: IP del servidor Local, direcciones de la aplicación WEB, Activar/desactivar el servidor WEB Local, Tiempos de espera, etc.

Toda la codificación, así como las estructuras utilizadas en la programación tanto de librerías como de la aplicación (tareas y procedimientos) puede verse detalladamente en Anexo B que se adjunta a esta memoria, por lo que a continuación pasamos a ver el diagrama de ejecución y los casos de uso de la aplicación.

[Diagrama de Ejecución Aplicacion \(main.c\)](#)

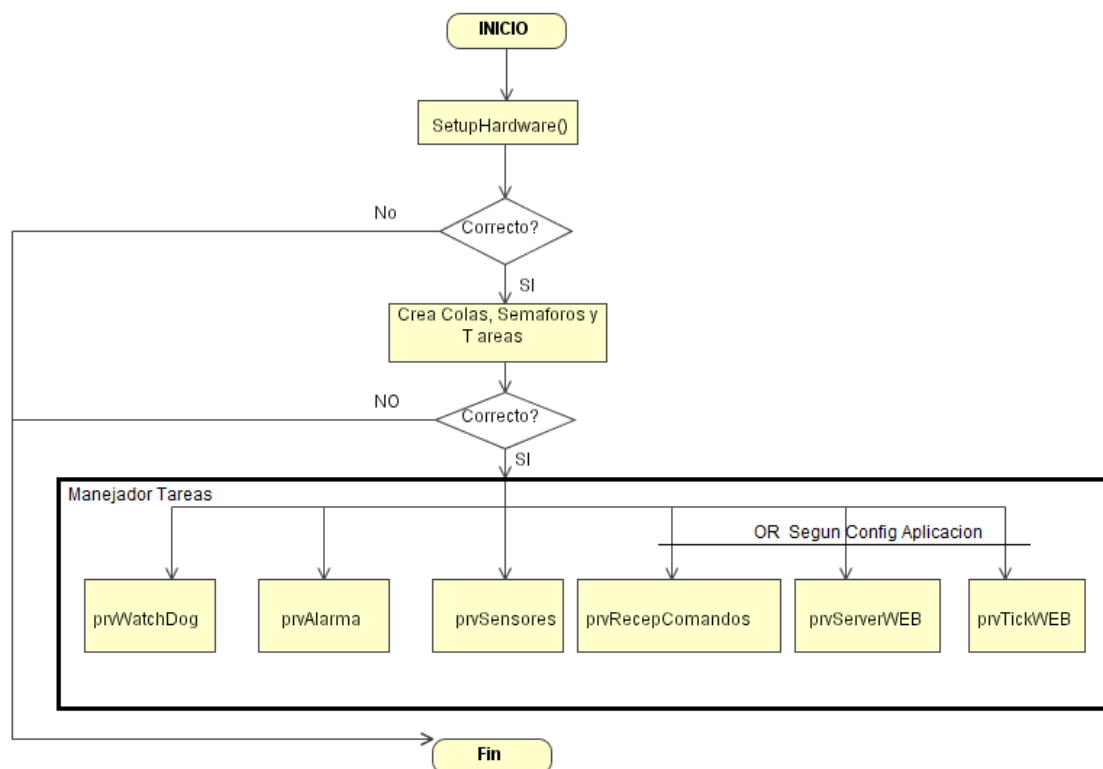


Fig. 46 Diagrama de Ejecución Aplicación

En la aplicación del sistema pueden diferenciarse dos tres de usuarios, que hemos denominado:

- Usuario-Local: Interactúa con el sistema mediante LCDG con Touch Panel
- Usuario-PC: Interactúa con el sistema mediante PC
- Usuario-WEB: Interactúa con el sistema mediante aplicación WEB a través de Red Internet.

Para cada tipo de usuario podemos distinguir los siguientes casos de uso que se representan en el siguiente diagrama.

[Diagrama de Casos de Uso](#)

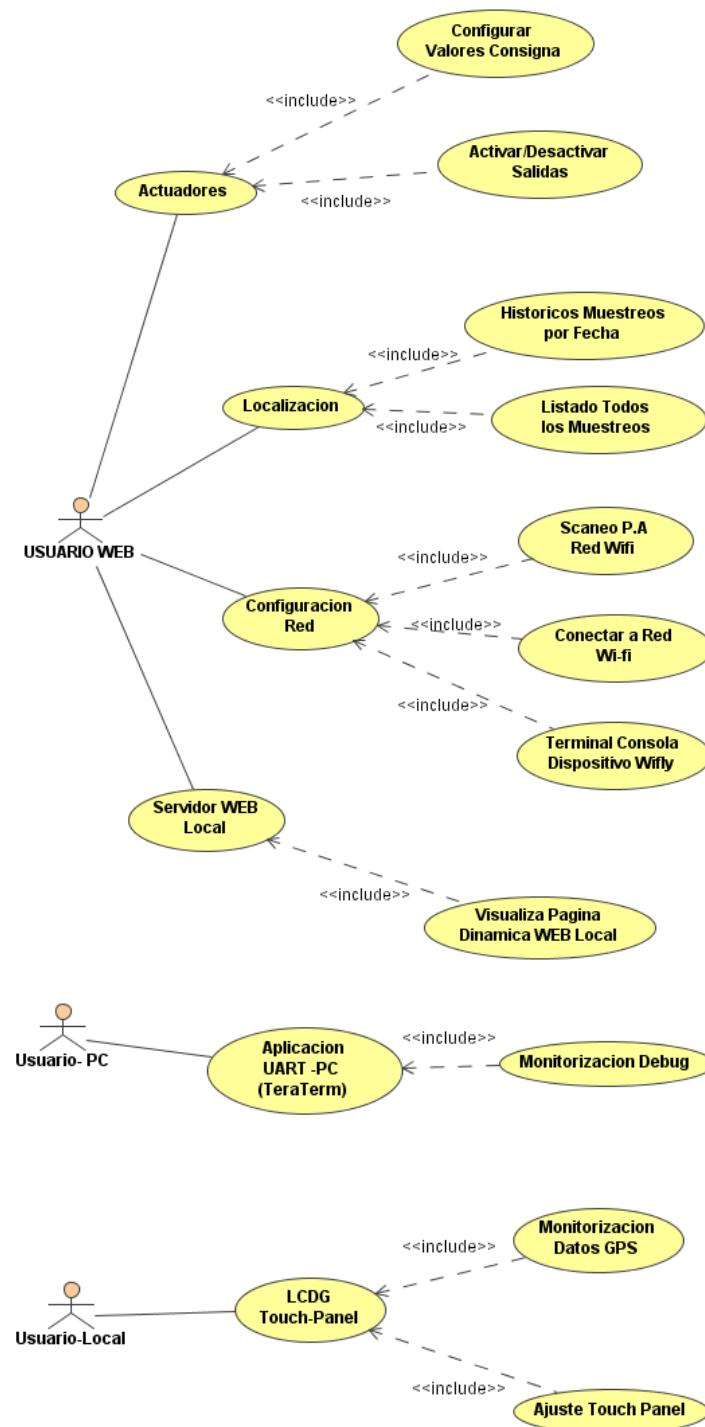


Fig.47 Diagrama de Casos de uso.

Actuadores:

- Configurar valores consigna: Permite al usuario la configuración de los valores consigna a partir de los cuales el sistema tomara la decisión de activar o desactivar la salida asociada a cada uno de las medidas, temperatura, iluminación, presión, humedad. Tras introducir los datos el sistema, el usuario hara click sobre "**Enviar**" el sistema le solicitara confirmación, la cual si acepta los nuevos datos será grabados en el sistema como nuevos valores consignas caso contrario se mantendrán los existentes.

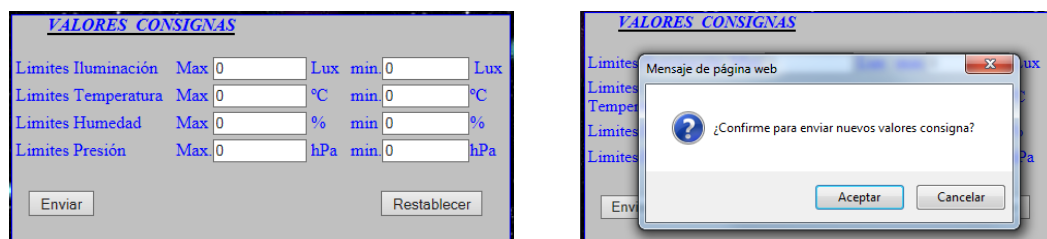


Fig. 48 Interfaz Caso uso valores consignas

- Activar/Desactivar salidas: El usuario activa o desactiva de forma directa los actuadores de salida. Lo realizara en todo el conjunto de salidas (grupo de 8 casillas, una por actuador de salida) el usuario debera indicar en su totalidad el estado que se desean adquieran las salidas. Si el usuario marca una casilla el sistema entenderá que la salida correspondiente debe ser activada si la deja sin marcar se entenderá que debe ser desactivada. Seleccionado el estado el usuario envía la orden haciendo Click sobre "**Envía Salidas**". El sistema solicita confirmación y si es aceptada por el usuario la orden será enviada caso contrario será cancelada.



Fig. 49 Interfaz Caso uso Control Salidas

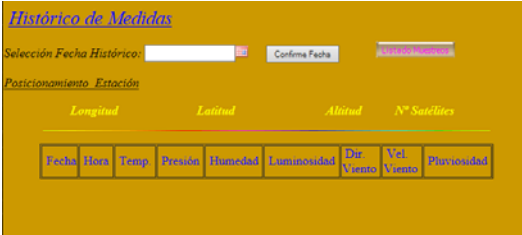
Localización:

- Históricos por Fechas: El usuario solicita se presenten los datos de muestreos correspondientes a la fecha indicada. Para ello el usuario indica la fecha haciendo click sobre el **icono calendario** existente para dicho propósito seleccionando la fecha en el calendario emergente, a continuación hará click en "**Confirmar Fecha**", si existen datos correspondientes a esa fecha estos serán mostrados, indicando en su encabezado la localización GPS de la estación en el momento que se hizo el muestreo, y su representación sobre el mapa. sino existen datos no mostrara ninguna acción.



Fig. 50 Interfaz Caso uso Histórico por fecha

- Listado de Muestreos: El usuario solicita un listado de todos los muestreos existente en la base de datos, haciendo click sobre "**Listado Muestreos**", a lo cual el sistema presenta una listado en pagina WEB de todos los datos existentes.

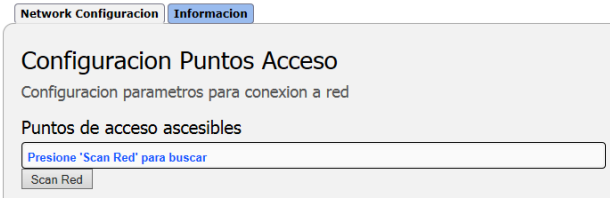


Núm. Muestra	Fecha	Hora	Temperatura	Presión	Humedad	Luminosidad	Dir. Viento	Vel. Viento	PLuviosidad	N. Satélites	Latitud	OCi	Longitud	OCg	Altitud
1819	21-01-2014	23:07:57	21.9	957.685	51.9	3351	0	0	0	4	3712.4174 N	0	337.7494 W	0	722.8
1818	21-01-2014	22:48:37	21.8	957.685	52.3	3259	0	0	0	5	3712.4213 N	0	337.7539 W	0	732.4
1817	21-01-2014	22:37:17	21.8	957.685	52.4	2921	0	0	0	0	337.7539 W	0	337.7539 W	0	732.4
1816	21-01-2014	22:16:52	21.7	959.04	52.4	2988	0	0	0	6	3712.4211 N	0	337.7471 W	0	745.4
1815	21-01-2014	22:01:56	21.7	959.04	52.5	2944	0	0	0	6	3712.4205 N	0	337.7469 W	0	726.7
1814	21-01-2014	21:45:56	21.7	959.406	51.6	3026	0	0	0	22	3712.4196 N	0	337.7502 W	0	743.8
1813	21-01-2014	21:36:12	21.7	960.028	52.3	3393	0	0	0	5	3712.4208 N	0	337.7482 W	0	743.8
1812	21-01-2014	20:06:58	21.5	958.674	52.8	3270	0	0	0	5	3712.4322 N	0	337.749 W	0	727.7
1811	21-01-2014	19:55:14	21.5	958.307	52.7	3054	0	0	0	0	337.7534 W	0	337.7534 W	0	750
1810	21-01-2014	15:37:01	21.1	959.294	52.6	4820	0	0	0	4	3712.4279 N	0	337.7838 W	0	817.9

Fig. 51 Interfaz Caso Listado de Muestreos

Configuración Red:

- Escaneo P.A Red Wifi: El usuario haciendo click sobre "**Scan Red**" solicita al sistema listado de puntos de acceso disponibles para el sistema a través de su dispositivo WiFly. El sistema, si puede conectar con el dispositivo y este esta activo, mostrara listado de los puntos de acceso Wifi disponibles en su alcance.



Network Configuración | **Información**

Configuracion Puntos Acceso

Configuracion parametros para conexcion a red

Puntos de acceso ascesibles

[Presione 'Scan Red' para buscar](#)

Fig. 52 Interfaz Caso uso Scaneo P.A.

- Conexión a P.A: El usuario indicando el SSID y la clave de acceso a la red, haciendo click sobre "**Grabar configuración**", envía al sistema el comando de grabación de los parámetros de la red indicados, configurando el dispositivo para que en su arranque efectúe conexión con la red indicada por el usuario.



Acceder a Red SSID

Passphrase

Network Configuración

Configuracion parametros para la red (Nota: se recomienda DHCP)

Use DHCP

☒ Seleccione para activar DHCP

Fig. 53 Interfaz Caso uso Conexión a P.A.

- Terminal Consola Wifly: El usuario solicitara listado de comandos disponibles para interactuar con el modulo Wifly del sistema. Si indica el comando "**help**", mostrando los comandos disponibles.



Network Configuración | **Información**

Wifly Terminal

Terminal para envio de comandos a dispositivo Wifly

Nota: Tecla 'help' para mostrar opciones de comandos

Comando:

Fig. 54 Interfaz Caso uso Terminal Consola Wifly

Servidor Web Local:

[\[Ir al índice\]](#)

- Visualiza Pagina Dinámica Servidor WEB Local: El usuario haciendo click sobre el botón "**Servidor local**" del menú principal de la aplicación WEB solicita conexión a la pagina dinámica del Servidor WEB local propio del sistema embebido. El sistema responderá mostrando el marco principal la pagina monitorizando medidas en tiempo real, estado de salidas, estado de redes, y posicionamiento de la estación y nº de satélites en uso.

Casos de USO Usuario-PC

- Monitorización Debug: El usuario conecta al sistema mediante PC a través de USB, haciendo uso de programa de comunicación serie como puede ser TERA TERM. Efectuando conexión el sistema responderá enviando a dicho terminal, el estado del sistema en cada momento y el resultado de las operaciones que este llevando a cabo. Puede verse su interfaz en la Fig 19 pag. 29 de esta memoria.

Casos de USO Usuario-Local

- Calibrar Touch-Panel: El usuario hace click sobre la opción **"Calibrar Touch-Panel"** del sub-menu configuración del sistema, y solicita al sistema el ajuste y calibrado del Touch-Panel. El sistema responde mostrando los puntos extremos y solicitando al usuario que haga click sobre ellos guardando los correspondientes valores en memoria Eeprom, quedando el Touch-Panel calibrado y listo para su uso. Esta opción se auto-ejecuta si al iniciar el sistema detecta que existe Touch Panel instalado y no se ha llevado a cabo la calibración del mismo.



Fig. 55 Interfaz Caso de uso Calibrar Touch Panel

- Monitorización datos GPS: El usuario hace click en la opción **"Datos GPS"** del menú principal, el sistema responderá presentando en pantalla las coordenadas de la estación, el numero de satélites en uso y la altitud de la estación, y fecha hora de la adicción de los datos.



Fig. 56 Interfaz Caso de uso Monitorizar Datos GPS

****Anotación:** Estos casos de uso y su programación se ha realizado únicamente para demostración de la posibilidad de una interfaz LCDG con Touch-Panel, debiendo estar incluidos como opciones en sub-menús, y menú principal de una aplicación menú para interfaz LCDG - Touch Panel (incluido en mejoras).

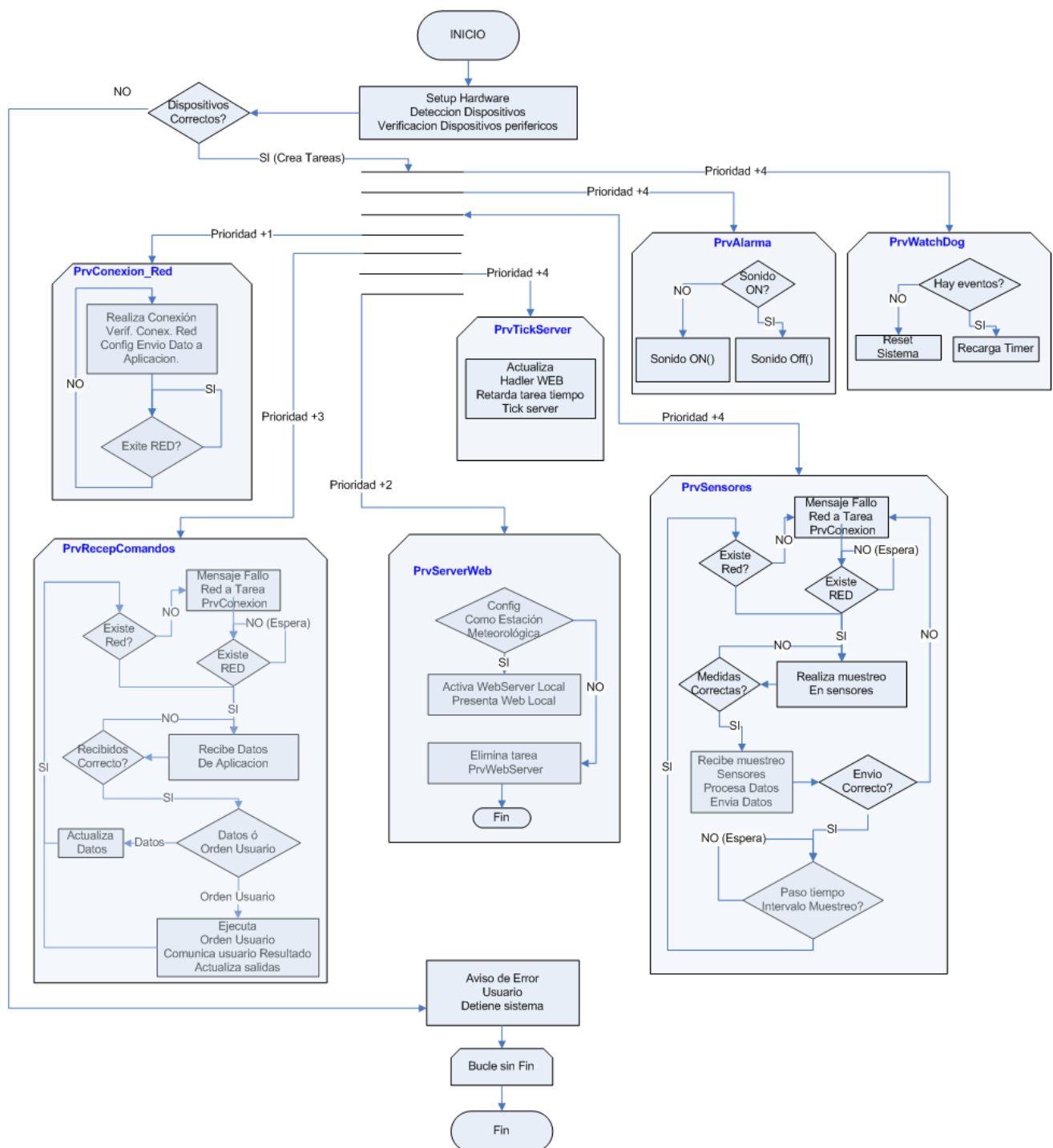


Fig. 57 Diagrama de flujo Aplicación del Sistema

6.1 Estudio Viabilidad Técnica

El proyecto como puede comprobarse esta falta de madurez para poder ser considerado desde un punto de vista comercial totalmente viable, dado que no se han implementado los diseños de placa, ni los diseños de caja para su ensamblado, al igual que la programación de interfaces de usuario se han realizado de una forma simple a modo de demostración, debido al corto tiempo disponible para su realización.

Sin embargo, desde el punto de vista técnico, si puede considerarse un proyecto viable, ya que en la selección de sus componentes se ha tenido presente siempre la consecución de unos fines profesionales, teniendo en cuenta la calidad y precisión de los componentes. Por otro lado resuelve ampliamente el problema planteado, obteniendo un producto con capacidad para la captación de fenómenos atmosféricos, del registro de estos en base de datos para su tratamiento posterior, así como capacidad para tomar decisiones como causa-efecto de esos datos recabados sobre dispositivos externos dando por tanto solución al problema que se planteaba en este proyecto.

Como puntos débiles, remarcar que el principal punto débil es su falta de madurez, a partir aquí se pueden observar otros como son:

- El no estar dotado de una total independencia, ya que el sistema depende de la existencia de una red bien sea Bluetooth, Ethernet o Wifi.
- El dispositivo GPS es externo lo que hace que requiera un dispositivo adicional que por otro lado no suele ser de un costo bajo.
- El alto nivel de competencia en ambos sectores que pretende cubrir, ya que existen una amplia gama de productos tanto en el sector de medición de factores meteorológicos como en el sector de domótica.

Como puntos fuertes se puede considerar los siguiente puntos:

- Uso de un microcontrolador que tiene una potencia considerable, una amplia escalabilidad y con un tecnología que puede considerarse de las punteras en sistemas embebidos.
- La implementación de los distintos tipos de intercomunicación con otros dispositivos, no dejando al sistema limitado por el uso exclusivo de ninguno de ellos, esto lo hace fácilmente escalable y aplicable en diversos ámbitos tanto en la domótica, como en el sector industrial o agrario.
- La implementación de distintos interfaces con el usuario, permitiendo de un lado el control del sistema tanto de forma local como remota, y por otro que el sistema pueda informar en todo momento de su estado.
- La alta precisión de los sensores empleados estando todos ellos en un margen de error por debajo del 0.2% lo hacen un dispositivo de alta precisión.
- Disponibilidad del sistema de diversos tipos de alimentación opcionales, hace que el sistema este dotado de una alto grado de autonomía, pudiendo ser utilizado en sitios donde no exista suministro eléctrico a través de sus paneles solares.
- La posibilidad de anexar nodos externos de una forma facil.

Por lo que se puede concluir que: Implementando en profundidad su aplicación WEB y de control Local para que permita una correcta actuación sobre el sistema y sobre red WSN, cosa que no se ha podido llevar a cabo por falta de tiempo, se puede considerar que el proyecto es técnicamente viable.

7. COSTES

[\[Ir al índice\]](#)

6.1 Presupuesto Prototipo

7PRESUPUESTO INVESTIGACION DISEÑO Y MONTAJE PROTOTIPO

Descripción Componentes	Cantidad	Precio Unidad(€)	Precio Total(€)
Placas Circuito impreso protoboard Soldable Ariston CEB31	2	6.5	13.00
Rollo Estaño sin plomo 100g	1	8.68	8.68
Condensadores Cerámicos multicapas varios valores	28	0.16	4.48
Condensadores Electrolíticos varios valores	13	0.25	3.25
Resistencias varios valores 1/4 Watio	40	0.10	4.00
Resistencia array 9RAS10K	1	0.45	0.45
Resistencia Ajustable 10K	1	0.35	0.35
Diodos 1N5819	7	0.23	1.61
Diodos 1N4448	4	0.04	0.16
Diodos BA41	1	0.50	0.50
Diodo Led Bicolor	1	0.60	0.60
Puente Rectificador B80 C1500/1000	1	1.10	1.10
Reguladores LM317T	2	0.70	1.40
Filtros F.A. VK200	4	0.69	2.67
Transistor CMOS BSS84	2	0.35	0.75
Transistor BC546	1	0.10	0.10
Transistor BC548	1	0.10	0.10
C.I. ICL7665ACSA	1	6.50	6.50
Circuito integrado PCF8574P	2	1.94	3.88
Circuito integrado PCF8575	1	3.75	3.75
Bateria 7.5Vcc 2.2Ah	1	38.17	38.17
Bateria CR 2032	1	0.55	0.55
Paneles Solares MSX-005F (0.8W 4V)	4	7.48	29.92
Buzzer con oscilador incorporado	1	2.5	2.5
Conector Phoenix-2	1		
Conector compacto LAN-USB	1	3.20€	3.20€
Regletas Header Hembra (regletas 25 pins)	4	1.20	4.80
Regletas Header Macho (regletas 25 pins)	4	1.20	4.80
Zócalo DIP 16	1	0.2	0.20
Zócalo Batería Botón PP2035	1	0.76	0.76
LPCXPresso LPC1769	1	14.04	14.04
CP2102	2	9.78	19.56
WiFly RN-171	2	24.65	49.3
Modulo Bluetooth H-05	1	5.75	5.75
Placa 8 Salidas Opto acoplada	1	10.00	10.00
LCD Grafico con Touch Panel	1	25.00€	25.00
Modulo Circuito Electrónico Driver Touch-Panel	1	9.20	9.20
Sensor TMP 101	1	3.04	3.04
Sensor SMD Barómetro MPL115A	1	2.09	2.09
Sensor SMD Iluminación - Proximidad- Infrarojo VCNL4010	1	2.30	2.30
Sensor Temperatura-Humedad DTH-22	1	6.20	6.20
Total Coste Materiales			289.71 €
Horas Investigación, desarrollo, montaje y programación	480h.	15.00	6.750.00 €
COSTE BASE			7039.71 €
IVA 21%			1478.33 €
COSTE TOTAL			8518.04 €

Aunque queda fuera del alcance del proyecto con la finalidad de tener idea aproximada del coste total que supondría poder llevar a cabo la producción en serie del sistema central propuesto, se han efectuado consulta sobre el costo para la realización de las placas de circuito impreso y fotolitos para posteriores tiradas en serie, al igual que para la fabricación de un molde para posterior inyección en plástico de lo que sería la fabricación en serie de las cajas de ensamblaje del sistema central, esto son datos aproximados en función de las medidas, ya que los diseños no se han realizado.

Para ello se ha consultado con la empresa **NovaTeck (Barcelona)** especialista en la realización en serie de circuitos impresos y con **Moldes Barcelona** para conocer el coste aproximado de la preparación del citado molde.

PRESUPUESTO CIRCUITOS IMPRESOS Y MOLDE CAJA ENSAMBLADO.

Descripción	Cantidad	Precio Unidad(€)	Precio Total(€)
Circuitos Impresos que incluyen			
Placa base principal (primeras 10 unidades)	10	10,90	109.00
Placa sensores (primeras 10 unidades)	10	5.40	54.00
Placa F.A Paneles (primeras 10 unidades)	10	4.76	47.60
Gastos preparación	305.18	305.18	305.18
Fotolitos	150.17	150.17	150.17
Molde para inyección Plástico			
Fabricación Molde para inyección en Plástico	2750.00	2750.00	2750
TOTAL			3415.95 €

PRESUPUESTO FINAL PROTOTIPO PARA FABRICACION SERIE

Descripción	Cantidad	Precio Unidad(€)	Precio Total(€)
Circuitos Impresos para fabricación en serie	1	1	665.95
Molde para inyección caja ensamblaje sistema central	1	1	2750.00
Investigación, diseño montaje y programación prototipo	1	1	7039.71
COSTE BASE			10455.66 €
IVA 21%			2195.68 €
TOTAL COSTE			12652.34€

8.1 Objetivos alcanzados

El desarrollo del proyecto ha resultado una tarea bastante complicada debido al gran numero de horas que supone desarrollarlo con detalle y como contrapartida el poco tiempo disponible para la realización.

Hay que indicar que ha resultado un reto muy interesante y me ha permitido aprender bastante sobre el tema. La dificultad del tema escogido, y la falta de tiempo debida principalmente a compaginar el proyecto con otras tareas y estudios, no me han permitido llevar el proyecto a puntos de realización que me hubiera gustado vistas las grandes posibilidades que ofrece los dispositivos de que se han podido disponer.

Indicar que básicamente estoy satisfecho con el producto final obtenido el cual ha presentado un funcionamiento correcto consiguiendo alcanzar los objetivos marcados, si bien hay que reconocer que falta puntos importantes para alcanzar cierto nivel de madurez como diseño de placa, diseño de caja exterior, una programación de interfaces mas exhaustiva que alcance de alcance al usuario a todas las posibilidades del sistema etc.. y realizar algunas mejoras de las cuales se proponen las principales vistas.

8.2 Propuesta de mejoras

A lo largo del desarrollo del proyecto y en la presente memoria se han indicado varios puntos como mejoras del sistema tanto en software como en hardware, así se propondrían las siguientes mejoras:

Software:

- Programación adecuada y eficaz del modo bajo consumo (no efectuado por falta de tiempo)
- Programación de un menú de Interfaz Usuario-sistema mediante LCGD - Touch Panel
- Programación de una Interfaz Usuario-sistema WEB mucho mas detallada y avanzada.
- Programación de Interfaz USB que incluye la propia MCU (solo se ha efectuado Drivers).

Hardware:

- Inclusión de un modulo GPS al sistema pudiendo dejar así el dispositivo *Bluetooth* para comunicaciones con dispositivos domótica cercanos, y no dependiendo el sistema de GPS externo.
- Inclusión de zócalo tarjeta *microSD*, lo que podría dotar al sistema de una total autonomía para registro de datos y poder realizar un servidor WEB Local mas avanzado.
- Dotar al sistema de conexión 3G dando un total independencia al mismo.
- Eliminación Modulo CP2102, ya que el sistema dispone de USB propio.

Estas son mejoras que básicamente en Hardware no incrementan el costo de un forma excesiva, ya que el dispositivo GPS interno es mas económico que un GPS externo por *Bluetooth* (se ha efectuado así por que se disponía del externo), se ha consultado un modulo que incluye GPS y 3G para Arduino el cual su costo es de 105€. Por otro lado se ahorraría costo al eliminar el dispositivo CP2102 utilizando el USB incorporado en la MCU, y los costos de zócalo tarjeta *microSD* son mínimos (aprox. 10€).

8.3 Impacto ambiental

[\[Ir al índice\]](#)

En el presente proyecto se ha tenido en cuenta a la hora de desarrollarlo el consumo energético de ahí que se prevea que el sistema y los nodos puedan ser alimentados con baterías de litio, las cuales contaminan menos y mediante energía renovable como es la energía solar.

Las plataformas de hardware utilizadas se ha cuidado que cumplan con la normativa RoHS la cual restringe el uso de materiales peligrosos como son mercurio, plomo, cadmio, cromo VI, etc..

8.4 Autoevaluación.

Considerando que los requisitos mínimos básicamente consistían en:

- Adquisición de datos mediante sensores
- Programación de interfaces de comunicación
- Muestreo periódico de datos y envío a la aplicación local
- Conexión a Internet mediante Wifi

Creo que se han cumplido los objetivos y se han incorporando bastantes de las mejoras que se dieron como optativas en la planificación inicial.

Así este TFG me ha brindado la oportunidad de hacer una breve introducción en el mundo de los sistemas embebidos así como de las redes de sensores inalámbricos (WSN), conociendo el funcionamiento al menos de forma básica de los microcontroladores, sus formas de comunicación con otros dispositivos periféricos.

De igual forma me ha permitido conocer y utilizar un sistema operativo en tiempo real basado en planificación de tareas y las diferentes formas de comunicación y sincronismo mediante el uso de herramientas como las colas o semáforos.

Por ultimo también me ha dado la oportunidad para conocer por encima las comunicaciones con dispositivos GPS, conociendo los distintos tipos de datos que pueden recabarse de la red de satélites mediante los distintos protocolos mas en profundidad el protocolo utilizado NMEA0183 v2.20 (National Marine Electronics Association).

A

Arp@Lab: Aplicación web desarrollada para envío de datos y su representación gráfica.

ADC: Conversión analógica digital.

B

Bluetooth: Es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos.

Baudrate: Número de unidades de señal por segundo.

C

Cortex-M3: procesador de 32 bits de ARM.

CPU: Unidad central de procesamiento.

D

DHCP: protocolo de configuración dinámica de host.

Diagramas de Gantt: Herramienta grafica para representación de planificación de trabajos y tareas.

E

E/S: Entrada/Salida.

F

FreeRTOS: Sistema operativo en tiempo real para dispositivo embebidos basado en Lenguaje C.

G

GUI: Interface gráfica de usuario.

GPIO: Puerta programable de entrada y salida de datos.

H

HTML: Siglas de HyperText Markup Language.

I

IDE: Entorno de desarrollo integrado.

IP: Identificación lógica y jerárquica, de interfaz de dispositivo dentro de una red que utilice el protocolo IP.

I2C: Bus de comunicaciones en serie.

L

Lan: Red de área local.

M

Mota: o mote es una placa con procesador y transmisión/recepción vía radio.

MCU: *Microcontrolador* circuito integrado programable, con capacidad de ejecución y memoria.

Mysql: Sistema de gestión de bases de datos relacional, y multiusuario

O

Open Source: Es el término con el que se conoce al software distribuido y desarrollado libremente.

P

PHP: Lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

R

RSSI: Indicador de fuerza de señal de recepción.

SSID: Nombre identificación de una Red Inalambrica (Wi-Fi)

T

TCP/IP: conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre computadoras.

U

UART: Transmisor-Receptor Asíncrono Universal

W

Wifi: Mecanismo de conexión de dispositivos electrónicos de forma inalámbrica.

Wireless: Comunicación inalámbrica o sin cables

- (Material Docente UOC) Universidad Oberta d Catalunya. Sistemas Empotrados, 2011. José María Gomez Cama, Francisco Hernández Ramírez, José López Vicario, Antoni Morell Pérez, Juan Daniel Prades García, Ignasi Vilajosana Guillen, Xavier Filajosana Guillen.
- (Material Docente UOC) Universidad Oberta de Catalunya. Presentación de documentos y elaboración de presentaciones. 2013 Roser Beneito Montagut
- Pérez García, Miguel Ángel .2011 . Instrumentación Electrónica. Editorial Thomson-Paraninfo
- LOS SANTOS, 2004. Alberto Los Santos Aransay, Aplicación de las redes de sensores en el entorno vehicular. Mayo 2009.
<http://www.albertolsa.com/wp-content/uploads/2010/04/rsi-aplicacion-de-las-redes-de-sensores-en-el-entorno-vehicular-alberto-los-santos.pdf>
- LORENZATI, 2010. Ing. Marcelo Lorenzati, Desarrollo de drivers y aplicaciones para FreeRTOS. Marzo 2010.
http://laboratorios.fi.uba.ar/lse/sase/2010/slides/SASE-2010_-FreeRTOS_Drivers_-Lorenzati.pdf
- FREERTOS, 2012. Richard Barry y Real Time Engineers Ltd. FreeRTOS – Market leading RTOS for embedded systems. 2004- 2012
<http://www.freertos.org/>
- NXP, 2012. NXP Semiconductors. LPCXpresso. 2012.
<http://knowledgebase.nxp.com>
- BRC-Electronics, 2012. Libraries & Examples. 2011.
<http://www.brc-electronics.nl/librarys>
- MBED, 2012. Rapid Prototyping for Microcontrollers. 2012. www.mbed.org

Anexo A

ESTRUCTURA TRAMAS DE DATOS

1 Tramas de datos

El sistema efectúa y recibe una serie de tramas de datos mediante protocolo HTTP, haciendo uso de las operaciones GET y POST. Estas tramas de datos deben de guardar el siguiente formato mostrado a continuación:

Indicar que las tramas debido a la longitud de buffer del dispositivo Wifly han tenido que ser divididas en tramas mas pequeñas, debiendo enviar dos tramas para estado de salidas y dos tramas para registro de muestreos datos sensores.

TRAMAS ESTADO DE SALIDAS

DIRECCION SISTEMA-APLICATIVO

Estas son enviadas a las aplicaciones:

- 1) <http://www.infoteli.es/proto/start/regstatus.php>
- 2) <http://www.infoteli.es/proto/start/regstatusb.php>

Enviando las siguientes tramas respectivamente:

- 1) <http://www.infoteli.es/proto/start/regstatus.php?O=X&A=X&B=X&SB=X&SW=X&SA=X>
- 2) <http://www.infoteli.es/proto/start/regstatusb.php?O=X&C=X&D=X&E=X&F=X&G=X&H=X>
(donde X es el valor 1=Activo 0=desactivado salvo para opción
O=X la cual indica Comando_Usuario=1 Comando_Consignas=0)

y las designaciones se corresponden con

O=Comando A=Salida 1 B=Salida 2 C=Salida 3 D=Salida 4 E=Salida 5 F=Salida 6
G=Salida 7 H=Salida 8 SB=Estado Red Bluetooth SW=Estado Red Wifi SA=Estado Alarma

TRAMAS MUESTREO SENSORES y GPS

Estas son enviadas a las aplicaciones:

- 1) <http://www.infoteli.es/proto/start/regsen.php>
- 2) <http://www.infoteli.es/proto/start/regsenb.php>
- 3) <http://www.infoteli.es/proto/start/reggps.php>
- 4) <http://www.infoteli.es/proto/start/reggpsb.php>

Igualmente estas han tenido que ser divididas en tramas menores por el problema indicado anteriormente

- 1) <http://www.infoteli.es/proto/start/regsen.php?T=X&P=X&H=X&L=X>
T=Temperatura, P=Presion, H=Humedad, L=Luminosidad. (X=Valor correspondiente)
- 2) <http://www.infoteli.es/proto/start/regsenb.php?DV=X&VV=X&P=X>
DV=Direccion Viento, VV=Velocidad Viento, P=Pluviosidad (X=Valor correspondiente)
- 3) <http://www.infoteli.es/proto/start/reggps.php?S=X&L=X&NS=X>
S=Nº Satélites, L=Latitud, NS=Norte-Sur (X=Valor correspondiente)
- 4) <http://www.infoteli.es/proto/start/reggpsb.php?T=X&EW=X&A=X>
T=Longitud EW=Este-Oeste A=Altitud (X=Valor correspondiente)

TRAMAS ACTUACION SOBRE SALIDAS Y VALORES CONSIGNAS

DIRECCION APLICATIVO-SISTEMA

Comando Recepción Actuación Salidas

El aplicativo envía orden de actuación sobre salidas en una única trama siendo recibido en el siguiente orden y formato desde la aplicación <http://www.infoteli.es/proto/feed/status.php>

RX:0,A,B,C,D,E,F,G,H,SB,SW,SA

y las designaciones se corresponden con los valores de:

**0=Comando A=Salida 1 B=Salida 2 C=Salida 3 D=Salida 4 E=Salida 5 F=Salida 6
G=Salida 7 H=Salida 8 SB=Estado Red Bluetooth SW=Estado Red Wifi SA=Estado Alarma**

Comando Recepcion Valores Consigna

El aplicativo envía orden de cambio de nuevos valores consignas en una única trama siendo recibido en el siguiente orden y formato desde la aplicación <http://www.infoteli.es/proto/feed/limites.php>

RX:0,LminT,LmaxT,LminP,LmaxP,LminL,LmaxL,LminH,LmaxH,Prxmin,Prxmax,VVmin,VVmax,RSSI

y las designaciones se corresponden con los valores de:

0=Comando	LminT=Límite min. Temperatura	LmaxT=límite max Temperatura
	LminP=Límite min. Presión	LminP=límite max Presión
	LminL=Límite min. Luminosidad	LminL=límite max Luminosidad
	LminH=Límite min. Humedad	LminH=límite max Humedad
	Prxmin=Límite min. Proximidad	Prxmax=límite max Proximidad
	VVmin=Límite min. Vel Viento	VVmax=límite max Vel viento

TRAMAS NODO EXTERNO VALORES MUESTREO

DIRECCION NODO EXTERNO-SISTEMA

Estas deben ser enviadas a la aplicación <http://www.infoteli.es/proto/start/regNodoExt.php>

El formato es el siguiente:

<http://www.infoteli.es/proto/start/regNodoExt.php?DV=X&VV=X&P=X>

Donde:

DV= Dirección Viento, VV=Velocidad Viento P=Pluviosidad (X=Valor correspondiente)

Por ultimo indicar que la trama correspondiente al Sensor WindSonic es la siguiente, la cual debe ser tratada por el nodo externo

Trama Sensor WinSonic

Trama de ejemplo → <stx> Q, 229,002.74, M, 00 <etx> 16

Dirección del nodo Windsonic Dirección del viento Velocidad Unidades Status Checksum

Anexo B

MANUAL USUARIO

1 Opciones Configuración Aplicación(Fichero Config_sys.h)

La programación del sistema se ha implementado utilizando para la configuración de parámetros que son necesarios para su funcionamiento un fichero cabecera (*Config_sys.h*).

En este fichero se debe de indicar ANTES de efectuar la compilación y grabación en el dispositivo parámetros básicos algunos de los que son importantes se indica aquí para así facilitar su localización y puedan ser cambiados, e indicar sus valores por defecto si los tienen:

Parámetros de la Red Wifi a Conectar:

```
#define SSID            "XXXXXXX" //SSID de la Red a conectar
#define Clave          "XXXXXXX" // Clave de la red Wifi
```

Parámetros configuración para el servidor Local

```
#define MYIP_1          192      // Definición de la IP servidor Local
#define MYIP_2          168
#define MYIP_3          1
#define MYIP_4          20
#define SUBMASK_1       255      //Submascara de Red
#define SUBMASK_2       255
#define SUBMASK_3       255
#define SUBMASK_4       0
#define GWIP_1          192      //Puerta de enlace predeterminada
#define GWIP_2          168
#define GWIP_3          1
#define GWIP_4          1
#define TCP_PORT_HTTP   8080     // Puerto para el servidor Local
```

Parámetros para Aplicación WEB (Servidor de aplicacion WEB)

```
#define DnsArpLab        "www.infoteli.es" //DNS aplicación WEB
#define PortArpLab       80              //Puerto remoto aplicación WEB
```

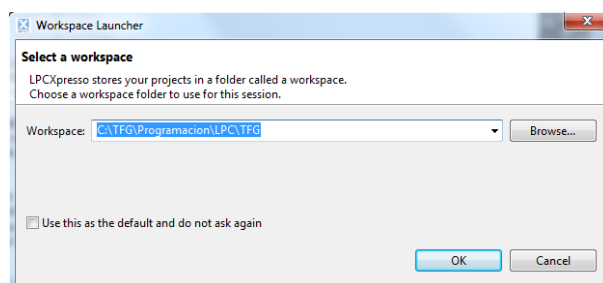
Parámetros Bluetooth y GPS, para incluir dispositivos bluetooth y GPS por Bluetooth

```
#define Addr_btw         "2013,8,260927" //Indicar MAC dispositivo Bluetooth instalado
#define GPS_MAC          "000C,A5,0F5DC9" //Indicar MAC dispositivo GPS a utilizar
#define Clave_Gps        "NAVMAN"       //Indicar Clave acceso dispositivo GPS utilizado
```

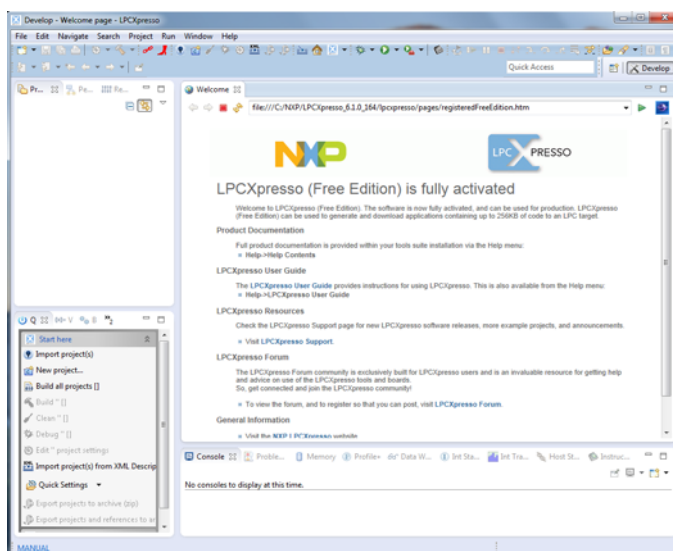
2. Compilación y grabación de la aplicación.

Adjunta a esta documentación se aporta el Workspace, con el código fuente de la aplicación en fichero [TFG_Final.zip](#), para su compilación y grabación en la MCU del sistema seguiremos los siguientes pasos.

- 1- **Instalación del IDE LPCXpresso:** Para ellos debe en primer lugar obtener el IDE así como los drivers de la placa LPCXpresso1769, estos los puede efectuar realizando el registro y descarga correspondientes en la Web indicada en el siguiente enlace ([Descargas](#)). Una vez que se ha procedido a su instalación se lleva a cabo la activación del IDE efectuando la solicitud del Código de activación en el menú de Ayuda del IDE, una vez recibido dicho código por email, se introduce quedando por tanto el entorno IDE de la placa listo para su uso.
- 2- **Importación del proyecto:** En primer lugar ejecutamos la aplicación LPCXpresso anteriormente instalada, esta nos presentará una ventana solicitando le indiquemos el directorio donde deseamos crear el Workspace de la aplicación.

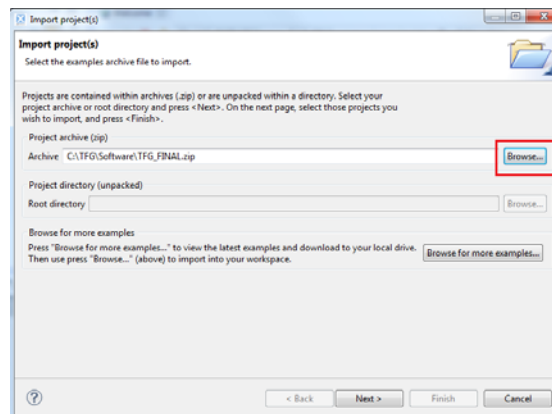


Indicado el directorio hacemos click sobre OK, lo que nos abrirá el IDE de trabajo presentándonos la ventana de bienvenida.

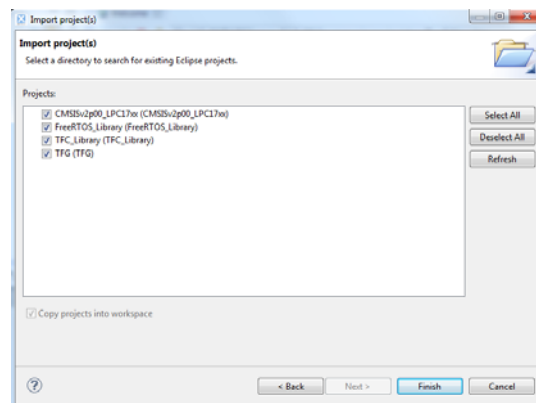


Una vez aquí hacer click sobre la opción [Import project](#) de la ventana inferior izquierda menú Iniciar aquí ([start here](#)). lo que nos presentará el menú de importación de proyectos, en el cual haciendo click sobre la opción buscador ([Browse](#)) del apartado Archivo de proyecto ([Project Archive \(zip\)](#)) nos da la opción para localizar y seleccionar el fichero del Workspace correspondiente a este proyecto que como

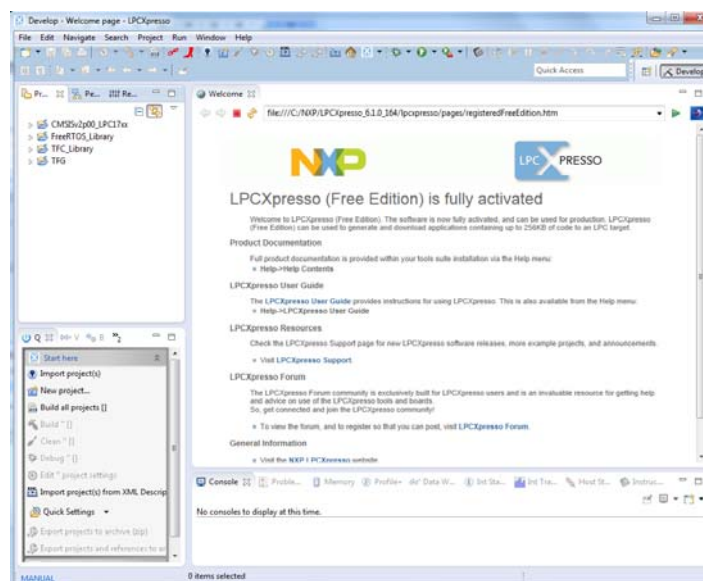
se ha indicado es [TFG_Final.zip](#). También puede hacerse directamente indicando en dicho espacio la ruta y nombre del fichero como puede verse en la siguiente captura.



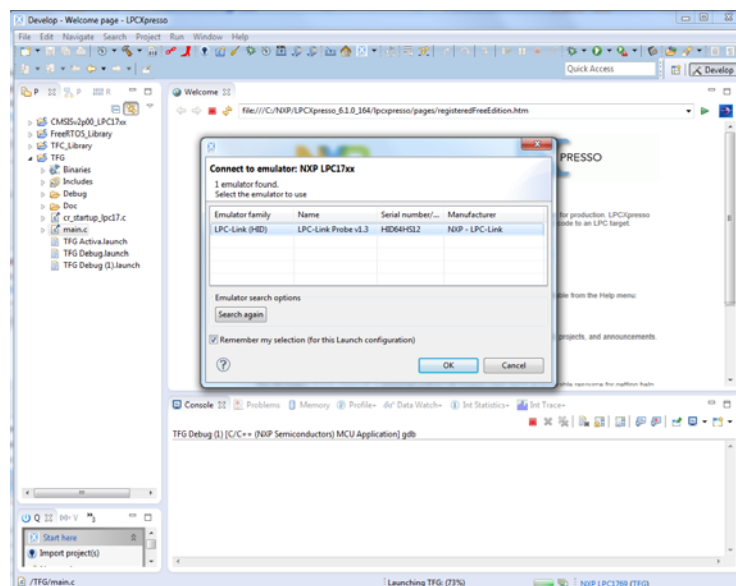
A continuación hacemos click sobre Siguiente (Next) y aparecerán las carpetas del Proyecto, (tantos las de sistemas utilizado, como las de librería y aplicación), debemos seleccionarlas todas.



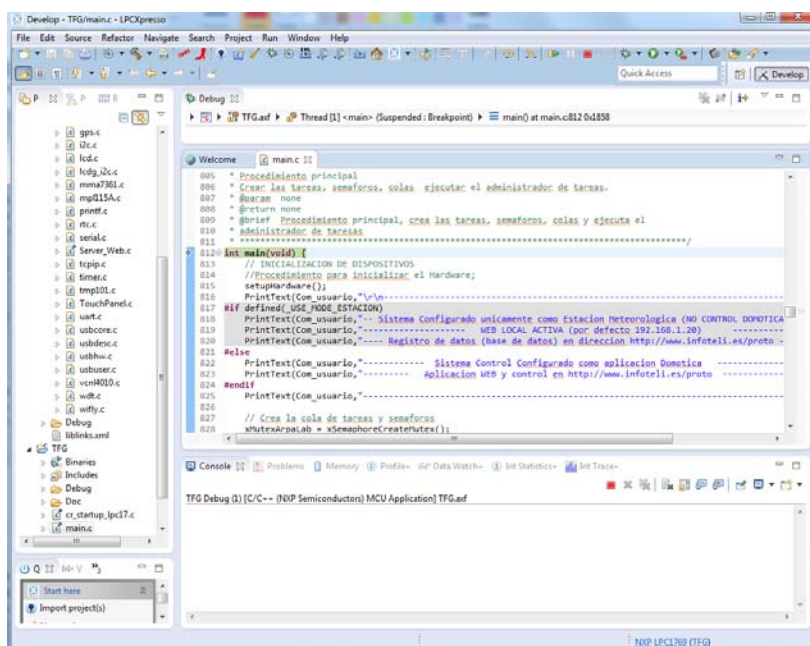
Haciendo click sobre Finalizar ([Finish](#)) quedara el entorno de trabajo cargado y configurado ya que este incluye todos los ficheros de configuración necesarios. Viendo por tanto las carpetas de los correspondientes proyectos en la ventana superior izquierda del IDE de trabajo.



- 3- **Compilación y Grabación:** Esto puede llevarse a cabo de diversas formas. La forma mas simple es abriendo la carpeta de proyecto TFG, nos situaremos sobre el fichero main(), a continuación en la barra de herramientas, haremos click sobre opción Debug, esto llevara a la compilación y grabación en la placa a traves del JTAG. Presentándonos la ventana de conexión con el emulador antes de llevar a la grabación del código en la placa.



Haciendo click sobre OK, pasara a llevar a cabo la grabación en el dispositivo, quedando la aplicación lista para su ejecución. Esta puede iniciarse ya que se ha indicado haciendo click sobre la opción *resume(F8)* de la barra de herramientas del IDE (menú *Run*).



Anotación: Si lo que se desea es que la aplicación inicie directamente deben de configurarse las opciones de debug para que no realice la parada del inicio del programa. Para se ha de desmarcar la opción *Stop in Startup*, dentro de opciones de *configuración de Debug*.

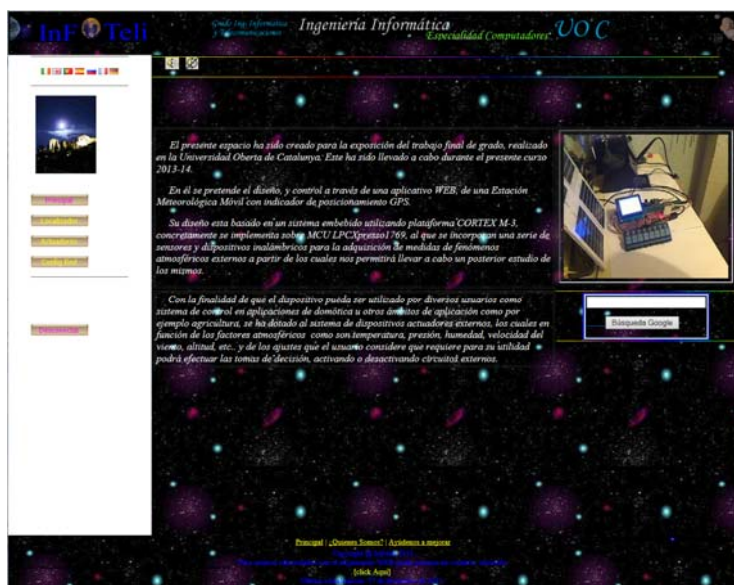
3. Manual Aplicación WEB

[\[Ir al índice\]](#)

Como se ha indicado el usuario interactúa con el sistema a través del aplicativo WEB. El código de la aplicación se acompaña adjunto a este documento en fichero comprimido [Aplicativo_WEB.zip](#), este se ha instalar en un servidor WEB. En este caso se encuentra instalado en la Url: <http://www.infoteli.es/proto>.

[Pagina Principal](#)

Mediante esta dirección se accede a la pagina principal que nos hace una breve presentación del proyecto como podemos ver en la siguiente figura y presenta al usuario el menú en un marco a la izquierda y otras opciones que describimos a continuación.



Principal: Este botón, permitirá al usuario en todo momento regresar a esta pagina principal de la aplicación.

Localización: Esta opción del menú haciendo click sobre ella traslada al usuario a la ventana de monitorización de los datos recabados por el sistema, permitiéndole conocer históricos por fechas, posicionamiento de la estación, y listado de todas las medidas tomadas.

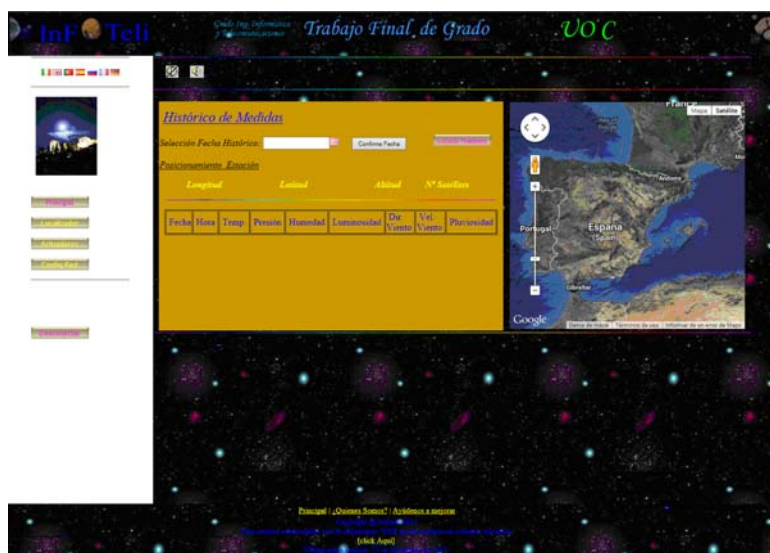
Actuadores: Esta opción del menu haciendo click sobre ella traslada al usuario a la ventana de la aplicación correspondiente a la interacción del usuario con el sistema en cuanto a poder introducir los valores consignas, activar o desactivar salidas, asi como monitorizar el estado de las mismas en todo momento, de igual forma podrá conocer el estado de alarma del sistema y el estado de las redes de que dispone el sistema, Wifi, Bluetooth y GPS.

Config Red: Esta opción del menú haciendo click sobre ella traslada al usuario a la ventana de configuración e interacción con el dispositivo Wifly del sistema, pudiendo conocer las redes en alcance del dispositivo y conectarse a ellas, configurar parametros de la red Wifi y envío de comandos específicos al dispositivo.

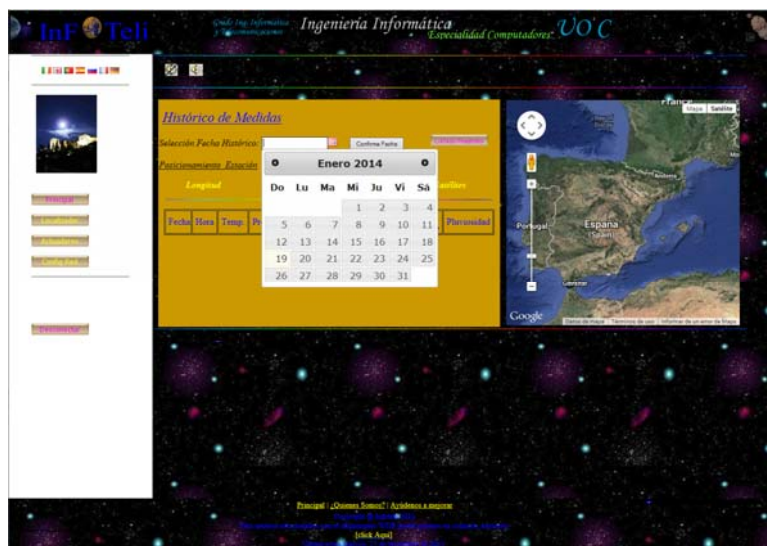
Sonido: Los iconos simulando altavoces en la parte superior permiten al usuario activar o anular el sonido de la aplicación WEB.

Por ultimo en la parte inferior se han incluido hipervínculos par utilidades de administración de la WEB de aplicación, como son: Contacto por email con el administrador [\[Click Aquí\]](#), pagina de opinión ([Ayúdenos a mejorar](#)) , pagina de presentación ([¿Quiénes Somos?](#)), y enlace con la pagina principal ([Principal](#)).

Esta se presenta al usuario como se muestra en la siguiente figura, en la cual aparecerá un listado de los datos recabados por la estación meteorológica de temperatura presión, humedad, luminosidad, etc.. además de los datos de longitud, latitud, altitud de la estación y nº de satélites en uso por el GPS, para recabar dichos datos. Igualmente presentara sobre el mapa, el cual es configurable por el usuario pudiendo obtener visión de la estación en tipo mapa satélite o callejero, la marca de situación de la posición de la estación.



Selección de Histórico por fechas: El usuario podrá indicar directamente la fecha en formato dd/mm/yy, o bien hacer click sobre el icono calendario pudiendo seleccionar la fecha del aplicativo calendario que se despliega como puede verse en la siguiente figura.



Una vez seleccionada una fecha, si existen datos recabados de esa fecha, estos serán mostrados en listado y el plano se situara la marca sobre la posición donde la estación tomo dichos datos.

Seleccionando la fecha actual, si el sistema esta activo, mostrara los datos que va recabando y la localización de la estación en ese momento, en tiempo real.

En la siguiente figura puede verse los datos y posición correspondientes al día 6/01/2014.



De esta forma los datos son presentados al usuario de una forma comprensible y mediante una interfaz que se ha considerado de fácil manejo.

Listado Muestreos: Haciendo Click sobre esta opción el usuario podrá obtener un listado completo de todos los datos existentes como se muestra a continuación.

Area Diseño Sistemas Embebidos. VOIC

Num. Muestra	Fecha	Hora	Temperatura	Presión	Humedad	Luminosidad	Dir Viento	Vel Viento	PLuviosidad	N_Satélites	Latitud	OLt	Longitud	OLg	Altitud
1525	10/01/2014	07:21:24	18.8	954.892	55.4	0	0	0	0	7	3712.4229 N	337.7444 W	717		
1524	10/01/2014	07:15:55	19	954.277	54.6	0	0	0	0	6	3712.4245 N	337.7422 W	707		
1523	10/01/2014	06:47:55	18.6	956.238	55.3	0	0	0	0	5	3712.4119 N	337.7643 W	776		
1522	10/01/2014	06:35:55	18.5	955.871	56.1	0	0	0	0	7	3712.4251 N	337.7463 W	719.5		
1521	10/01/2014	06:22:50	18.5	954.525	56.6	0	0	0	0	5	3712.4264 N	337.7523 W	736.1		
1520	10/01/2014	06:13:01	18.2	956.481	57.3	0	0	0	0	6	3712.4237 N	337.7493 W	729.8		
1519	10/01/2014	06:01:06	18.4	955.503	56.6	0	0	0	0	5	3712.423 N	337.753 W	732.3		
1518	10/01/2014	05:52:10	18.4	956.238	56.7	0	0	0	0	5	3712.4374 N	337.7559 W	733.7		
1516	10/01/2014	05:30:57	18.5	953.545	55.7	0	0	0	0	5	3712.429 N	0	0		
1517	10/01/2014	05:39:54	18.4	955.871	56.8	0	0	0	0	4	3712.4493 N	337.7653 W	763.4		
1515	10/01/2014	05:15:53	18.6	955.871	56.4	0	0	0	0	3	3712.6887 N	338.0535 W	738.8		
1514	10/01/2014	05:03:05	18.5	954.277	56.1	0	0	0	0	4	3712.4192 N	337.7494 W	731.7		
1513	10/01/2014	04:49:45	18.5	954.158	56.1	0	0	0	0	3	3712.4236 N	337.774 W	738.8		
1512	10/01/2014	04:43:20	18.6	954.525	55.9	0	0	0	0	4	3712.4346 N	337.7931 W	744.6		
1511	10/01/2014	04:30:48	18.4	954.525	56.9	0	0	0	0	4	3712.4334 N	337.816 W	745.2		
1510	10/01/2014	04:04:56	18.5	955.871	57.5	0	0	0	0	4	3712.4243 N	337.7637 W	757		
1509	10/01/2014	03:56:47	18.5	957.217	57	0	0	0	0	5	3712.4187 N	337.7509 W	725.2		
1508	10/01/2014	03:47:35	18.3	957.585	56.9	0	0	0	0	5	3712.4198 N	337.7428 W	719.5		
1507	10/01/2014	03:41:59	18.3	956.849	57.5	0	0	0	0	4	3712.4216 N	337.7451 W	721.3		
1506	10/01/2014	03:33:59	18.4	956.849	56.9	0	0	0	0	5	3712.4235 N	337.7539 W	730.9		
1505	10/01/2014	03:15:11	18.4	957.217	57.5	0	0	0	0	0	0	337.8466 W	667.3		
1504	10/01/2014	03:08:22	18.4	955.871	57.7	0	0	0	0	3	3712.5022 N	337.7984 W	667.3		
1503	10/01/2014	02:56:06	18.4	955.503	57.8	0	0	0	0	3	3712.3467 N	337.7337 W	667.3		
1502	10/01/2014	02:46:34	18.5	957.217	58.2	0	0	0	0	3	3712.3591 N	337.7454 W	667.3		
1500	10/01/2014	02:21:58	18.8	957.953	56.7	0	0	0	0	4	3712.4247 N	0	0		
1501	10/01/2014	02:28:54	18.8	956.606	57	0	0	0	0	3	3712.3706 N	337.7634 W	657.3		
1499	10/01/2014	02:07:01	18.7	957.585	57	0	0	0	0	3	3712.4202 N	337.7471 W	715.4		
1498	10/01/2014	02:00:37	18.6	957.217	57.1	0	0	0	0	4	3712.4217 N	337.7482 W	726.9		
1497	10/01/2014	01:53:09	18.7	957.217	56.5	0	0	0	0	3	3712.4194 N	337.7434 W	743.4		
1496	10/01/2014	01:44:13	18.7	957.953	56.1	0	0	0	0	3	3712.4247 N	337.7574 W	743.4		
1495	10/01/2014	01:17:56	18.9	957.953	56.9	0	0	0	0	3	3712.4242 N	337.7579 W	744.8		
1494	10/01/2014	01:08:04	19	958.321	56.7	0	0	0	0	4	3712.4169 N	337.7405 W	711.6		
1493	10/01/2014	00:54:12	18.9	956.238	56.6	0	0	0	0	5	3712.4124 N	337.7397 W	711.3		

Principal | ¿Quieres Sonar? | Ayúdanos a mejorar

Copyright © InF Teli 2013

Para sonidos rotacionados con el alojamiento VOIC puede ponerlos en columna, haciendo click Aquí

Última versión: 27 de diciembre de 2013

Mediante la aplicación de la pagina de Actuadores el usuario pude realizar las siguientes operaciones:

Configuración Valores Consigna: El usuario podrá introducir los valores consigna a partir de los cuales el sistema tomara actuaciones, como son limites máximos y mínimos de temperatura, presión, humedad y luminosidad, así si los valores detectados están fuera de dichos rangos, estos actuaran sobre la salida correspondiente. Así la programación efectuada a modo de demostración sigue los siguientes criterios:

- Temperatura superior al Limite max.: Activa salida Nº 1
- Temperatura inferior al Limite máx.: Desactiva la salida Nº 1
- Iluminación por debajo de limite min.: Activa Salida Nº 3
- Iluminación por encima del limite min.: Desactiva Salida Nº 3
- Presión por debajo de limite min.: Activa Salida Nº 6
- Presión por encima de limite min.: Desactiva Salida Nº 6
- Humedad por debajo de limite min.: Activa Salida Nº 5
- Humedad por encima de limite min.: Desactiva Salida Nº 5

Para ello el usuario indicara los valores en los campos que se presenta y hará click sobre enviar para enviar los valores al sistema, este presentara una ventana de confirmación como puede verse en la figura de la izquierda la cual si acepta los valores serán enviados sino la acción será cancelada.



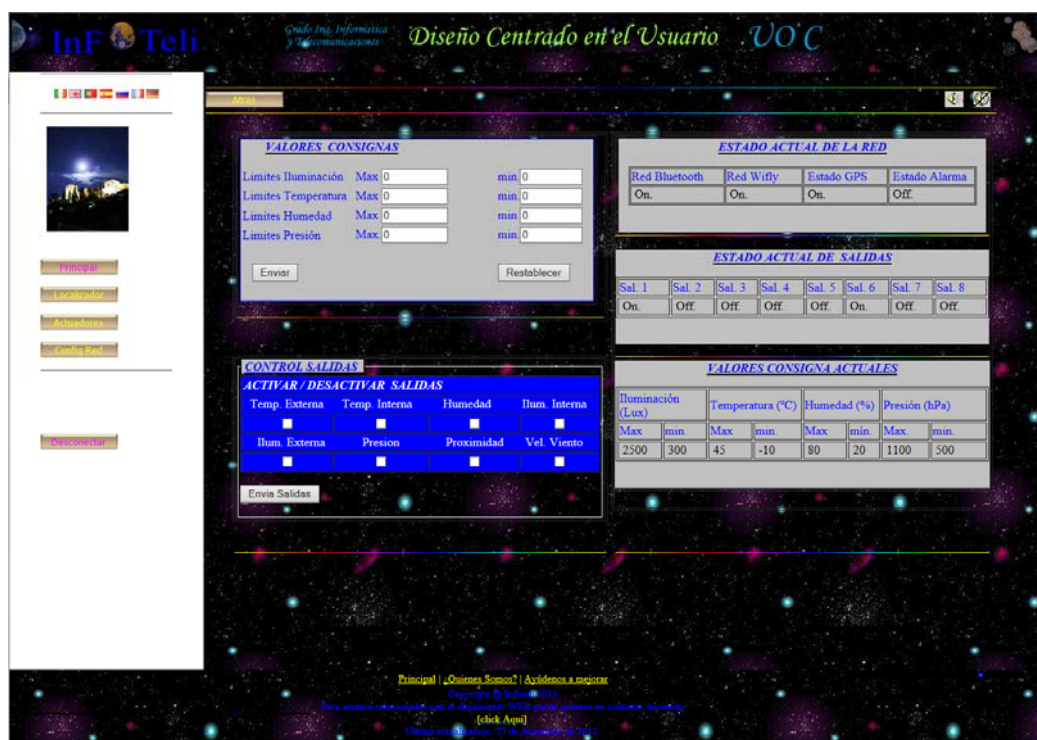
Activación y desactivación de salidas: El usuario podrá activar o desactivar, marcando sobre las casillas de las de los correspondientes actuadores en control de salidas, una vez indicado el estado de todas las salidas como se deseen tener, haciendo click sobre Enviar salidas, se presentara un mensaje para solicitar confirmación, si el usuario acepta los datos serán enviados y las salidas del sistema tomaran el estado que el usuario haya indicado.



Monitorización de estados:

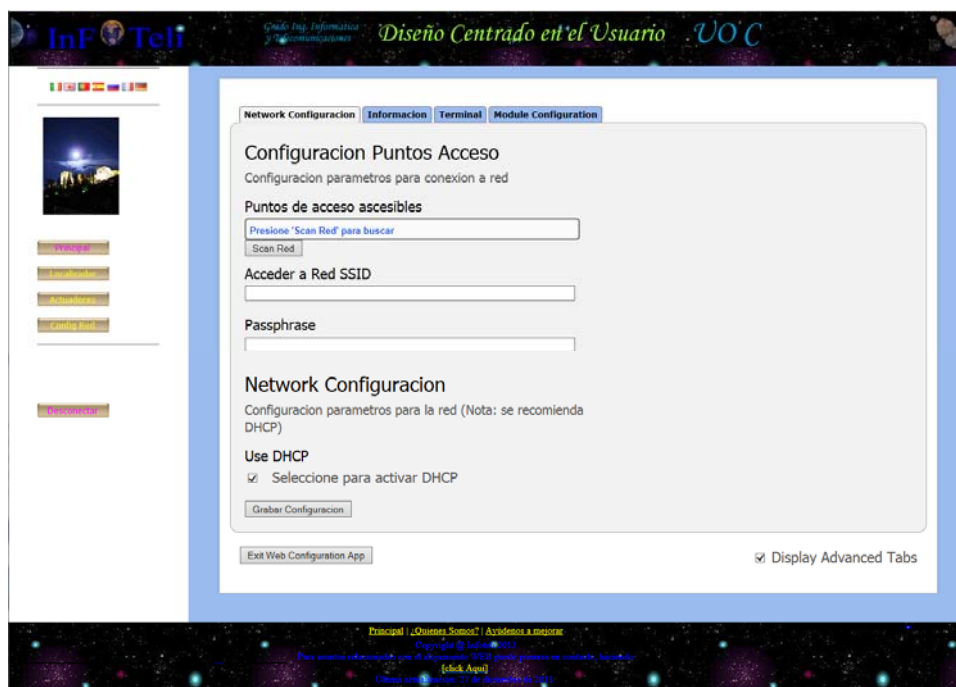
Por ultimo en la parte de la izquierda se muestran en todo momento:

- El estado de las redes
- El estado de alarma
- El estado de salidas
- Los Valores consignas en vigor



Opción Menú Configuración Red

Mediante esta parte del aplicativo el usuario puede configurar el punto de acceso a la red Wifi de la estación meteorológica, conociendo los puntos de acceso disponibles y pudiendo configurar el dispositivo para su acceso a uno de ellos.



Las diversas opciones de este aplicativo son:

Network Configuración: Permite escanear los puntos de acceso disponibles y configurar el dispositivo Wifly para conexión a uno de ellos.

Información: Da al usuario la información del dispositivo Wifly instalado

Terminal: Abre un terminal para poder interactuar con el dispositivo Wifly

Configuración modulo: Permite configurar parámetros del modulo Wifly como velocidad del UART.

Opción Menú Desconectar

Esta opción simula el cierre de la conexión con la aplicación, devolviendo a la pagina que debiera ser la entrada mediante usuario y contraseña, ya que el aplicativo debería llevarse a cabo mediante conexión segura y lógicamente protegido por contraseña.

No se ha implementado por falta de tiempo, ya que la interfaz es únicamente una demostración de como puede realizarse, quedando la misma fuera del alcance del proyecto.

Por tanto esta opción redirige a la pagina principal del servidor Web <http://www.infoteli.es>, para volver a la aplicación hay que volver a ingresar a la url <http://www.infoteli.es/proto>

Anexo C

DOCUMENTACION API DEL SOFTWARE

DOCUMENTACION API
APLICACION
ESTACION METEOROLOGICA MOVIL
CON
POSICIONAMIENTO
GPS

Miguel Angel Iglesias Jimenez
Version V1.0
22 Enero 2014

Índice de estructura de datos

Estructura de datos 7

Lista de estructuras con una breve descripción:

attribute	7
_SER_BUF_T	7
_REQUEST_TYPE::_BM	8
_BM	8
_CDC_ABSTRACT_CONTROL_MANAGEMENT_DESCRIPTOR	8
_CDC_CALL_MANAGEMENT_DESCRIPTOR	9
_CDC_HEADER_DESCRIPTOR	9
_CDC_LINE_CODING	9
_CDC_UNION_1SLAVE_DESCRIPTOR	10
_CDC_UNION_DESCRIPTOR	10
_HID_DESCRIPTOR	10
_HID_DESCRIPTOR_LIST	11
_HID_DESCRIPTOR::_HID_DESCRIPTOR_LIST	11
_MSC_CBW	12
_MSC_CSW	12
_REQUEST_TYPE	13
_USB_COMMON_DESCRIPTOR	13
_USB_CONFIGURATION_DESCRIPTOR	13
_USB_DEVICE_DESCRIPTOR	14
_USB_DEVICE_QUALIFIER_DESCRIPTOR	14
_USB_DMA_DESCRIPTOR	15
_USB_ENDPOINT_DESCRIPTOR	15
_USB_EP_DATA	16
_USB_INTERFACE_DESCRIPTOR	16
_USB_SETUP_PACKET	17
_USB_STRING_DESCRIPTOR	17
ACL_Data	17
ActSalidas	18
ADC_channels_data	18
Coord_Touch	19
Limites	19
MENU	20
RTCTime	20

Indice de archivos

Lista de archivos.....19

Lista de todos los archivos documentados y con descripciones breves:

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/actuadores.h (Fichero cabecera:(libreria manejo Actuadores de sistema consistentes en 8 rele controlados por PCF8574))	21
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/adc.h (Fichero cabecera: (Configuracion canales ADC))	24
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/arpalab.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/btw.h (Fichero cabecera: (Libreria BTW H05))	26
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Config_sis.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/DTH22.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Eeprom.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/ethmac.	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/gps.h (Fichero cabecera:(Recepcion datos GPS via bluetooth))	34
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/i2c.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/lcd.h (Fichero cabecera:(lib control dispositivo Eeprom 24LXXX))	36
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/lcdg_i2c.h (Fichero cabecera:(libreria manejo LCD grafico 128x64 (192x64) por bus I2C mediante dispositivo BV4611))	42
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/menu.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/MMA7361.h (Fichero cabecera:(Driver dispositivo acelerometro MMA7361))	50
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/MPL115A.h (Fichero cabecera:(libreria manejo sensor presuin MPL115A por bus I2C))	52
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/printf.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/rtc.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/serial.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Server_Web.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/tcpip.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/timer.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/tmp101.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/TouchPanel.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/type.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/uart.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/VCNL4010.h (Fichero cabecera:(libreria manejo sensor luminosidad y proximidad VCNL4010A))	54
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/wdt.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/wifly.h;Error! Marcador no definido.	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Fonts/Arial5x8.h	
definido.	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Fonts/Comic5x8.h (Fich fuente para GLCD 128x54 fuente Comic5x8))	32

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Fonts/font5x8.h (Fichero fuente para GLCD 128x54 fuente 5x8)	33
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Images/Logo_UOC.h ;	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/adcuser.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/audio.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/cdc.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/cdcuser.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/hid.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/hiduser.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/msc.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/mscuser.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usbcfg.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usbcore.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usbdesc.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usbhw.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usbreg.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usbuser.h	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Web/webside.h (Fichero cabecera: Para declaracion constantes Paginas Web)	56
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/actuadores.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/adc.c (Fichero funciones control ADC sistema LPC1769)	57
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/arpalab.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/btw.c (Liberia Libreria para manejo de dispositivo Btw H05)	59
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/dth22.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/Eeprom.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/ethmac.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/Fonts.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/gps.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/i2c.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/lcd.c	

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/lcdg_i2c.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/mma7361.c (Libreria para manejo del dispositivo acelemetro DHT22A Sensor Humedad Realativa y temperatura)	64
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/mpl115A.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/printf.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/rtc.c (Libreria para manejo RTC)	67
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/serial.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/Server_Web.c (Liberia Libreria para implementacion de servidor para monitoreo del sistema de medidas)	74
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/tcpip.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/timer.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/tmp101.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/TouchPanel.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/uart.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/usbcore.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/usbdesc.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/usbhw.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/usbuser.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/vcnl4010.c	
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/wdt.c (Liberia configuracion y uso de watchDog)	76
C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/wifly.c	
C:/TFG/Programacion/LPC/Nuevo/TFG/cr_startup_lpc17.c	
C:/TFG/Programacion/LPC/Nuevo/TFG/main.c (Sistema Estacion Meteorologica con posicionamiento GPS)	78
C:/TFG/Programacion/LPC/PAC3/TFC_Library/Debug/src/arpalab.d	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/Debug/src/printf.d	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/Debug/src/timer.d	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/Debug/src/uart.d	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/Debug/src/wifly.d	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/inc/arpalab.h	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/inc/Config_sis.h	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/inc/printf.h	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/inc/timer.h	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/inc/type.h	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/inc/uart.h	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/inc/wifly.h	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/src/arpalab.c	
TFG/Programacion/LPC/PAC3/TFC_Library/src/printf.c	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/src/timer.c	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/src/uart.c	
C:/TFG/Programacion/LPC/PAC3/TFC_Library/src/wifly.c	

Documentación de las estructuras de datos

Referencia de la Unión `__attribute__`

Métodos públicos

- `struct {`
- `uint8_t L`
- `uint8_t H`
- `} __attribute__ ((packed)) WB`

Campos de datos

- `uint16_t W`

Descripción detallada

Definición en la línea 23 del archivo `usb.h`.

La documentación para esta unión fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h`

Referencia de la Estructura `__SER_BUF_T`

Campos de datos

- `unsigned char data [SER_BUF_SIZE]`
- `unsigned int wrIdx`
- `unsigned int rdIdx`

Descripción detallada

Definición en la línea 42 del archivo `serial.c`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/serial.c`

Referencia de la Estructura **_REQUEST_TYPE::_BM**

Campos de datos

- uint8_t **Recipient**: 5
 - uint8_t **Type**: 2
 - uint8_t **Dir**: 1
-

Descripción detallada

Definición en la línea 51 del archivo usb.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h

Referencia de la Estructura **_BM**

Campos de datos

- uint8_t **Recipient**: 5
 - uint8_t **Type**: 2
 - uint8_t **Dir**: 1
-

Descripción detallada

Definición en la línea 45 del archivo usb.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h

Referencia de la Estructura **_CDC_ABSTRACT_CONTROL_MANAGEMENT_DESCRIPTOR**

Campos de datos

- uint8_t **bFunctionLength**
 - uint8_t **bDescriptorType**
 - uint8_t **bDescriptorSubtype**
 - uint8_t **bmCapabilities**
-

Descripción detallada

Definición en la línea 196 del archivo cdc.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/cdc.h

Referencia de la Estructura **_CDC_CALL_MANAGEMENT_DESCRIPTOR**

Campos de datos

- `uint8_t bFunctionLength`
 - `uint8_t bDescriptorType`
 - `uint8_t bDescriptorSubtype`
 - `uint8_t bmCapabilities`
 - `uint8_t bDataInterface`
-

Descripción detallada

Definición en la línea 185 del archivo `cdc.h`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/cdc.h`

Referencia de la Estructura **_CDC_HEADER_DESCRIPTOR**

Campos de datos

- `uint8_t bFunctionLength`
 - `uint8_t bDescriptorType`
 - `uint8_t bDescriptorSubtype`
 - `uint16_t bcdCDC`
-

Descripción detallada

Definición en la línea 175 del archivo `cdc.h`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/cdc.h`

Referencia de la Estructura **_CDC_LINE_CODING**

Campos de datos

- `uint32_t dwDTERate`
 - `uint8_t bCharFormat`
 - `uint8_t bParityType`
 - `uint8_t bDataBits`
-

Descripción detallada

Definición en la línea 223 del archivo `cdc.h`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/cdc.h`

Referencia de la Estructura `_CDC_UNION_1SLAVE_DESCRIPTOR`

Campos de datos

- `CDC_UNION_DESCRIPTOR sUnion`
 - `uint8_t bSlaveInterfaces [1]`
-

Descripción detallada

Definición en la línea 215 del archivo `cdc.h`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/cdc.h`

Referencia de la Estructura `_CDC_UNION_DESCRIPTOR`

Campos de datos

- `uint8_t bFunctionLength`
 - `uint8_t bDescriptorType`
 - `uint8_t bDescriptorSubtype`
 - `uint8_t bMasterInterface`
-

Descripción detallada

Definición en la línea 206 del archivo `cdc.h`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/cdc.h`

Referencia de la Estructura `_HID_DESCRIPTOR`

Estructuras de datos

- `struct _HID_DESCRIPTOR_LIST`

Métodos públicos

- `struct`
- `_HID_DESCRIPTOR::_HID_DESCRIPTOR_LIST __attribute__((packed)) DescriptorList[1]`

Campos de datos

- `uint8_t bLength`
 - `uint8_t bDescriptorType`
 - `uint16_t bcdHID`
 - `uint8_t bCountryCode`
 - `uint8_t bNumDescriptors`
-

Descripción detallada

Definición en la línea 40 del archivo hid.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/hid.h

Referencia de la Estructura `_HID_DESCRIPTOR_LIST`

Campos de datos

- `uint8_t bDescriptorType`
- `uint16_t wDescriptorLength`

Descripción detallada

Definición en la línea 67 del archivo hid.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/hid.h

Referencia de la Estructura `_HID_DESCRIPTOR::_HID_DESCRIPTOR_LIST`

Campos de datos

- `uint8_t bDescriptorType`
- `uint16_t wDescriptorLength`

Descripción detallada

Definición en la línea 47 del archivo hid.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/hid.h

Referencia de la Estructura `_MSC_CBW`

Campos de datos

- `uint32_t dSignature`
 - `uint32_t dTag`
 - `uint32_t dDataLength`
 - `uint8_t bmFlags`
 - `uint8_t bLUN`
 - `uint8_t bCBLength`
 - `uint8_t CB [16]`
-

Descripción detallada

Definición en la línea 53 del archivo `msc.h`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/msc.h`

Referencia de la Estructura `_MSC_CSW`

Campos de datos

- `uint32_t dSignature`
 - `uint32_t dTag`
 - `uint32_t dDataResidue`
 - `uint8_t bStatus`
-

Descripción detallada

Definición en la línea 64 del archivo `msc.h`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/msc.h`

Referencia de la Unión `_REQUEST_TYPE`

Estructuras de datos

- `struct _BM`

Métodos públicos

- `struct _REQUEST_TYPE::_BM __attribute__((packed)) BM`

Campos de datos

- `uint8_t B`
-

Descripción detallada

Definición en la línea 50 del archivo `usb.h`.

La documentación para esta unión fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h`

Referencia de la Estructura `_USB_COMMON_DESCRIPTOR`

Campos de datos

- `uint8_t bLength`
 - `uint8_t bDescriptorType`
-

Descripción detallada

Definición en la línea 223 del archivo `usb.h`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h`

Referencia de la Estructura `_USB_CONFIGURATION_DESCRIPTOR`

Campos de datos

- `uint8_t bLength`
- `uint8_t bDescriptorType`
- `uint16_t wTotalLength`
- `uint8_t bNumInterfaces`
- `uint8_t bConfigurationValue`
- `uint8_t iConfiguration`
- `uint8_t bmAttributes`
- `uint8_t bMaxPower`

Descripción detallada

Definición en la línea 181 del archivo usb.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h

Referencia de la Estructura **_USB_DEVICE_DESCRIPTOR**

Campos de datos

- uint8_t **bLength**
- uint8_t **bDescriptorType**
- uint16_t **bcdUSB**
- uint8_t **bDeviceClass**
- uint8_t **bDeviceSubClass**
- uint8_t **bDeviceProtocol**
- uint8_t **bMaxPacketSize0**
- uint16_t **idVendor**
- uint16_t **idProduct**
- uint16_t **bcdDevice**
- uint8_t **iManufacturer**
- uint8_t **iProduct**
- uint8_t **iSerialNumber**
- uint8_t **bNumConfigurations**

Descripción detallada

Definición en la línea 150 del archivo usb.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h

Referencia de la Estructura **_USB_DEVICE_QUALIFIER_DESCRIPTOR**

Campos de datos

- uint8_t **bLength**
- uint8_t **bDescriptorType**
- uint16_t **bcdUSB**
- uint8_t **bDeviceClass**
- uint8_t **bDeviceSubClass**
- uint8_t **bDeviceProtocol**
- uint8_t **bMaxPacketSize0**
- uint8_t **bNumConfigurations**
- uint8_t **bReserved**

Descripción detallada

Definición en la línea 168 del archivo usb.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h

Referencia de la Estructura **_USB_DMA_DESCRIPTOR**

Campos de datos

- uint32_t **BufAdr**
- uint16_t **BufLen**
- uint16_t **MaxSize**
- uint32_t **InfoAdr**
- union {
- struct {
- uint32_t **Link**: 1
- uint32_t **IsoEP**: 1
- uint32_t **ATLE**: 1
- uint32_t **Rsrvd**: 5
- uint32_t **LenPos**: 8
- } **Type**
- uint32_t **Val**
- } **Cfg**

Descripción detallada

Definición en la línea 65 del archivo usbhw.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usbhw.h

Referencia de la Estructura **_USB_ENDPOINT_DESCRIPTOR**

Campos de datos

- uint8_t **bLength**
 - uint8_t **bDescriptorType**
 - uint8_t **bEndpointAddress**
 - uint8_t **bmAttributes**
 - uint16_t **wMaxPacketSize**
 - uint8_t **bInterval**
-

Descripción detallada

Definición en la línea 206 del archivo usb.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h

Referencia de la Estructura **_USB_EP_DATA**

Campos de datos

- uint8_t * **pData**
- uint16_t **Count**

Descripción detallada

Definición en la línea 24 del archivo usbcore.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usbcore.h

Referencia de la Estructura **_USB_INTERFACE_DESCRIPTOR**

Campos de datos

- uint8_t **bLength**
- uint8_t **bDescriptorType**
- uint8_t **bInterfaceNumber**
- uint8_t **bAlternateSetting**
- uint8_t **bNumEndpoints**
- uint8_t **bInterfaceClass**
- uint8_t **bInterfaceSubClass**
- uint8_t **bInterfaceProtocol**
- uint8_t **iInterface**

Descripción detallada

Definición en la línea 193 del archivo usb.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h

Referencia de la Estructura `_USB_SETUP_PACKET`

Campos de datos

- `REQUEST_TYPE` **bmRequestType**
 - `uint8_t` **bRequest**
 - `WORD_BYTE` **wValue**
 - `WORD_BYTE` **wIndex**
 - `uint16_t` **wLength**
-

Descripción detallada

Definición en la línea 82 del archivo `usb.h`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h`

Referencia de la Estructura `_USB_STRING_DESCRIPTOR`

Campos de datos

- `uint8_t` **bLength**
 - `uint8_t` **bDescriptorType**
 - `uint16_t` **bString**
-

Descripción detallada

Definición en la línea 216 del archivo `usb.h`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/USBCDC/usb.h`

Referencia de la Estructura `ACL_Data`

Campos de datos

- `float` **Medida_X**
 - `float` **Medida_Y**
 - `float` **Medida_Z**
 - `float` **rho**
 - `float` **phi**
 - `float` **theta**
-

Descripción detallada

Definición en la línea 23 del archivo `MMA7361.h`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/MMA7361.h

Referencia de la Estructura ActSalidas

Campos de datos

- int **Comando**
- int **Act_1**
- int **Act_2**
- int **Act_3**
- int **Act_4**
- int **Act_5**
- int **Act_6**
- int **Act_7**
- int **Act_8**
- int **Bluet**
- int **Wifly**
- int **Alarma**

Descripción detallada

Definición en la línea 44 del archivo Config_sis.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Config_sis.h

Referencia de la Estructura ADC_channels_data

Campos de datos

- int **channelA**
- int **channelB**
- int **channelC**
- int **channelD**
- int **channelE**
- int **channelF**
- int **channelG**
- int **channelH**

Descripción detallada

Definición en la línea 41 del archivo adc.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/adc.h

Referencia de la Estructura Coord_Touch

Campos de datos

- int **X_coord**
 - int **Y_coord**
 - int **X_max**
 - int **X_min**
 - int **Y_max**
 - int **Y_min**
-

Descripción detallada

Definición en la línea 21 del archivo TouchPanel.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/TouchPanel.h

Referencia de la Estructura Limites

Campos de datos

- int **Comando**
 - double **Tmin**
 - double **Tmax**
 - double **Pmin**
 - double **Pmax**
 - double **Hmin**
 - double **Hmax**
 - double **Lmin**
 - double **Lmax**
 - double **VVmin**
 - double **VVmax**
 - double **Prmin**
 - double **Prmax**
 - int **RSSImin**
-

Descripción detallada

Definición en la línea 60 del archivo Config_sis.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Config_sis.h

Referencia de la Estructura MENU

Campos de datos

- char **TipoMenu**
 - char * **TextoMenu**
 - char * **PtrMenuSeleccion**
-

Descripción detallada

Definición en la línea 12 del archivo menu.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/menu.h

Referencia de la Estructura RTCTime

Campos de datos

- uint32_t **RTC_Sec**
 - uint32_t **RTC_Min**
 - uint32_t **RTC_Hour**
 - uint32_t **RTC_Mday**
 - uint32_t **RTC_Mon**
 - uint32_t **RTC_Year**
 - uint32_t **RTC_Wday**
 - uint32_t **RTC_Yday**
-

Descripción detallada

Definición en la línea 28 del archivo rtc.h.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/rtc.h

Documentación de archivos

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/actuadores.h

Fichero cabecera:(libreria manejo Actuadores de sistema consistentes en 8 reles controlados por PCF8574)

'defines'

- #define **Act_Salidas** 8
- #define **AL1** (0x1)
- #define **AL2** (0x2)
- #define **AT1** (0x4)
- #define **AT2** (0x8)
- #define **AH1** (0x10)
- #define **AH2** (0x20)
- #define **AP1** (0x40)
- #define **AP2** (0x80)
- #define **Mask_Lum_1** (AP2 | AP1 | AH2 | AH1 | AT2 | AT1 | AL2)
- #define **Mask_Lum_2** (AP2 | AP1 | AH2 | AH1 | AT2 | AT1 | AL1)
- #define **Mask_Temp_1** (AP2 | AP1 | AH2 | AH1 | AT2 | AL2 | AL1)
- #define **Mask_Temp_2** (AP2 | AP1 | AH2 | AH1 | AT1 | AL2 | AL1)
- #define **Mask_Hum_1** (AP2 | AP1 | AH2 | AT2 | AT1 | AL2 | AL1)
- #define **Mask_Hum_2** (AP2 | AP1 | AH1 | AT2 | AT1 | AL2 | AL1)
- #define **Mask_Prox_1** (AP2 | AH2 | AH1 | AT2 | AT1 | AL2 | AL1)
- #define **Mask_Prox_2** (AP1 | AH2 | AH1 | AT2 | AT1 | AL2 | AL1)

Funciones

- int **lee_estado_salidas** (void)
- int **lee_salida** (uint8_t Salida)
- int **Act_des_salida** (uint8_t Salida, int Act_Desact)
- int **Desactiva_salidas** ()
- int **Verifica_Driver_salidas** ()

Descripción detallada

Fichero cabecera:(libreria manejo Actuadores de sistema consistentes en 8 reles controlados por PCF8574)

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera:(funciones manejo LCDG grafico 128x64 (192x64) 1602 configuracion 4bit)

Fecha:

25/10/2013

Nota:

Copyright Modificación del obtenido con la asignatura para manejo de los 4 UART.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **actuadores.h**.

Documentación de las funciones

int Act_des_salida (uint8_t Salida, int Act_Desact)

Descripción: Activa o desactiva una salida concreta

Parámetros:

Salida	Nº de salida (0-8)
Act_Desact	0 desactiva la salida 1 la activa

Devuelve:

Estado de la salida o (-1) si error

Definición en la línea 125 del archivo actuadores.c.

int Desactiva_salidas ()

Descripción: Desactiva todas las salidas

Parámetros:

none	
------	--

Devuelve:

Estado de las salidas o (-1) si error

Definición en la línea 98 del archivo actuadores.c.

int lee_estado_salidas (void)

Descripción: Retorna el estado de todas las salidas

Parámetros:

none	
------	--

Devuelve:

Estado de las salidas o (-1) si error

Descripción: Envía I2C Envía los comandos a I2C

Parámetros:

num	Numero de datos a enviar
comando	enviara comando o datos

Devuelve:

none respuesta del dispositivo o el error si no responde Descripción: Retorna el estado de todas las salidas

Parámetros:

none	
------	--

Devuelve:

Estado de las salidas o (-1) si error

Definición en la línea 58 del archivo actuadores.c.

int lee_salida (uint8_t Salida)

Descripción: Retorna el estado de una determinada salida

Parámetros:

Salida	Nº de salida (0-8)
--------	--------------------

Devuelve:

Estado de la salida o (-1) si error

Definición en la línea 82 del archivo actuadores.c.

int Verifica_Driver_salidas ()

Descripción: Verifica comunicación con el driver de salidas

Parámetros:

none	
------	--

Devuelve:

TRUE si correcto, Num Error caso contrario

Descripción: Verifica comunicación con el driver de salidas

Parámetros:

none	
------	--

Devuelve:

(-1) si hay error o Estado salidas en caso contrario

Definición en la línea 159 del archivo actuadores.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/adc.h

Fichero cabecera: (Configuracion canales ADC)

Estructuras de datos

- struct **ADC_channels_data**

'defines'

- #define **ADC_INTERRUPT_FLAG** 0/* 1 is interrupt driven, 0 is polling */
- #define **BURST_MODE** 0 /* Burst mode works in interrupt driven mode only. */
- #define **ADC_DEBUG** 1
- #define **ADC_OFFSET** 0x10
- #define **ADC_INDEX** 4
- #define **ADC_DONE** 0x80000000
- #define **ADC_OVERRUN** 0x40000000
- #define **ADC_ADINT** 0x00010000
- #define **ADC_NUM** 4
- #define **ADC_CLK** 1000000 /* set to 1Mhz */

Funciones

- void **ADCInit** (uint32_t ADC_Clk)
- uint32_t **ADCRead** (uint8_t channelNum)
- void **ADCBurstRead** (void)
- void **LeeAdc** (int Num_Canales, **ADC_channels_data** *Datos)

Descripción detallada

Fichero cabecera: (Configuracion canales ADC)

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

ASIGNATURA SISTEMAS EMBEBIDOS - PAC 3

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera: (Config, envio y recepcion datos a UART)

Fecha:

25/11/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **adc.h**.

Documentación de las funciones

void ADCBurstRead (void)

Funcion que efectua lectura del ADC0 en modo BurstRead

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 219 del archivo adc.c.

void ADCInit (uint32_t *ADC_Clk*)

Funcion ADCInit inicializa canal ADC indicado

Parámetros:

<i>ADC</i>	clock rate
------------	------------

Devuelve:

none

Definición en la línea 103 del archivo adc.c.

uint32_t ADCRead (uint8_t *channelNum*)

Funcion ADCRead Efectua lectura del canal ACD indicado en parametros

Parámetros:

<i>ChannelNum</i>	Numero del canal a leer
-------------------	-------------------------

Devuelve:

Valor de la lectura si esta en modo interrupcion retorna el canal

Definición en la línea 176 del archivo adc.c.

void LeeAdc (int *Num_Canales*, ADC_channels_data * *Datos*)

Funcion que efectua lectura del DAC y envia los datos a la cola

Parámetros:

<i>Num_Canales</i>	Numero de canales a muestrear
<i>DatosADC</i>	Structura con los datos de los 3 canales ADC utilizados

Devuelve:

none

Definición en la línea 239 del archivo adc.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/btw.h

Fichero cabecera: (Libreria funciones dispositivo BTW H05)

'defines'

- #define **ON** (1)
- #define **OFF** (0)

Funciones

- void **BtwDelayMs** (int ms)
- void **BtwCleanBuffers** ()
- uint32_t **BtwRx** (char *String, uint32_t Timeout)
- uint32_t **BtwLogResponse** (int TiempoEspera, int Imprime)
- uint32_t **BtwComand** (char *Comando)
- uint32_t **BtwOK** (void)
- uint32_t **BtwTest_AT** (void)
- uint32_t **BtwReset** (void)
- char * **BtwVersion** (void)
- uint32_t **BtwResetFactory** (void)
- char * **BtwVerConfig** ()
- uint32_t **BtwInit** (int On_Debug)
- void **BtwSleep** (char *addr_disp)
- char * **BtwChangeConfigUART** (uint32_t baudRate, int bit_Stop, int Flow_Control)
- void **BtwConfigSlave** (int Modo_Slave)
- uint32_t **BtwConect** (char *Bind, char *Pswd, int Modo, int t_esp)

Descripción detallada

Fichero cabecera: (Libreria funciones dispositivo BTW H05)

Fichero cabecera: (Libreria funciones dispositivo TMP101)

Fichero cabecera: (Libreria funciones dispositivo serial)

Fichero cabecera: (Libreria funciones rtc)

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera: **Btw.h** (Libreria funciones dispositivo BTW H05)

Fecha:

10/10/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera: **serial.h** (Libreria funciones manejo serial LPC1760)

Fecha:

10/10/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera: **tmp101.h** (Libreria funciones dispositivo tmp101) Sensor de Temperatura bus I2C

Fecha:

10/10/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **btw.h**.

Documentación de las funciones

char* BtwChangeConfigUART (uint32_t *baudRate*, int *bit_Stop*, int *Parity*)

Funcion "BtwChangeBaudRate" cambia el baud Rate del dispositivo Btw y el UART asignado para respuestas.

Parámetros:

<i>baudRate</i>	velocidad puerto UART dispositivo Btw
<i>bit_Stop</i>	bit de stop 0 = 1 bit stop 1= 2 bit Stop
<i>Paridad</i>	(0) sin bit de paridad (1) bit de paridad

Devuelve:

Parametros cadena con la nueva configuracion del cambio efectuado
Definición en la línea 389 del archivo btw.c.

void BtwCleanBuffers (void)

Funcion "BtwCleanBuffers": Para vaciado de los Buffers TX, RX, LOG

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 75 del archivo btw.c.

uint32_t BtwComand (char * *Comando*)

Funcion "BtwComand" envia comando al dispositivo. Se hace uso del semaforo para garatizar exclusion mutua

Parámetros:

<i>Comando</i>	= Buffer con el comando que deseamos enviar a través del UART.
----------------	--

Devuelve:

valor 0 si se ejecuta correctamente

Definición en la línea 116 del archivo btw.c.

uint32_t BtwConect (char * *Bind*, char * *Pswd*, int *Modo*, int *t_esp*)

Funcion "BtwConect" efectua grabacion de la configuracion del dispositivo y lo resetea.

Parámetros:

<i>Bind</i>	direccion MAC dispositivo a conectar
<i>Pswd</i>	Clave para sincronizacion dispositivo
<i>Modo</i>	0 indicara modo conexion con dispositivo especifico, 1 conexion con cualquier dispositivo al alcance
<i>temp_esp</i>	tiempo maximo espera para conexion

Devuelve:

TRUE si se efectua la conexion FALSE si falla

Definición en la línea 314 del archivo btw.c.

void BtwConfigSlave (int *Modo_Slave*)

Funcion "BtwConect_FTP" realiza una conexion FTP y lee o envia un fichero

Parámetros:

<i>Modo_Slave</i>	0=Modo Slave 2 = modo Slave-Loop
-------------------	----------------------------------

Devuelve:

none

Definición en la línea 421 del archivo btw.c.

void BtwDelayMs (int *ms*)

Descripcion: Funcion "BtwDelayMs" Introduce un tiempo de espera en milisegundos

Parámetros:

<i>ms</i>	= Num. milisegundos de retardo.
-----------	---------------------------------

Devuelve:

none

Definición en la línea 62 del archivo btw.c.

uint32_t BtwInit (int *On_Debug*)

Funcion "BtwInit" Inicializa el dispositivo Btw en el sistema Este debe ser llamado antes de crear las tareas.

Parámetros:

<i>On_Debug</i>	(0,1) activa o desactiva modo depuracion errores
-----------------	--

Devuelve:

1 si inicializa correctamente 0 si hay error

Definición en la línea 255 del archivo btw.c.

uint32_t BtwLogResponse (int *TiempoEspera*, int *Imprime*)

Funcion "BtwLogResponse": Presenta las respuesta del modulo en el UART indicado

Parámetros:

<i>TiempoEspera</i>	= Tiempo que escuchamos la respuesta.
<i>Imprime</i>	1 imprime respuesta Btw en UART 0 no la imprime

Devuelve:

Retorna 0 si se ejecuta correctamente

Definición en la línea 105 del archivo btw.c.

uint32_t BtwOK (void)

Funcion "BtwOK" verifica si la respuesta del dispositivo es correcta debe utilizarse inmediatamente despues de la recepcion

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

TRUE si respuesta OK FALSE si respuesta error

Definición en la línea 133 del archivo btw.c.

uint32_t BtwReset (void)

Funcion para realizacion de reset del dispositivo Btw mediante el pin 24 del LPC1769

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 183 del archivo btw.c.

uint32_t BtwResetFactory (void)

Funcion para realizacion de reset del dispositivo Btw mediante software

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 197 del archivo btw.c.

uint32_t BtwRx (char * *String*, uint32_t *Timeout*)

Función "BtwRx": Recibe respuestas a los comandos recibidos por el modulo Btw almacenados en el buffer

Parámetros:

<i>String</i>	= buffer caracteres recibidos por el UART0Buffer.
<i>Timeout</i>	= Tiempo que esperamos para recibir la respuesta.

Devuelve:

NewCount = Num. caracteres recibidos.

Definición en la línea 88 del archivo btw.c.

void BtwSleep (char * *addr_disp*)

Función "BtwSleep" pone el modulo Btw en modo Sleep

-

- **Parámetros:**

<i>addr</i>	Dirección MAC dispositivo bluetooth
-------------	-------------------------------------

- **Devuelve:**

none

-

Definición en la línea 355 del archivo btw.c.

uint32_t BtwTest_AT (void)

Función "BtwConsole": Entra en modo comando al dispositivo Btw Se debe hacer uso del semaforo creado para no enviar comandos simultaneamente al dispositivo.

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

valor 0 si se ejecuta correctamente

Definición en la línea 150 del archivo btw.c.

char* BtwVerConfig (void)

Función para realización visualizar los parametros del dispositivo: Dirección, Class, Nombre, PIN, Modo Funcionamiento.

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

Cadena con los Parametros indicados

Definición en la línea 227 del archivo btw.c.

char* BtwVersion (void)

Función "BtwVersion" envia comando para consulta de la version del dispositivo.

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 212 del archivo btw.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Fonts/Comic5x8.h

Fichero fuente para GLCD 128x54 fuente Comic5x8.

Descripción detallada

Fichero fuente para GLCD 128x54 fuente Comic5x8.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14
UNIVERSIDAD OBERTA DE CATALUÑA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fecha:

10/10/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **Comic5x8.h**.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Fonts/font5x8.h

Fichero fuente para GLCD 128x54 fuente 5x8.

Descripción detallada

Fichero fuente para GLCD 128x54 fuente 5x8.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14
UNIVERSIDAD OBERTA DE CATALUÑA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fecha:

10/10/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **font5x8.h**.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/gps.h

Fichero cabecera:(Recepcion de datos GPS via bluetooth)

Funciones

- int **ConfigConect** (char *Bind, char *Pswd, int t_esp)
- int **Gps_con_des** (int conex)
- int **Recibe_datos_gps** (char GPS_Datos[10][125], int temp_esp)

Descripción detallada

Fichero cabecera:(Recepcion de datos GPS via bluetooth)

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14
UNIVERSIDAD OBERTA DE CATALUÑA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera: **gps.h** (Recepcion de datos GPS via bluetooth)

Fecha:

22/11/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **gps.h**.

Documentación de las funciones

int **ConfigConect** (char * *Bind*, char * *Pswd*, int *t_esp*)

Funcion "ConfigConect" efectua configuracion para la conexion con GPS

Parámetros:

<i>Bind</i>	direccion MAC dispositivo GPS
<i>Pswd</i>	Clave para sincronizacion dispositivo
<i>temp_esp</i>	tiempo maximo espera para conexion

Devuelve:

TRUE si se efectua la conexion FALSE si falla

Definición en la línea 54 del archivo gps.c.

int Gps_con_des (int conex)

Funcion "Gps_con_des efectua conexion o desconexion del dispositivo GPS

Parámetros:

<i>conex</i>	(0= Desconecta) (1=Conecta)
--------------	-----------------------------

Devuelve:

TRUE si se efectua la conexion FALSE si falla

Definición en la línea 108 del archivo gps.c.

int Recibe_datos_gps (char *GPS_Datos*[10][125], int *temp_esp*)

Funcion "Gps_con_des efectua conexion o desconexion del dispositivo GPS

Parámetros:

<i>temp_esp</i>	tiempo espera para recepcion de datos (ms)
-----------------	--

Devuelve:

devuelve datos o Null

Funcion Recibe_datos_pps realiza la recepcion de datos del dispositivo gps extrayendolos en array para su posterior tratamiento

Parámetros:

<i>GPS_Datos</i>	Array con datos GPS,
<i>int</i>	tiempo espera para recepcion de datos

Devuelve:

TRUE si se ejecuta correctamente

Definición en la línea 129 del archivo gps.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/lcd.h

Fichero cabecera:(libreria control dispositivo Eeprom 24LXXX).

```
#include "LPC17xx.h"
#include "timer.h"
#include "Config_sis.h"
```

'defines'

- #define **SetPuerto** FIODIR
- #define **High** FIOSET
- #define **Down** FIOCLR
- #define **GetPuerto** FIOPIN
- #define **Enmarcara** FIOMAS
- #define **Port_Datos** LPC_GPIO2
- #define **Port_Ctr** LPC_GPIO0
- #define **Port_BL** LPC_GPIO4
- #define **LCD_BACKLIGHT** (1 << 29)
- #define **LCDRS** (1 << 22)
- #define **LCDEN** (1 << 27)
- #define **LCDRW** (1 << 28)
- #define **LCD_D4** (1 << 5)
- #define **LCD_D5** (1 << 6)
- #define **LCD_D6** (1 << 7)
- #define **LCD_D7** (1 << 8)
- #define **lcdDataPins** 5
- #define **LCD_DATA_MASK** (LCD_D4 | LCD_D5 | LCD_D6 | LCD_D7)
- #define **LCD_CTR_MASK** (LCDRS | LCDEN | LCDRW)
- #define **LCD_CTRL_MASK** (~(0x0F << lcdDataPins))
- #define **MASK_NB_H** (LCD_D6 | LCD_D7)
- #define **MASK_NB_L** (LCD_D4 | LCD_D5)
- #define **LCD_BUSY_FLAG** (LCD_D7)
- #define **Lcd_BackLinght_On** Port_BL->High=(LCD_BACKLIGHT)
- #define **Lcd_BackLinght_Off** Port_BL->Down=(LCD_BACKLIGHT)
- #define **LCD_display_control** 0x08
- #define **LCD_cursor_shift** 0x10
- #define **LCD_Funcion_set** 0x20
- #define **LCD_Set_CGRam_ADDR** 0x40
- #define **LCD_Set_DDRam_ADDR** 0x80
- #define **Borrar** 0x01
- #define **Ret_Home** 0x02
- #define **CursorIncr** 0x06
- #define **CursorDec** 0x04
- #define **Cursor_On** 0x0F
- #define **Cursor_Off** 0x08
- #define **CursorON** 0x0E
- #define **Blink_ON** 0x0D
- #define **Cursor_BI** 0x0F
- #define **No_BlinkCur** 0x0C
- #define **LCD_Entrada_Right** 0x00

- **#define LCD_Entrada_Left** 0x02
- **#define LCD_Entrada_Shift_Increment** 0x01
- **#define LCD_Entrada_Shift_Decrement** 0x00
- **#define Pos_Linea1** 0x80
- **#define Pos_Linea2** 0xC0
- **#define PosicionLinea1** 0x00
- **#define PosicionLinea2** 0x40
- **#define PosicionLinea3** 0x14
- **#define PosicionLinea4** 0x54
- **#define LCD_2Lineas8Bits5x7** 0x38
- **#define LCD_2Lineas8Bits5x10** 0x3C
- **#define LCD_2Lineas4Bits5x7** 0X28
- **#define LCD_2Lineas4Bits5x10** 0X2C
- **#define LCD_1Lineas8Bits5x7** 0x30
- **#define LCD_1Lineas8Bits5x10** 0x34
- **#define LCD_Set_CGRAM** 0x40
- **#define LCD_Set_DDRAM** 0x80
- **#define LCD_funcion_set_4bit** 0x20
- **#define LCD_funcion_set_8bit** 0x30
- **#define LCD_Entrada_modoset** 0x06
- **#define Max_char_line** 16
- **#define Lcd_Back_Light_timeout** 1000
- **#define LCD_CLEAR** 0x01
- **#define LCD_RETHOM** 0x02
- **#define LCD_LINE1** 0x80
- **#define LCD_LINE2** 0xC0
- **#define LCD_DDRAM** 0x80
- **#define LCD_CGRAM** 0x40
- **#define LCD_CURSOR** 0x0E
- **#define LCD_BLINK** 0x0D
- **#define LCD_CURBLK** 0x0F
- **#define LCD_NOCURBLK** 0x0C

Funciones

- void **lcd_init** (void)
- void **lcd_putstr** (unsigned char line, char *string)
- void **lcd_puts** (int linea, char *s)
- int **lcd_gotoxy** (unsigned int x, unsigned int y)
- void **lcd_clear** (void)
- void **lcd_data** (char dat)
- int **lcd_i2c_init** (uint8_t Bus_I2c, uint8_t lcd_Addr, uint8_t lcd_cols, uint8_t lcd_lin)
- void **lcd_i2c_gotoxy** (unsigned int x, unsigned int y)
- void **lcd_i2c_printf** (int linea, char *s)
- void **lcd_i2c_printf_xy** (int linea, int pos, char *s)
- void **lcd_i2c_clear** (void)
- int **lcd_i2c_cmd** (uint8_t com)
- int **lcd_i2c_data** (char dat)

Descripción detallada

Fichero cabecera:(libreria control dispositivo Eeprom 24LXXX.

Fichero C:(libreria control dispositivo Eeprom 24LXXX.

Fichero cabecera:(libreria manejo LCD 1602 /4 bit)

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera:(funciones manejo LCD 1602 configuracion 4bit)

Fecha:

25/10/2013

Nota:

Copyright Modificacion del obtenido con la asignatura para manejo de los 4 UART.

License: Se permiten el uso y la redistribuci3n de este c3digo con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera:(funciones manejo LCD 1602 configuracion 4bit) pudiendo este estar conectado a puerto o mediante expansor I/O PCF8574 al bus I2C

Fecha:

25/10/2013

Nota:

Copyright Modificacion del obtenido con la asignatura para manejo de los 4 UART.

License: Se permiten el uso y la redistribuci3n de este c3digo con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera:(funciones manejo dispositivos Eeprom 24LXXX para LPC 1769

Fecha:

06/11/2013

Nota:

Copyright Modificacion del obtenido con la asignatura para manejo de los 4 UART.

License: Se permiten el uso y la redistribuci3n de este c3digo con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.
Definici3n en el archivo **lcd.h**.

Documentaci3n de las funciones

void lcd_i2c_clear (void)

Limpia la pantalla del LCD y regresa el cursor a la primera posici3n

Par3metros:

<i>none</i>	
-------------	--

Devuelve:

none

Limpia la pantalla del LCD y regresa el cursor a la primera posici3n de la l3nea 1.

Definici3n en la l3nea 452 del archivo lcd.c.

int lcd_i2c_cmd (uint8_t *com*)

Env3an instrucciones de comando al LCD.

Par3metros:

<i>com</i>	Comando del LCD
------------	-----------------

Devuelve:

Respuesta del dispositivo o el Error si no responde

Definici3n en la l3nea 463 del archivo lcd.c.

int lcd_i2c_data (char *dat*)

Efectua escritura de caracter en DDRAM/CGRAM

Par3metros:

<i>dat</i>	caracter con dato para escribir en DDRM/CGRAM
------------	---

Devuelve:

Respuesta del dispositivo o el Error si no responde

Definici3n en la l3nea 475 del archivo lcd.c.

void lcd_i2c_gotoxy (unsigned int *x*, unsigned int *y*)

Ubica el cursor del LCD en la columna *x* de la l3nea *y*.

Par3metros:

	<i>x</i> posicion dentro de la linea
	<i>y</i> linea a posicionarse

Devuelve:

none

Ubica el cursor del LCD en la columna *c* de la l3nea *r*.

•

• **Devuelve:**

none

-

Definición en la línea 384 del archivo lcd.c.

int lcd_i2c_init (uint8_t *Bus_I2c*, uint8_t *lcd_Addr*, uint8_t *lcd_cols*, uint8_t *lcd_lin*)

Funcion "lcd_2ec_init inicializa el LCD

Parámetros:

<i>Bus_I2c,Numero</i>	bus I2C al que esta conectado el dispositivo
<i>lcd_Addr</i>	direccion del dispositivo LCD (I2C)
<i>Numero</i>	de caracteres por linea
<i>lcd_lin</i>	Numero de lineas del dispositivo

Devuelve:

Respuesta del dispositivo o el Error si no responde

Funcion "lcd_2ec_init inicializa el LCD

Parámetros:

<i>Bus_I2c,Numero</i>	bus I2C al que esta conectado el dispositivo
<i>lcd_Addr</i>	direccion del dispositivo LCD (I2C)
<i>Numero</i>	de caracteres por linea
<i>lcd</i>	rows Numero de lineas del dispositivo

Devuelve:

Respuesta del dispositivo o el Error si no responde

Definición en la línea 317 del archivo lcd.c.

void lcd_i2c_printf (int *linea*, char * *s*)

Funcion "lcd_printf": Imprime linea de caracteres en LCD si sobre pasa de medida cambiara de linea, y si agota la linea limpiara pantalla y continuara escritura tras retardo para dando tiempo a lectura.

Parámetros:

<i>s</i>	Cadena de caracteres
<i>int</i>	NÂ° linea donde inicia la escritura.

Devuelve:

none

Funcion "lcd_printf": Imprime linea de caracteres en LCD

Parámetros:

<i>s</i>	Cadena de caracteres
<i>int</i>	NÂ° linea donde escribe

Devuelve:

none

Definición en la línea 396 del archivo lcd.c.

void lcd_i2c_printf_xy (int *linea*, int *pos*, char * *s*)

Funcion "lcd_printf_xy": Imprime linea de caracteres en LCD

Parámetros:

<i>s</i>	Cadena de caracteres
<i>int</i>	NÂ° linea donde escribe

Devuelve:

none

Definición en la línea 423 del archivo lcd.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/lcdg_i2c.h

Fichero cabecera:(libreria manejo LCD grafico 128x64 (192x64) por bus I2C mediante dispositivo BV4611)

'defines'

- #define **G_RS** (0x1)
- #define **G_RW** (0x2)
- #define **G_EN** (0x4)
- #define **G_CS2** (0x8)
- #define **G_CS1** (0x10)
- #define **G_RST** (0x20)
- #define **G_XX1** (0x40)
- #define **G_XX2** (0x80)
- #define **D0** (0x1)
- #define **D1** (0x2)
- #define **D2** (0x4)
- #define **D3** (0x8)
- #define **D4** (0x10)
- #define **D5** (0x20)
- #define **D6** (0x40)
- #define **D7** (0x80)
- #define **LCDG_ON_OFF_MASK** (D7 | D6 | D4 | D3 | D2 | D1 | D0)
- #define **LCDG_RESET_MASK** (D7 | D6 | D5 | D3 | D2 | D1 | D0)
- #define **LCDG_BUSY_MASK** (D6 | D5 | D4 | D3 | D2 | D1 | D0)
- #define **LCDG_DATA_MASK** (D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0)
- #define **LCDG_CTRL_MASK** (G_XX2|G_XX2|G_RST|G_CS1|G_CS2|G_EN|G_RW|G_RS)
- #define **Lcdg_Busy** (0x80)
- #define **Est_On_Off** (0x20)
- #define **Est_Reset** (0x10)
- #define **Glcd_Left** 0
- #define **Glcd_Right** 1
- #define **On** 1
- #define **Off** 0
- #define **FALSE** 0
- #define **TRUE** 1
- #define **RELLENO** 1
- #define **VACIO** 0
- #define **JUST_IZQ** 1
- #define **JUST_DER** 2
- #define **JUST_CENTRADO** 3
- #define **GLCD_USE_MODE_FAST**
- #define **GLCD_USE_RW**
- #define **BUFSIZE_LCD** 256
- #define **LCDG_Encendido** 0x3F
- #define **LCDG_Apagado** 0x3E
- #define **LCD_G_set_pos_X** 0xB8
- #define **LCD_G_set_pos_Y** 0x40
- #define **LCD_G_Busy** 0x80
- #define **Inicio_linea** 0xC0

Funciones

- int **lcdg_write** (uint8_t dato, int side, int comando)
- int **lcdg_actualiza** ()
- void **Lcdg_linea** (uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2, int borrar, int actualiza)
- void **lcdg_punto** (uint8_t x, uint8_t y, int borrar, int actualiza)
- void **Lcdg_Circulo** (uint8_t cx, uint8_t cy, uint8_t radius, int Relleno, int borrar, int actual)
- void **Lcdg_Rectangulo** (uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2, int Relleno, int borrar, int actual)
- void **Lcdg_Image** (unsigned char x, unsigned char y, unsigned char LargoX, unsigned char LargoY, char Data, int borrar, int actualiza)
- void **Lcdg_Image2** (unsigned char x, unsigned char y, const unsigned char *Imagen, int borrar, int actualiza)
- int **SET_FontSize** (int Size)
- int **SET_Inverse** (int inv)
- int **GET_Inverse** (void)
- int **Reset** ()
- int **Cls** ()
- int **Cursor_HOME** ()
- int **SET_CursorPixel** (uint8_t x, uint8_t y)
- int **Cursor** (int estado)
- int **Lcdg_On_Off** (int estado)
- int **Lcdg_BackLight** (int estado)
- unsigned char **lcdg_Lee_caracter_ROM** (char *ptr)
- void **Lcdg_Write_Text** (uint8_t x, uint8_t Lin, char *StringTexto, int fuente, int color, int actualizar)
- int **Lcd_init** (int N_Bus_I2C, int Dir, int On_Off)
- int **GET_FirmwareVersion** ()

Descripción detallada

Fichero cabecera:(libreria manejo LCD grafico 128x64 (192x64) por bus I2C mediante dispositivo BV4611)

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera:(funciones manejo LCDG grafico 128x64 (192x64) 1602 configuracion 4bit)

Fecha:

25/10/2013

Nota:

Copyright Modificacion del obtenido con la asignatura para manejo de los 4 UART.

License: Se permiten el uso y la redistribuci3n de este c3digo con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definici3n en el archivo **lcdg_i2c.h**.

Documentación de las funciones

int Cls ()

Descricion: Borra pantalla secuencia "esc0x1B0[2J"ACK"\[2J"

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Descricion: Borra pantalla

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 496 del archivo lcdg_i2c.c.

int Cursor (int estado)

Descricion: Cursor Presenta o quita el cursor del GLCD

Parámetros:

<i>estado</i>	(ON=1) pone cursor (ON=0)quita el cursor
---------------	--

Devuelve:

none

Definición en la línea 600 del archivo lcdg_i2c.c.

int Cursor_HOME ()

Descricion: Cursor Home secuencia Setpage(0) Setaddres(0)

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 563 del archivo lcdg_i2c.c.

int GET_FirmwareVersion ()

Descricion: Version del Firmware

Parámetros:

<i>none</i>	
<i>none</i>	

Devuelve:

none

Definición en la línea 843 del archivo lcdg_i2c.c.

int GET_Inverse (void)

Descricion: Devuelve el valor de Inversion de la pantalla

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 467 del archivo lcdg_i2c.c.

int Lcd_init (int *N_Bus_I2C*, int *Dir*, int *On_Off*)

Descripción: Lcd_init Inicializa el dispositivo grafico LCD

Parámetros:

<i>N_SET_Bus_I2C</i>	Numero bus donde esta conectado el dispositivo (0,1,2), modo (apagado=0, encendido=1)
<i>Dir</i>	Dirección del dispositivo LCD

Devuelve:

none

Definición en la línea 787 del archivo lcdg_i2c.c.

int lcdg_actualiza ()

Descripción: Actualiza la imagen de la pantalla en array para el uso del modo FAST MODEEnvía comando

Parámetros:

<i>none</i>	
<i>side</i>	selección de pantalla 0, CS1, 1 CS2, otro Ninguna
<i>Rs</i>	0 = comando 1=Dato

Devuelve:

none

int Lcdg_BackLight (int *estado*)

Descripción: Lcdg_Backlight Mantiene encendida o apaga luz en el GLCD

Parámetros:

<i>estado</i>	(ON=1) mantiene luz (ON=0) apaga luz del GLCD
---------------	---

Devuelve:

none

Definición en la línea 621 del archivo lcdg_i2c.c.

void Lcdg_Circulo (uint8_t *cx*, uint8_t *cy*, uint8_t *radius*, int *Relleno*, int *borrar*, int *actual*)

Descripción: Circulo secuencia

Parámetros:

<i>x,y</i>	coordenadas del centro
<i>Radius</i>	(pixel de radio del circulo)
<i>Borrar</i>	(1=borrar circulo) (0=Escribir circulo)
<i>Actualiza</i>	(1=Actualiza pantalla) (0=solo cambia en buffer) para posterior actualización si se esta graficando un imagen completa.

Devuelve:

none

Definición en la línea 299 del archivo lcdg_i2c.c.

void Lcdg_Image (unsigned char x, unsigned char y, unsigned char LargoX, unsigned char LargoY, char Data, int borrar, int actualiza)

Descripcion: Carga una imagen en el GLCD

Parámetros:

<i>x,y</i>	coordenadas del vertice superior izquierdo
<i>LargoX</i>	ancho de la imagen
<i>LargoY</i>	alto de la imagen
<i>Data</i>	Imagen a cargar

Devuelve:

none

void Lcdg_Image2 (unsigned char x, unsigned char y, const unsigned char * Imagen, int borrar, int actualiza)

Descripcion: Carga una imagen en el GLCD

Parámetros:

<i>x,y</i>	coordenadas del vertice superior izquierdo
<i>LargoX</i>	ancho de la imagen
<i>LargoY</i>	alto de la imagen
<i>Data</i>	Imagen a cargar

Devuelve:

none

Definición en la línea 400 del archivo lcdg_i2c.c.

void Lcdg_linea (uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2, int borrar, int actualiza)

Descripcion: Representa o borra una línea en el LCD

Parámetros:

<i>x1</i>	y1 coornedaos inicio linea
<i>x2</i>	y3 coordenadas final linea
<i>borrar</i>	p_color color linea
<i>actualiza</i>	1=Actualiza el LCD, 0=Solo actualiza el buffer para el una posterior actualizacion si se esta graficando una imagen

Devuelve:

none

Descripcion: Actualiza la imagen de la pantalla en array para el uso del modo FAST MODEEnvia comando

Parámetros:

<i>none</i>	
<i>side</i>	selecccion de pantalla 0, CS1, 1 CS2, otro Ninguna
<i>Rs</i>	0 = comando 1=Dato

Devuelve:

none Descripcion: Representa o borra una línea en el LCD

Parámetros:

<i>x1</i>	y1 coornedaos inicio linea
<i>x2</i>	y3 coordenadas final linea
<i>borrar</i>	p_color color linea
<i>actualiza</i>	1=Actualiza el LCD, 0=Solo actualiza el buffer para el una posterior actualizacion si se esta graficando una imagen

Devuelve:

none

Definición en la línea 186 del archivo lcdg_i2c.c.

int Lcdg_On_Off (int *modo*)

Descripción: Lcdg_activo Apaga o enciende el LCD

Parámetros:

<i>estado</i>	(ON=1) enciende el LCD (ON=0) lo apaga
<i>@return</i>	none

Descripción: Enciende/Apaga el LCD RS RW DB7.....DB0 (L L L L H H H H H H)

Parámetros:

<i>modo</i>	TRUE = Encendido, False = Apagado
-------------	-----------------------------------

Devuelve:

Resultado ok o error del dispositivo.

Definición en la línea 542 del archivo lcdg_i2c.c.

void lcdg_punto (uint8_t x, uint8_t y, int *borrar*, int *actualiza*)

Descripción: Representa o borra un punto en el LCD

Parámetros:

<i>x</i>	y coordenadas del punto
<i>borrar</i>	(1=borra punto) (0=Escribe punto)
<i>Actualiza</i>	(1=Actualiza pantalla) (0=solo cambia en buffer) para posterior actualizacion si se esta graficando un imagen completa.

Devuelve:

none

Definición en la línea 246 del archivo lcdg_i2c.c.

void Lcdg_Rectangulo (uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2, int *Relleno*, int *borrar*, int *actual*)

Descripción: rectangulo

Parámetros:

<i>x1,y1</i>	coordenadas del vertice superior izquierdo
<i>x2,y2</i>	coordenadas vertice inferior derecho
<i>relleno</i>	1=Rectangulo relleno 0= Contorno
<i>Borrar</i>	(1=borrar rectangulo) (0=Escribir rectangulo)
<i>Actualiza</i>	(1=Actualiza pantalla) (0=solo cambia en buffer) para posterior actualizacion si se esta graficando un imagen completa.

Devuelve:

none

Definición en la línea 353 del archivo lcdg_i2c.c.

int lcdg_write (uint8_t *dato*, int *side*, int *comando*)

Descripción: Envia comando al dispositivo GLCD bien un comando o dato

Parámetros:

<i>comando</i>	a enviar
<i>side</i>	seleccción de sección pantalla 0=Izquierda (CS1) 1=Derecha(CS2) Otro

	Ninguna (para envio en modo dato)
<i>comando</i>	1 para envio de comando 0 para envio de dato

Devuelve:

none

Definición en la línea 124 del archivo lcdg_i2c.c.

void Lcdg_Write_Text (uint8_t x, uint8_t Lin, char * *StringTexto*, int *fuelle*, int *Color*, int *actualizar*)

Descricion: Envia Cadena de caracteres

Parámetros:

<i>x,y</i>	Coordenadas x e y poscion inicio texto
<i>StrintTexto</i>	Texto a presentar
<i>int</i>	fuelle uso de la fuente de caracteres a utilizar 1=font5x8
<i>actualizar</i>	Actualiza la pantalla o solo el buffer para posterior presentacion del texto.

Devuelve:

none

Descricion: Envia una cadena de caracteres al GLCD

Parámetros:

<i>text</i>	Cadena de caracteres
<i>x_pos</i>	posiciones x del GLCE
<i>ptr</i>	fuelle puntero a la fuente de caracteres ROM
<i>fuelle</i>	de caracteres 1= font5x8;

Devuelve:

none

Definición en la línea 755 del archivo lcdg_i2c.c.

unsigned char lcgd_Lee_caracter_ROM (char * *ptr*)

Descricion: Envia una cadena de caracteres al GLCD

Parámetros:

<i>text</i>	Cadena de caracteres
<i>x_pos</i>	posiciones x del GLCE
<i>ptr</i>	fuelle puntero a la fuente de caracteres ROM
<i>Justificado</i>	1=Izquierda 2=Centro 3=Derecha

Devuelve:

none Descricion: Lee byte de la memoria ROM

Parámetros:

<i>char</i>	Byte a leer
-------------	-------------

Devuelve:

none

Descricion: Lee byte de la memoria ROM

Parámetros:

<i>char</i>	Byte a leer
-------------	-------------

Devuelve:

none

Definición en la línea 657 del archivo lcdg_i2c.c.

int Reset ()

Descripción: Resetea el dispositivo Poniendo bit RST a 0 y reponiendolo a 1

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

Resultado operacion o el error

Definición en la línea 476 del archivo lcdg_i2c.c.

int SET_CursorPixel (uint8_t x, uint8_t y)

Descripción: Situa el cursos en un pixel determinado

Parámetros:

<i>x,y</i>	coordenadas donde situar el cursor
------------	------------------------------------

Devuelve:

none

Descripción: Situa el cursos en un pixel determinado

Parámetros:

<i>x,y</i>	coordenadas donde situar el cursor
------------	------------------------------------

Devuelve:

Devuelve la seccion del LCD donde se esta el pixel

Definición en la línea 577 del archivo lcdg_i2c.c.

int SET_FontSize (int Size)

Descripción: Selección de fuente Resetea el dispositivo LCD secuencia "\\(x"

Parámetros:

<i>int</i>	Fuente valores entre (1,2,3)
------------	------------------------------

Devuelve:

none

Definición en la línea 637 del archivo lcdg_i2c.c.

int SET_Inverse (int inv)

Descripción: Inversa secuencia

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 455 del archivo lcdg_i2c.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/MMA7361.h

Fichero cabecera: (Driver dispositivo acelerometro MMA7361)

Estructuras de datos

- struct **ACL_Data**

Funciones

- int **Init_Acelerometro** (int Rango)
- void **SelfTestAcelerometro** (void)
- void **AjusteAcelerometro** (int Num_Muestras)
- void **Read_Acelerometro** (int Num_muestreo, **ACL_Data** *Datos)

Descripción detallada

Fichero cabecera: (Driver dispositivo acelerometro MMA7361)

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

ASIGNATURA SISTEMAS EMBEBIDOS - PRACTICA

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera: (Config, lectura canales ADC)

Fecha:

02/12/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **MMA7361.h**.

Documentación de las funciones

void **AjusteAcelerometro** (int *Num_Muestras*)

Funcion AjusteAcelerometro realiza el ajuste de error en medidas del dispositivo acelerometro

Parámetros:

<i>Num_Muestras</i>	Nmero de medidas para calculo del error
---------------------	---

Devuelve:

none

Definición en la línea 125 del archivo mma7361.c.

int Init_Acelerometro (int *Rango*)

Funcion Init_Acelerometro Inicializa el dispositivo Acelerometro MMA7361

Parámetros:

<i>Rango</i>	(1= Rango 206 mg/V 0=Rango 800mg/V)On_Debug (0,1)
<i>activa</i>	o desactiva modo depuracion errores

Devuelve:

none

Definición en la línea 69 del archivo mma7361.c.

void Read_Acelerometro (int *Num_muestreo*, *ACL_Data* * *Datos*)

Funcion Efectua lectura ejes acelerometro devolviendo el resultado en la medida del ADC y en grados de los ejes

Parámetros:

<i>Num_muestreo</i>	Numero de muestras a tomar
<i>ACL_Data</i>	Structura donde se recibirán los datos

Devuelve:

none

Definición en la línea 149 del archivo mma7361.c.

void SelfTestAcelerometro (void)

Funcion AjusteAcelerometro realiza el ajuste de error en medidas del dispositivo acelerometro

Parámetros:

<i>Num_Muestras</i>	Nmero de medidas para calculo del error
---------------------	---

Devuelve:

none Funcion SelfTest Acelerometro Efectua Test del dispositivo para lo que hay que colocar el dispositivo boca abajo y efectuar la llamada a la funcion nos presentara las medidas.

Parámetros:

<i>Num_Muestras</i>	Nmero de medidas para calculo del error
---------------------	---

Devuelve:

none

Funcion SelfTest Acelerometro Efectua Test del dispositivo para lo que hay que colocar el dispositivo boca abajo y efectuar la llamada a la funcion nos presentara las medidas.

Parámetros:

<i>Num_Muestras</i>	Nmero de medidas para calculo del error
---------------------	---

Devuelve:

none

Definición en la línea 94 del archivo mma7361.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/MPL115A.h

Fichero cabecera:(libreria manejo sensor presuin MPL115A por bus I2C)

Funciones

- **int SPresion_init** (int N_Bus_I2C, int Dir, int On_Off)
- **int Lee_pres_temp** (int N_Bus_I2C, int Dir, double BufMed[2])
- **int Lee_coeficientes** (int N_Bus_I2C, int Dir)

Descripción detallada

Fichero cabecera:(libreria manejo sensor presuin MPL115A por bus I2C)

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera:(funciones manejo LCDG grafico 128x64 (192x64) 1602 configuracion 4bit)

Fecha:

25/10/2013

Nota:

Copyright Modificacion del obtenido con la asignatura para manejo de los 4 UART.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **MPL115A.h**.

Documentación de las funciones

int Lee_coeficientes (int N_Bus_I2C, int Dir)

Descricion: Lee_coeficientes Efectua lectura de los coeficientes del dispositivo sensor

Parámetros:

<i>Dir</i>	direccion del dispositivo en bus I2C
<i>BufCoef</i>	[8] Toma medidas de temperatura y de presion

Devuelve:

TRUE si se efectua lectura correcta FALSE si falla

Descricion: Lee_coeficientes Inicializa el dispositivo sensor y verifica comunicacion con el mismo

Parámetros:

<i>BufCoef</i>	[8] coeficientes de Presion y temperatura
----------------	---

Devuelve:

none respuesta del dispositivo o el error si no responde

Definición en la línea 124 del archivo mpl115A.c.

int Lee_pres_temp (int *N_Bus_I2C*, int *Dir*, double *BufMed*[2])

Descricion: Lee_pres_temp Efectua lectura de las medidas de presion y temperatura

Parámetros:

<i>BufMed</i>	[8] coeficientes de Presion y temperatura
---------------	---

Devuelve:

respuesta del dispositivo o el error si no responde

Descricion: Lee_presion_temperatura Inicializa el dispositivo sensor y verifica comunicacion con el mismo

Parámetros:

<i>BufCoef</i>	[8] coeficientes de Presion y temperatura
----------------	---

Devuelve:

TRUE si se efectua lectura correcta FALSE si falla

Definición en la línea 245 del archivo mpl115A.c.

int SPresion_init (int *N_Bus_I2C*, int *Dir*, int *On_Off*)

Descricion: Sensor_init Inicializa el dispositivo sensor y verifica comunicacion con el mismo

Parámetros:

<i>N_SET_Bus_I2C</i>	Numero bus donde esta conectado el dispositivo (0,1,2),
<i>Dir</i>	Direccion del dispositivo LCD
<i>On_Off</i>	modo debug activado=1 o desactivado=0

Devuelve:

none

Descricion: SPresion_init Inicializa el dispositivo sensor y verifica comunicacion con el mismo

Parámetros:

<i>N_SET_Bus_I2C</i>	Numero bus donde esta conectado el dispositivo (0,1,2),
<i>Dir</i>	Direccion del dispositivo LCD
<i>On_Off</i>	modo debug activado=1 o desactivado=0

Devuelve:

respuesta del dispositivo o el error si no responde

Definición en la línea 103 del archivo mpl115A.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/VCNL4010.h

Fichero cabecera:(libreria manejo sensor luminosidad y proximidad VCNL4010A)

'defines'

- #define **Com_Config_reg** 0x80
- #define **RegCommand** 0x80
- #define **Reg_Config_lux** 0x84
- #define **Com_read_lux** 0x85
- #define **Param_Config_lux** 0xDD
- #define **Com_Lux_Enable** 0x04
- #define **Com_lux_Demand** 0x10
- #define **Com_read_Prox** 0x87
- #define **Com_Prox_Demand** 0x08
- #define **Com_Prox_Enable** 0x02
- #define **lux_Prox_Deman** (0x8 | 0x10)
- #define **Comand_lux_mask** 0x40
- #define **Comand_prox_mask** 0x20

Funciones

- int **SLux_init** (int N_Bus_I2C, int Dir)
- uint32_t **Lee_lux** (int N_Bus_I2C, int Dir)
- uint32_t **Lee_prox** (int N_Bus_I2C, int Dir)

Descripción detallada

Fichero cabecera:(libreria manejo sensor luminosidad y proximidad VCNL4010A)

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera:(funciones manejo LCDG grafico 128x64 (192x64) 1602 configuracion 4bit)

Fecha:

25/10/2013

Nota:

Copyright Modificación del obtenido con la asignatura para manejo de los 4 UART.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **VCNL4010.h**.

Documentación de las funciones

uint32_t Lee_lux (int *N_Bus_I2C*, int *Dir*)

Descripción: Lee_lux Efectua lectura de las medidas de presión y temperatura

Parámetros:

<i>N_Bus_I2C</i>	numero bus del dispositivo
<i>dir</i>	dirección del dispositivo

Devuelve:

Luxes. Medida del nivel de luminosidad en Luxes

Descripción: Lee_lux Efectua lectura de las medidas de presión y temperatura

Parámetros:

<i>N_Bus_I2C</i>	numero bus del dispositivo
<i>dir</i>	dirección del dispositivo

Devuelve:

Luxes. Medida del nivel de luminosidad en Luxes o -1 si error

Definición en la línea 110 del archivo vcnl4010.c.

uint32_t Lee_prox (int *N_Bus_I2C*, int *Dir*)

Descripción: Lee_lux Efectua lectura de las medidas de presión y temperatura

Parámetros:

<i>N_Bus_I2C</i>	numero bus del dispositivo
<i>dir</i>	dirección del dispositivo

Devuelve:

Descripción: Lee_lux Efectua lectura de las medidas de presión y temperatura

Parámetros:

<i>N_Bus_I2C</i>	numero bus del dispositivo
<i>dir</i>	dirección del dispositivo

Devuelve:

Proximidad o -1 si error

Definición en la línea 139 del archivo vcnl4010.c.

int SLux_init (int *N_Bus_I2C*, int *Dir*)

Descripción: SLux_init Inicializa el dispositivo sensor y verifica comunicación con el mismo

Parámetros:

<i>N_SET_Bus_I2C</i>	Numero bus donde está conectado el dispositivo (0,1,2),
<i>Dir</i>	Dirección del dispositivo LCD

Devuelve:

Respuesta del dispositivo

Definición en la línea 83 del archivo vcnl4010.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/inc/Web/webside.h

Fichero cabecera: Para declaracion constantes Paginas Web.

Variables

- const unsigned char **WebSide** []

Descripción detallada

Fichero cabecera: Para declaracion constantes Paginas Web.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

ASIGNATURA SISTEMAS EMBEBIDOS - PRACTICA

Autor: Miguel Angel Iglesias Jimenez

Fichero cabecera: (Config, lectura canales ADC)

Fecha:

02/12/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribuci3n de este c3digo con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definici3n en el archivo **webside.h**.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/adc.c

Fichero funciones control ADC sistema LPC1769.

```
#include "lpc17xx.h"
#include "type.h"
#include "adc.h"
```

'defines'

- #define **_BV(_x_)** (1UL << (_x_))

Funciones

- void **ADCInit** (uint32_t ADC_Clk)
- uint32_t **ADCRead** (uint8_t channelNum)
- void **ADCBurstRead** (void)
- void **LeeAdc** (int Num_Canales, **ADC_channels_data** *Datos)

Variables

- volatile uint32_t **ADCValue** [ADC_NUM]
- volatile uint32_t **ADCIntDone** = 0
- volatile uint32_t **BurstCounter** = 0
- volatile uint32_t **OverRunCounter** = 0

Descripción detallada

Fichero funciones control ADC sistema LPC1769.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

ASIGNATURA SISTEMAS EMBEBIDOS - PAC 3

Autor: Miguel Angel Iglesias Jimenez

Fichero funciones control Canales ADC sistema LPC1769 para lectura de datos de los diversos canales y guardado de las mismas

Fecha:

25/11/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **adc.c**.

Documentación de las funciones

void ADCBurstRead (void)

Funcion que efectua lectura del ADC0 en modo BurstRead

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 219 del archivo adc.c.

void ADCInit (uint32_t *ADC_Clk*)

Funcion ADCInit inicializa canal ADC indicado

Parámetros:

<i>ADC</i>	clock rate
------------	------------

Devuelve:

none

Definición en la línea 103 del archivo adc.c.

uint32_t ADCRead (uint8_t *channelNum*)

Funcion ADCRead Efectua lectura del canal ACD indicado en parametros

Parámetros:

<i>ChannelNum</i>	Numero del canal a leer
-------------------	-------------------------

Devuelve:

Valor de la lectura si esta en modo interrupcion retorna el canal

Definición en la línea 176 del archivo adc.c.

void LeeAdc (int *Num_Canales*, ADC_channels_data * *Datos*)

Funcion que efectua lectura del DAC y envia los datos a la cola

Parámetros:

<i>Num_Canales</i>	Numero de canales a muestrear
<i>DatosADC</i>	Structura con los datos de los 3 canales ADC utilizados

Devuelve:

none

Definición en la línea 239 del archivo adc.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/btw.c

Liberia Libreria para manejo de dispositivo Btw H05.

```
#include "FreeRTOS.h"
#include "semphr.h"
#include "type.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Config_sis.h"
#include "uart.h"
#include "printf.h"
#include "btw.h"
```

Funciones

- void **BtwDelayMs** (int ms)
- void **BtwCleanBuffers** (void)
- uint32_t **BtwRx** (char *String, uint32_t Timeout)
- uint32_t **BtwLogResponse** (int TiempoEspera, int Imprime)
- uint32_t **BtwComand** (char *Comando)
- uint32_t **BtwOK** (void)
- uint32_t **BtwTest_AT** (void)
- uint32_t **BtwReset** (void)
- uint32_t **BtwResetFactory** (void)
- char * **BtwVersion** (void)
- char * **BtwVerConfig** (void)
- uint32_t **BtwInit** (int On_Debug)
- uint32_t **BtwConect** (char *Bind, char *Pswd, int Modo, int t_esp)
- void **BtwSleep** (char *addr_disp)
- char * **BtwChangeConfigUART** (uint32_t baudRate, int bit_Stop, int Parity)
- void **BtwConfigSlave** (int Modo_Slave)

Variables

- char **buffRx** [512]
- char **buffTx** [BUFSIZE]
- char **buffLog** [BUFSIZE]
- char **buffIp** [BUFSIZE]
- char **buffParametros** [BUFSIZE]
- char **Fin_com** [3] = {0x0D, 0x0A, 0x00}
- uint32_t **sizeAsk**
- xSemaphoreHandle **xMutexBtw**

Descripción detallada

Liberia Libreria para manejo de dispositivo Btw H05.

TRABAJO FINAL DE GRADO

Autor: Miguel Angel Iglesias Jimenez

Libreria para manejo dispositivo Bluetooth H05 incluyendo las funciones mas usuales de dicho dispositivo

Fecha:

10/10/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribucion de este codigo con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **btw.c**.

Documentación de las funciones

char* BtwChangeConfigUART (uint32_t *baudRate*, int *bit_Stop*, int *Parity*)

Funcion "BtwChangeBaudRate" cambia el baud Rate del dispositivo Btw y el UART asignado para respuestas.

Parámetros:

<i>baudRate</i>	velocidad puerto UART dispositivo Btw
<i>bit_Stop</i>	bit de stop 0 = 1 bit stop 1= 2 bit Stop
<i>Paridad</i>	(0) sin bit de paridad (1) bit de paridad

Devuelve:

Parametros cadena con la nueva configuracion del cambio efectuado
Definición en la línea 389 del archivo btw.c.

void BtwCleanBuffers (void)

Funcion "BtwCleanBuffers": Para vaciado de los Buffers TX, RX, LOG

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none
Definición en la línea 75 del archivo btw.c.

uint32_t BtwComand (char * *Comando*)

Funcion "BtwComand" envia comando al dispositivo. Se hace uso del semaforo para garantizar exclusion mutua

Parámetros:

<i>Comando</i>	= Buffer con el comando que deseamos enviar a través del UART.
----------------	--

Devuelve:

valor 0 si se ejecuta correctamente

Definición en la línea 116 del archivo btw.c.

uint32_t BtwConect (char * *Bind*, char * *Pswd*, int *Modo*, int *t_esp*)

Funcion "BtwConect" efectua grabacion de la configuracion del dispositivo y lo resetea.

Parámetros:

<i>Bind</i>	direccion MAC dispositivo a conectar
<i>Pswd</i>	Clave para sincronizacion dispositivo
<i>Modo</i>	0 indicara modo conexion con dispositivo especifico, 1 conexion con cualquier dispositivo al alcance
<i>temp_esp</i>	tiempo maximo espera para conexion

Devuelve:

TRUE si se efectua la conexion FALSE si falla

Definición en la línea 314 del archivo btw.c.

void BtwConfigSlave (int *Modo_Slave*)

Funcion "BtwConect_FTP" realiza una conexion FTP y lee o envia un fichero

Parámetros:

<i>Modo_Slave</i>	0=Modo Slave 2 = modo Slave-Loop
-------------------	----------------------------------

Devuelve:

none

Definición en la línea 421 del archivo btw.c.

void BtwDelayMs (int *ms*)

Descripcion: Funcion "BtwDelayMs" Introduce un tiempo de espera en milisegundos

Parámetros:

<i>ms</i>	= Num. milisegundos de retardo.
-----------	---------------------------------

Devuelve:

none

Definición en la línea 62 del archivo btw.c.

uint32_t BtwInit (int *On_Debug*)

Funcion "BtwInit" Inicializa el dispositivo Btw en el sistema Este debe ser llamado antes de crear las tareas.

Parámetros:

<i>On_Debug</i>	(0,1) activa o desactiva modo depuracion errores
-----------------	--

Devuelve:

1 si Inicializa correctamente 0 si hay error

Definición en la línea 255 del archivo btw.c.

uint32_t BtwLogResponse (int *TiempoEspera*, int *Imprime*)

Funcion "BtwLogResponse": Presenta las respuesta del modulo en el UART indicado

Parámetros:

<i>TiempoEspera</i>	= Tiempo que escuchamos la respuesta.
---------------------	---------------------------------------

<i>Imprime</i>	1 imprime respuesta Btw en UART 0 no la imprime
----------------	---

Devuelve:

Retorna 0 si se ejecuta correctamente

Definición en la línea 105 del archivo btw.c.

uint32_t BtwOK (void)

Funcion "BtwOK" verifica si la respuesta del dispositivo es correcta debe utilizarse inmediatamente despues de la recepcion

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

TRUE si respuesta OK FALSE si respuesta error

Definición en la línea 133 del archivo btw.c.

uint32_t BtwReset (void)

Funcion para realizacion de reset del dispositivo Btw mediante el pin 24 del LPC1769

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 183 del archivo btw.c.

uint32_t BtwResetFactory (void)

Funcion para realizacion de reset del dispositivo Btw mediante software

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 197 del archivo btw.c.

uint32_t BtwRx (char * *String*, uint32_t *Timeout*)

FunciÃ³n "BtwRx": Recibe respuestas a los comandos recibidos por el modulo Btw almacenados en el buffer

Parámetros:

<i>String</i>	= buffer caracteres recibidos por el UART0Buffer.
<i>Timeout</i>	= Tiempo que esperamos para recibir la respuesta.

Devuelve:

NewCount = Num. caracteres recibidos.

Definición en la línea 88 del archivo btw.c.

void BtwSleep (char * *addr_disp*)

Funcion "BtwSleep" pone el modulo Btw en modo Sleep

•

- **Parámetros:**

<i>addr</i>	Dirección MAC dispositivo bluetooth
-------------	-------------------------------------

- **Devuelve:**

none

-

Definición en la línea 355 del archivo btw.c.

uint32_t BtwTest_AT (void)

Función "BtwConsola": Entra en modo comando al dispositivo Btw. Se debe hacer uso del semáforo creado para no enviar comandos simultáneamente al dispositivo.

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

valor 0 si se ejecuta correctamente

Definición en la línea 150 del archivo btw.c.

char* BtwVerConfig (void)

Función para realización visualizar los parámetros del dispositivo: Dirección, Clase, Nombre, PIN, Modo Funcionamiento.

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

Cadena con los Parámetros indicados

Definición en la línea 227 del archivo btw.c.

char* BtwVersion (void)

Función "BtwVersion" envía comando para consulta de la versión del dispositivo.

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 212 del archivo btw.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/mma7361.c

Libreria para manejo del dispositivo acelemetro DHT22A Sensor Humedad Realativa y temperatura.

```
#include "lpc17xx.h"
#include "FreeRTOS.h"
#include "semphr.h"
#include "task.h"
#include "type.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "Config_sis.h"
#include "timer.h"
#include "uart.h"
#include "printf.h"
#include "adc.h"
#include "MMA7361.h"
```

'defines'

- #define **_360div2xPI** 57.29577951
- #define **_5000div4096** 1.220703125
- #define **MMA7361_Min_Resolucion** 206
- #define **MMA7361_Max_Resolucion** 800
- #define **MMA7361_Voltaje** 3300.0
- #define **Pasos_DAC** 4095
- #define **MMA7361_ZOUT_1G** 850

Funciones

- int **Init_Acelerometro** (int Rango)
- void **SelfTestAcelerometro** (void)
- void **AjusteAcelerometro** (int Num_Muestras)
- void **Read_Acelerometro** (int Num_muestreo, **ACL_Data** *Datos)

Variables

- const float **ZOUT1G** = 2450
- const float **ZOUT0G** = 1650
- const float **XOUT1G** = 2450
- const float **XOUT0G** = 1650
- const float **YOUT1G** = 2450
- const float **YOUT0G** = 1650
- float **XSelfTest** =0
- float **YSelfTest** =0
- float **ZSelfTest** =0
- float **XError** =0
- float **YError** =0
- float **ZError** =0
- float **z** =0

- float **x** =0
- float **y** =0
- int **Resolucion** =0

Descripción detallada

Libreria para manejo del dispositivo acelemetro DHT22A Sensor Humedad Realativa y temperatura.

Libreria para manejo del dispositivo acelemetro MMA7361.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

ASIGNATURA SISTEMAS EMBEBIDOS PRACTICAS

Autor: Miguel Angel Iglesias Jimenez

Libreria que implementa el driver para manejo del acelerometr MMA7361

Fecha:

02/12/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribuci3n de este c3digo con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definici3n en el archivo **mma7361.c**.

Documentaci3n de las funciones

void AjusteAcelerometro (int *Num_Muestras*)

Funcion AjusteAcelerometro realiza el ajuste de error en medidas del dispositivo acelerometro

Par3metros:

<i>Num_Muestras</i>	Nmero de medidas para calculo del error
---------------------	---

Devuelve:

none

Definici3n en la l3nea 125 del archivo mma7361.c.

int Init_Acelerometro (int *Rango*)

Funcion Init_Acelerometro Inicializa el dispositivo Acelerometro MMA7361

Par3metros:

<i>Rango</i>	(1= Rango 206 mg/V 0=Rango 800mg/V)On_Debug (0,1)
<i>activa</i>	o desactiva modo depuracion errores

Devuelve:

none

Definici3n en la l3nea 69 del archivo mma7361.c.

void Read_Acelerometro (int *Num_muestreo*, *ACL_Data* * *Datos*)

Funcion Efectua lectura ejes acelerometro devolviendo el resultado en la medida del ADC y en grados de los ejes

Parámetros:

<i>Num_muestreo</i>	Numero de muestras a tomar
<i>ACL_Data</i>	Structura donde se recibiran los datos

Devuelve:

none

Definición en la línea 149 del archivo mma7361.c.

void SelfTestAcelerometro (void)

Funcion SelfTest Acelerometro Efectua Test del dispositivo para lo que hay que colocar el dispositivo boca abajo y efectuar la llamada a la funcion nos presentara las medidas.

Parámetros:

<i>Num_Muestras</i>	Nmero de medidas para calculo del error
---------------------	---

Devuelve:

none

Definición en la línea 94 del archivo mma7361.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/rtc.c

Libreria para manejo del RTC.

```
#include "lpc17xx.h"
#include "type.h"
#include "rtc.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Config_sis.h"
```

'defines'

- #define **PCRTC** 9
- #define **CLKEN** 0
- #define **CTCRST** 1
- #define **CCALEN** 4
- #define **RTC_SEG** 0
- #define **RTC_MIN** 1
- #define **RTC_HOR** 2
- #define **RTC_DIA_MES** 3
- #define **RTC_DIA_SEMANA** 4
- #define **RTC_DIA_ANIO** 5
- #define **RTC_MES** 6
- #define **RTC_ANIO** 7
- #define **RTC_TAMANIO** 8

'typedefs'

- typedef unsigned char **u8**
- typedef unsigned int **u16**
- typedef unsigned long int **u32**

Funciones

- void **configurar_hora_rtc** (u16 vector_hora_inicial[])
- void **configurar_alarmas_rtc** (u16 vector_alarma[])
- void **habilitar_interrupciones_rtc** (u8 int_conteo, u8 int_alarma)
- void **calibrar_rtc** (u16 ajuste_conteo, u8 dir_correccion)
- void **guardar_info_rtc** (u32 datos_rtc[])
- void **leer_info_rtc** (u32 datos_rtc[])
- void **leer_hora_rtc** (u16 vector_hora[])
- void **RTC_IRQHandler** (void)
- void **RTCInit** (void)
- void **RTCStart** (void)
- void **RTCStop** (void)
- void **RTC_CTCTReset** (void)
- void **RTCSetTime** (RTCTime Time)
- void **RTCSetAlarm** (RTCTime Alarm)
- **RTCTime** **RTCGetTime** (void)
- void **RTCSetAlarmMask** (uint32_t AlarmMask)

- void **Inicializa_RTC** (uint32_t TSemDia, uint32_t THoras, uint32_t Tmin, uint32_t TMDia, uint32_t TMes, uint32_t TAny, int ABis)
- char * **Fecha_hora** (void)
- char * **Hora_min** (void)
- char * **Minutos_seg** (void)
- char * **Minutos** (void)
- char * **Segundos** (void)
- char * **Fecha** (void)
- uint32_t **Tiempo_intervalo** (int T_Anterior, int escala)
- uint32_t **Verifica_Hora_sistema** (u16 Actualiza_Hora[])

Variables

- char **DFecha_hora** [50]
- uint32_t **alarm_on**
- **RTCTime** local_time
- **RTCTime** alarm_time

Descripción detallada

Libreria para manejo del RTC.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO - TFG

Autor: Miguel Angel Iglesias Jimenez

Libreria que implementa el manejo del RTC del sistema LPC 1769

Fecha:

10/10/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **rtc.c**.

Documentación de las funciones

void **calibrar_rtc** (u16 *ajuste_conteo*, u8 *dir_correccion*)

Funcion **calibrar_rtc** Para calibrado del RTC

Parámetros:

	u16 <i>ajuste_conteo</i>
	u8 <i>dir_correccion</i> : None

Definición en la línea 139 del archivo rtc.c.

void configurar_alarmas_rtc (u16 vector_alarma[])

Funcion configurar_alarmas_rtc Configura alarma en el RTC

Parámetros:

	u16 vector_alarma[] : None
--	----------------------------

Definición en la línea 109 del archivo rtc.c.

void configurar_hora_rtc (u16 vector_hora_inicia[])

Funcion configurar_hora_rtc Configura hora y habilita RTC en el control de consumo reseta el RTC y no habilita la cuenta

Parámetros:

	None : None
--	-------------

Definición en la línea 69 del archivo rtc.c.

char* Fecha (void)

Funcion : Presenta fecha Formato HH:MM:SS DD/MM/AAA

Parámetros:

	none
--	------

Devuelve:

: cadena con fecha hora

Definición en la línea 501 del archivo rtc.c.

char* Fecha_hora (void)

Funcion : Presenta fecha hora Formato HH:MM:SS DD/MM/AAA

Parámetros:

	none
--	------

Devuelve:

: cadena con fecha hora

Definición en la línea 415 del archivo rtc.c.

void guardar_info_rtc (u32 datos_rtc[])

Funcion guardar_info_rtc Guarda informacion en registros RTC

Parámetros:

	u32 datos_rtc[] : None
--	------------------------

Definición en la línea 151 del archivo rtc.c.

void habilitar_interrupciones_rtc (u8 int_conteo, u8 int_alarma)

Funcion habilitar_interrupciones_rtc Activa la interrupcion del sistema RTC basada en RTC

Parámetros:

	u8 int_conteo
u8	int_alarma : None

Definición en la línea 127 del archivo rtc.c.

char* Hora_min (void)

Funcion : Hora_min Presenta Hora minutos segundos

Parámetros:

	none
--	------

Devuelve:

: cadena con Hora minutos

Definición en la línea 434 del archivo rtc.c.

void Inicializa_RTC (uint32_t *TSemDia*, uint32_t *THoras*, uint32_t *Tmin*, uint32_t *TMDia*, uint32_t *TMes*, uint32_t *TAny*, int *ABis*)

Funcion : Inicializa y Ajusta el timer haciendo uso de estructura. Formato (Dia Semana (1-7), hora(0-23) (min) dia mes aÃ±o (DD/MM/AAAA), Bisieso ON/OFF)

Parámetros:

	Tsec Segundos
	Tmin minutos
	THoras Horas (0-23)
	TMDia Dia del mes
	TMes Mes
	TAny AÃ±o
	TSemDia Dia de la semana (0-6)
	ABis (ON) any Bisiesto OFF (No bisiesto) : none

Definición en la línea 350 del archivo rtc.c.

void leer_hora_rtc (u16 vector_hora[])

Funcion leer_hora_rtc lee la hora actual del RTC

Parámetros:

	u16 vector_hora[] : None
--	--------------------------

Definición en la línea 179 del archivo rtc.c.

void leer_info_rtc (u32 datos_rtc[])

Funcion leer_info_rtc Lee informacion del RTC

Parámetros:

	u32 datos_rtc[] : None
--	------------------------

Definición en la línea 165 del archivo rtc.c.

char* Minutos (void)

Funcion : Minutos Presenta Minutos

Parámetros:

	none
--	------

Devuelve:

: cadena con minutos

Definición en la línea 469 del archivo rtc.c.

char* Minutos_seg (void)

Funcion : Minutos seg Presenta Minutos y segundos

Parámetros:

	none
--	------

Devuelve:

: cadena con minutos segundos

Definición en la línea 452 del archivo rtc.c.

void RTC_CTCTReset (void)

Funcion RTC_CTCTReset Efectua un reset en el contador del RTC

Parámetros:

	None : None
--	-------------

Definición en la línea 265 del archivo rtc.c.

void RTC_IRQHandler (void)

Funcion RTC_IRQHandler Activa la interrupcion del sistema RTC basada en ejecucion

Parámetros:

	None : None
--	-------------

Definición en la línea 201 del archivo rtc.c.

RTCTime RTCGetTime (void)

Funcion: RTCGetTime Lee el valor del RTC Get RTC timer value

Parámetros:

	None : Estructura con los datos del RTC
--	---

Definición en la línea 314 del archivo rtc.c.

void RTCInit (void)

Funcion RTCInit Inicializa el timer RTC

Parámetros:

	None : None
--	-------------

Definición en la línea 214 del archivo rtc.c.

void RTCSetAlarm (RTCTime Alarm)

Funcion: RTCSetAlarm Inicializa el RTC timer

Parámetros:

	None : None
--	-------------

Definición en la línea 295 del archivo rtc.c.

void RTCSetAlarmMask (uint32_t AlarmMask)

Funcion : RTCSetAlarmMask Ajusta la mascara de alarma del RTC

Parámetros:

	Mascara para ajuste de Alarma :
--	---------------------------------

Definición en la línea 332 del archivo rtc.c.

void RTCSetTime (RTCTime Time)

Funcion: RTCSetTime Ajusta el RTC timer

Parámetros:

	None : None
--	-------------

Definición en la línea 277 del archivo rtc.c.

void RTCStart (void)

Funcion: RTCStart ACTiva el Timer RTC Timer

Parámetros:

	None : None
--	-------------

Definición en la línea 237 del archivo rtc.c.

void RTCStop (void)

Funcion RTCStop

Para el Timer RTC

Parámetros:

	None : None
--	-------------

Definición en la línea 252 del archivo rtc.c.

char* Segundos (void)

Funcion : Minutos seg Presenta Minutos y segundos

Parámetros:

	none
--	------

Devuelve:

: cadena con segundos

Definición en la línea 485 del archivo rtc.c.

uint32_t Tiempo_intervalo (int *T_Anterior*, int *escala*)

Funcion "Tiempo_intervalo": Devuelve el tiempo entre el tiempo anterior pasado por parametro y el valor del RTC actual El tiempo pasado puede ir en segundo, minutos

Parámetros:

<i>T_Anterior</i>	Tiempo en segundos que se solicito el paquete.
<i>escala</i>	: 0 para tiempo en segundos, 1 para tiempo en minutos

Devuelve:

Temp_cal_recep Valor 0 si existe error o el tiempo en milisegundos que tarda en recibirlo

Definición en la línea 517 del archivo rtc.c.

uint32_t Verifica_Hora_sistema (u16 *Actualiza_Hora*[])

Funcion "Verifica Hora del sistema" Verifica la hora del sistema y si no es correcta la actualiza

Parámetros:

<i>T_Anterior</i>	Tiempo en segundos que se solicito el paquete.
<i>escala</i>	: 0 para tiempo en segundos, 1 para tiempo en minutos

Devuelve:

Temp_cal_recep Valor 0 si existe error o el tiempo en milisegundos que tarda en recibirlo

Definición en la línea 536 del archivo rtc.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/Server_Web.c

Liberia Libreria para implementacion de servidor para monitoreo del sistema de medidas.

```
#include "LPC17xx.h"
#include "FreeRTOS.h"
#include "semphr.h"
#include "task.h"
#include "Config_sis.h"
#include <cr_section_macros.h>
#include <NXP/crp.h>
#include "type.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "timer.h"
#include "uart.h"
#include "printf.h"
#include "Server_Web.h"
#include "ethmac.h"
#include "tcpip.h"
#include "Web/webside.h"
```

Funciones

- int **HTTPServer** (void)
- int **StartServer** (void)
- unsigned int **GetRangoVal** (void)

Variables

- unsigned int **aaPagecounter** =0
- unsigned int **adcValue** = 0
- unsigned int **aaScrollbar** = 400

Descripción detallada

Liberia Libreria para implementacion de servidor para monitoreo del sistema de medidas.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO -

Autor: Miguel Angel Iglesias Jimenez

Libreria para manejo dispositivo WiFly RN-XV 171 incluyendo las funciones mas usuales de dicho dispositivo

Fecha:

10/10/2013

Nota:

Copyright (C) 2013 - Angel Iglesias All rights reserved.

License: Se permiten el uso y la redistribucion de este codigo con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **Server_Web.c**.

Documentación de las funciones

unsigned int GetRangoVal (void)

Función GetRangoVal Obtiene los valores del rango para los ScroolBar de las representacion de barras

Parámetros:

<i>none</i>	
<i>none</i>	

Devuelve:

Valor Scroollbar

Definición en la línea 150 del archivo Server_Web.c.

int HTTPServer (void)

Función "HTTPServer": Implementa el servidor dinamico HTTP, espera hasta estar conectado y a continuacion envia una cabecera HTTP y el codigo HTML almacenado en la memoria. Para el envio hay que sustituir los variables con valores dinamicos

Definición en la línea 75 del archivo Server_Web.c.

int StartServer (void)

Función Start Server procedimiento para inicio del servidor WEB valores dinamicos en la pagina del servidor WEB. Esta funcion implementa la insercion de los datos de medida de los sensores en las correspondientes variables de la pagina WEB. Para el envio hay que sustituir los variables con valores dinamicos

Definición en la línea 127 del archivo Server_Web.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFC_Library/src/wdt.c

Liberia configuracion y uso de watchDog.

```
#include "LPC17xx.h"
#include "type.h"
#include "timer.h"
#include "wdt.h"
```

Funciones

- uint32_t **WDTInit** (void)
- void **WDTFeed** (void)
- void **WDT_IRQHandler** (void)

Variables

- volatile uint32_t **wdt_counter**

Descripción detallada

Liberia configuracion y uso de watchDog.

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

ASIGNATURA SISTEMAS EMBEBIDOS - TFG

Autor: Miguel Angel Iglesias Jimenez

Libreria para configuracion, y uso de watchdog

Fecha:

25/10/2013

License: Se permiten el uso y la redistribución de este código con modificaciones o sin ellas, siempre que se mantengan esta licencia y las notas de autor y copyright de arriba.

Definición en el archivo **wdt.c**.

Documentación de las funciones

void WDT_IRQHandler (void)

Interrupcion watchdog

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 58 del archivo wdt.c.

void WDTFeed (void)

Secuencia watchDog timer

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 48 del archivo wdt.c.

uint32_t WDTInit (void)

Inicializa la interrupcion del watchdog

Parámetros:

<i>none</i>	
<i>True</i>	o False segun se hay instalado o no la interrupcion.

Devuelve:

none

Definición en la línea 33 del archivo wdt.c.

Referencia del Archivo

C:/TFG/Programacion/LPC/Nuevo/TFG/main.c

Sistema Estacion Meteorologica con posicionamiento GPS".

```
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "semphr.h"
#include "LPC17xx.h"
#include <cr_section_macros.h>
#include <NXP/crp.h>
#include "type.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Config_sis.h"
#include "printf.h"
#include "uart.h"
#include "wifly.h"
#include "arpalab.h"
#include "timer.h"
#include "btw.h"
#include "rtc.h"
#include "i2c.h"
#include "actuadores.h"
#include "Eeprom.h"
#include "lcd.h"
#include "lcdg_i2c.h"
#include "TouchPanel.h"
#include "gps.h"
#include "wdt.h"
#include "MPL115A.h"
#include "vcnl4010.h"
#include "DTH22.h"
#include "MMA7361.h"
#include "tmp101.h"
#include "Images/Logo_UOC.h"
```

'defines'

- #define **Stack_size** configMINIMAL_STACK_SIZE
- #define **Pr_Conexion_Red** (tskIDLE_PRIORITY + 1)
- #define **Pr_ServerWeb** (tskIDLE_PRIORITY + 2)
- #define **Pr_Recep_Com** (tskIDLE_PRIORITY + 3)
- #define **Pr_TickServer** (tskIDLE_PRIORITY + 4)
- #define **Pr_Sensores** (tskIDLE_PRIORITY + 4)
- #define **Pr_WatchDog** (tskIDLE_PRIORITY + 4)
- #define **Pr_Alarma** (tskIDLE_PRIORITY + 4)
- #define **mainQUEUE_LENGTH** (20)

Funciones

- char * **Fecha_hora** (void)
- int **main** (void)
Procedimiento principal, crea las tareas, semaforos, colas y ejecuta el administrador de tareas.
- void **vApplicationIdleHook** (void)
Funcion que se ejecuta cuando la CPU esta sin ejecutar tareas.
- void **vApplicationTickHook** (void)
- void **vApplicationMallocFailedHook** (void)
Funcion que controla el error de creacion de tareas, colas o semaforos solo es ejecutada si existe un fallo, indicandonolo por UART.
- void **vApplicationStackOverflowHook** (xTaskHandle *pxTask, signed char *pcTaskName)
Funcion que controla el error de desbordamiento de pila. Nos indicara la tarea que lo produce.

Variables

- __CRP const unsigned int **CRP_WORD** = CRP_NO_CRP
- int **LCDG** =FALSE
- int **LCDT** =FALSE
- char **Buff_Log** [BUFSIZE]
- char **Buff_Log2** [BUFSIZE]
- volatile uint32_t **alarm_on**
- char **DFecha_hora** [50]
- u16 **Vector_hora_inicio** [8] = {0,7,3,3,7,0,01,2014}
- u16 **Hora_actual** [8] = {0,0,0,0,0,0,0,0}
- u16 **Actualiza_hora** [8] = {0,0,0,0,0,0,0,0}
- char **GPS_Datos** [10][125]
- char **dia** [2] = ""
- char **mes** [2] = ""
- char **anyo** [2] = ""
- char **hora** [2] = ""
- char **min** [2] = ""
- char **seg** [2] = ""
- char **latitud** [12] = ""
- char **longitud** [12] = ""
- char **altitud** [12] = ""
- char **N_S** [2] = ""
- char **E_W** [2] = ""
- char **Satelites** [2] = ""
- char **AFecha** [15]
- char **AHora** [15]
- char **Redes_Wifi** [10][BUFSIZE]
- int **Conexion_RED** =FALSE
- int **Desconexion_RED** =TRUE
- int **TickConexion** = 1000
- int **BTW_Conex** =FALSE
- int **GPS_ON** =FALSE
- int **Bluetooth** =FALSE
- int **ServerWebIniciado** =FALSE
- int **TickWebServer** =10
- int **EnvioArpLab** =FALSE
- char **NomAppenv** [30]
- int **Adquiriendo_Datos** =FALSE

- int **Error_Hum** =FALSE
- int **Error_Pres** =FALSE
- int **Error_Lum** =FALSE
- int **Error_Temp** =FALSE
- int **Error_Prox** =FALSE
- int **Error_GPS** =FALSE
- int **TickRecepComandos** =1000
- int **TickMuestreo** =1000
- int **Comando** =FALSE
- **Limites Niv_Limites**
- **ActSalidas Actuadores**
- uint8_t **SALIDAS** =0
- int **BIT_Alarma** =0
- int **BIT_Bluet** =0
- int **BIT_Wifly** =0
- xQueueHandle **xQueue_Server**
- xQueueHandle **xQueue_Conexion**
- xQueueHandle **xQueue_Sensores**
- xQueueHandle **xQueue_Recep_Com**
- xTaskHandle **xHandle_Sensores**
- xTaskHandle **xHandle_Conex_Red**
- xTaskHandle **xHandle_RecComandos**
- xTaskHandle **xHandle_Alarma**
- xTaskHandle **xHandle_WatchDog**
- xTaskHandle **xHandle_ServerWeb**
- xTaskHandle **xHandle_TickServer**
- xSemaphoreHandle **xMutexArpaLab**
- xSemaphoreHandle **xMutexMuestreo**
- volatile uint32_t **timer1_m0_counter**
- volatile uint32_t **feed_counter** = 0
- double **Coefficientes** [4]
- double **Med_MPL115A** [2]
- uint32_t **Luminosidad** =0
- uint32_t **Proximidad** =0
- **ACL_Data Datos_Acelerometro**

Descripción detallada

Sistema Estacion Meteorologica con posicionamiento GPS".

UNIVERSIDAD OBERTA DE CATALUNYA TFG 2013/14

TRABAJO FINAL DE GRADO - PAC 4

Autor: Miguel Angel Iglesias Jimenez

Aplicacion para sistema embebido Estacion meteorologica con posicionamiento GPS Procesador
LPC_1769

Versión:

Ver 1.0

Fecha:

21/01/2014

Nota:

Visitar pagina del autor: <http://www.infoteli.es>

Definición en el archivo **main.c**.

Documentación de las funciones

char* Fecha_hora (void)

Funcion : Presenta fecha hora Formato HH:MM:SS DD/MM/AAA

Parámetros:

	none
--	------

Devuelve:

: cadena con fecha hora

Definición en la línea 415 del archivo rtc.c.

int main (void)

Procedimiento principal, crea las tareas, semaforos, colas y ejecuta el administrador de tareas.

Procedimiento principal Crear las tareas, semaforos, colas ejecutar el administrador de tareas.

Parámetros:

none	
------	--

Devuelve:

none

Definición en la línea 816 del archivo main.c.

void vApplicationIdleHook (void)

Funcion que se ejecuta cuando la CPU esta sin ejecutar tareas.

Funcion ejecucion aplicaciones CPU desocupada

Parámetros:

none	
------	--

Devuelve:

none

INCORPORAR AQUI PROCEDIMIENTO PARA ACTIVAR BAJO CONSUMO SI SE ESTAN MAS

Definición en la línea 1428 del archivo main.c.

void vApplicationMallocFailedHook (void)

Funcion que controla el error de creacion de tareas, colas o semaforos solo es ejecutada si existe un fallo, indicandolo por UART.

Funcion Control Errores Malloc

Parámetros:

<i>none</i>	
-------------	--

Devuelve:

none

Definición en la línea 1459 del archivo main.c.

void vApplicationStackOverflowHook (xTaskHandle * *pxTask*, signed char * *pcTaskName*)

Funcion que controla el error de desbordamiento de pila. Nos indicara la tarea que lo produce.

Funcion Control Error desbordamiento de pila

Parámetros:

<i>pxTask</i>	Pila de la tarea
<i>pcTaskName</i>	Nombre de la tarea que produce el fallo

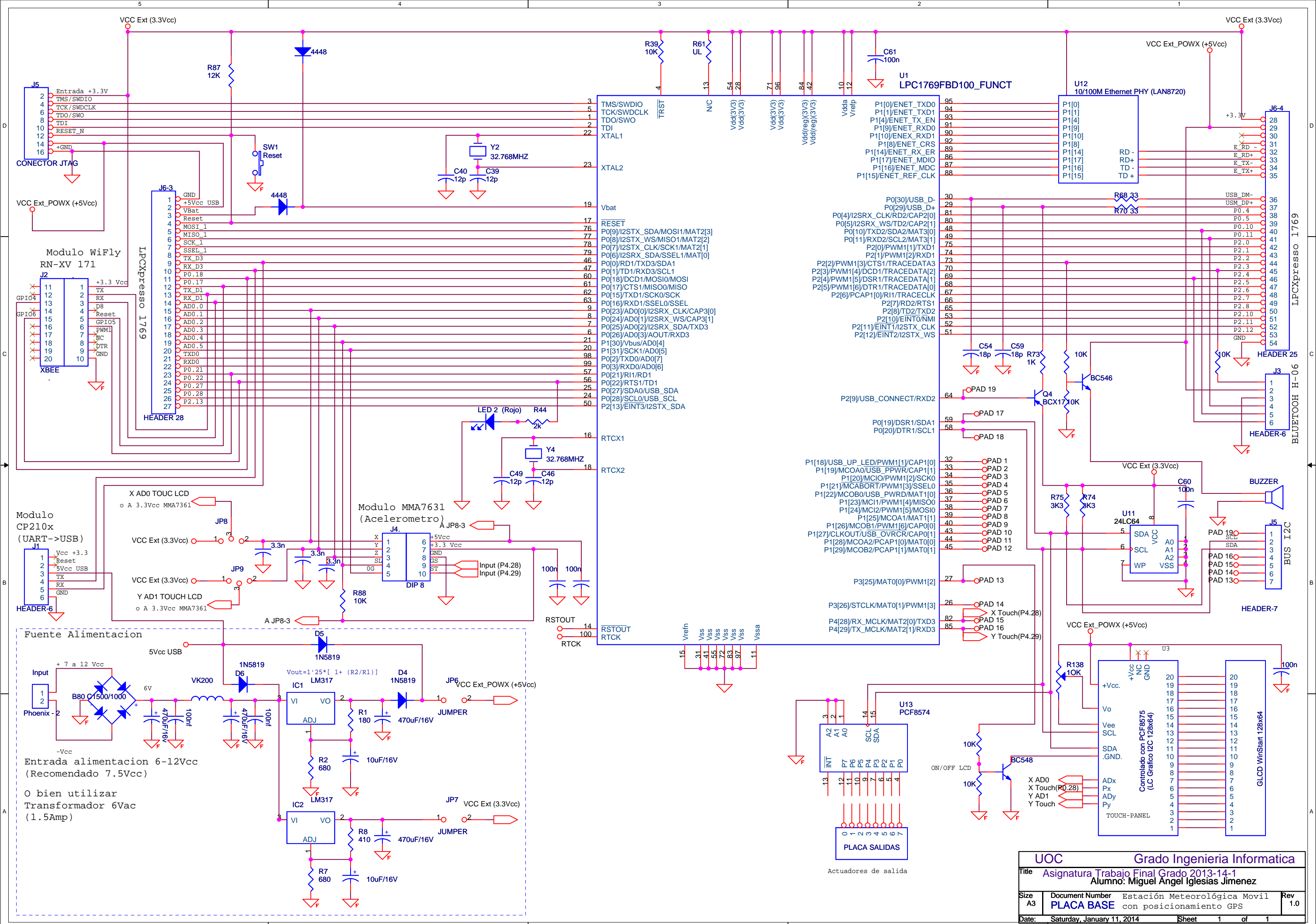
Devuelve:

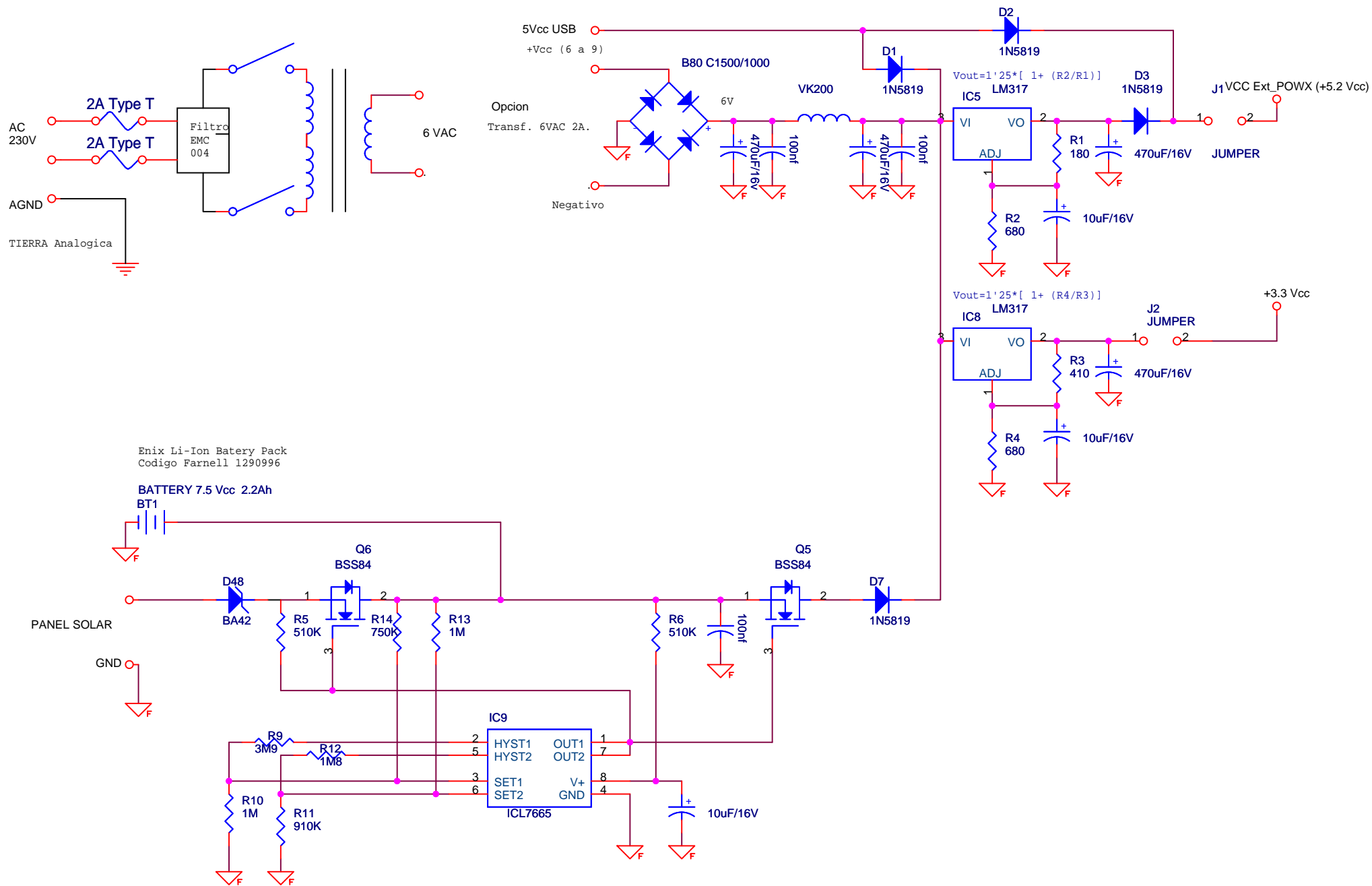
None

Definición en la línea 1476 del archivo main.c.

Anexo D

ESQUEMAS ELECTRICOS DEL SISTEMA





Page 2

LPC3154 Powering and Unused parts

Page 3

LPC3154 Digital I/O

Page 4

JTAG Interface

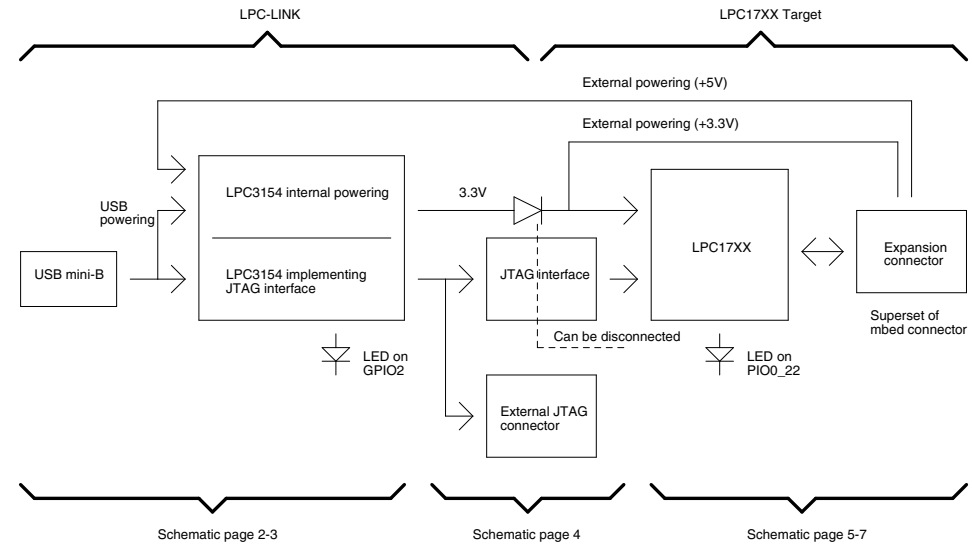
Page 5

LPC17XX with Expansion connector

Page 6

LPC17XX

Design Overview



UL = UnLoaded = normally not mounted component.

Default jumper settings are indicated in the schematic.
However, always check jumper positions on actual boards
since there is no guarantee that all jumpers are in default place.



(C) Embedded Artists AB

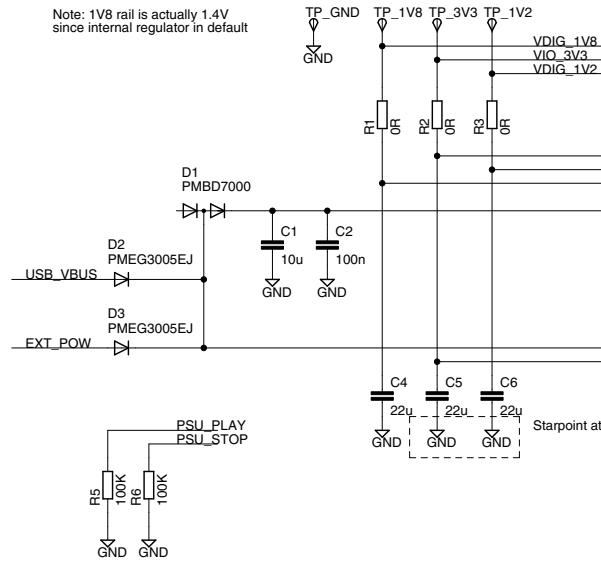
TITLE: LPCXpresso LPC1769 rev B

Document Number:

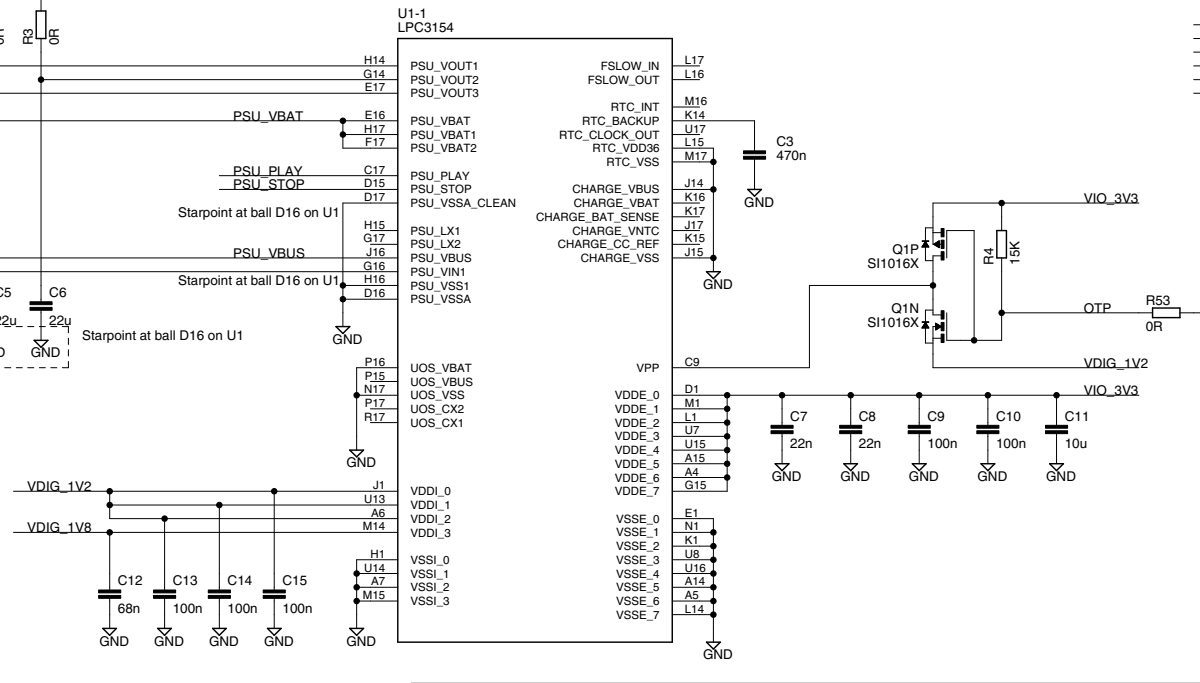
Date: 2010-10-19 14:19:36

Sheet: 1/7

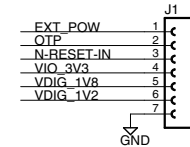
Note: 1V8 rail is actually 1.4V
since internal regulator in default



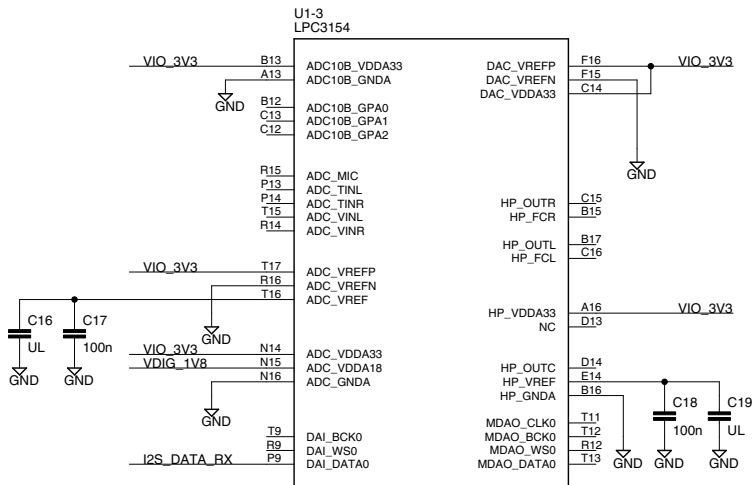
Power supply parts of LPC3154



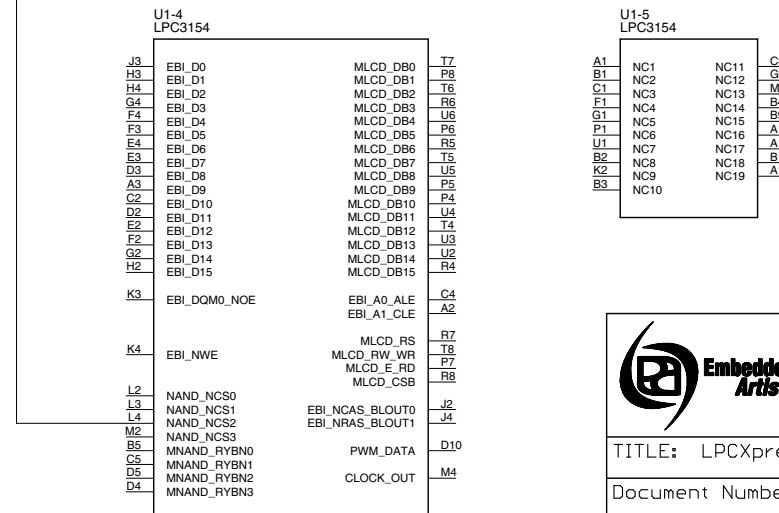
Production test connector



Analog parts of LPC3154



Not used parts of LPC3154



(C) Embedded Artists AB

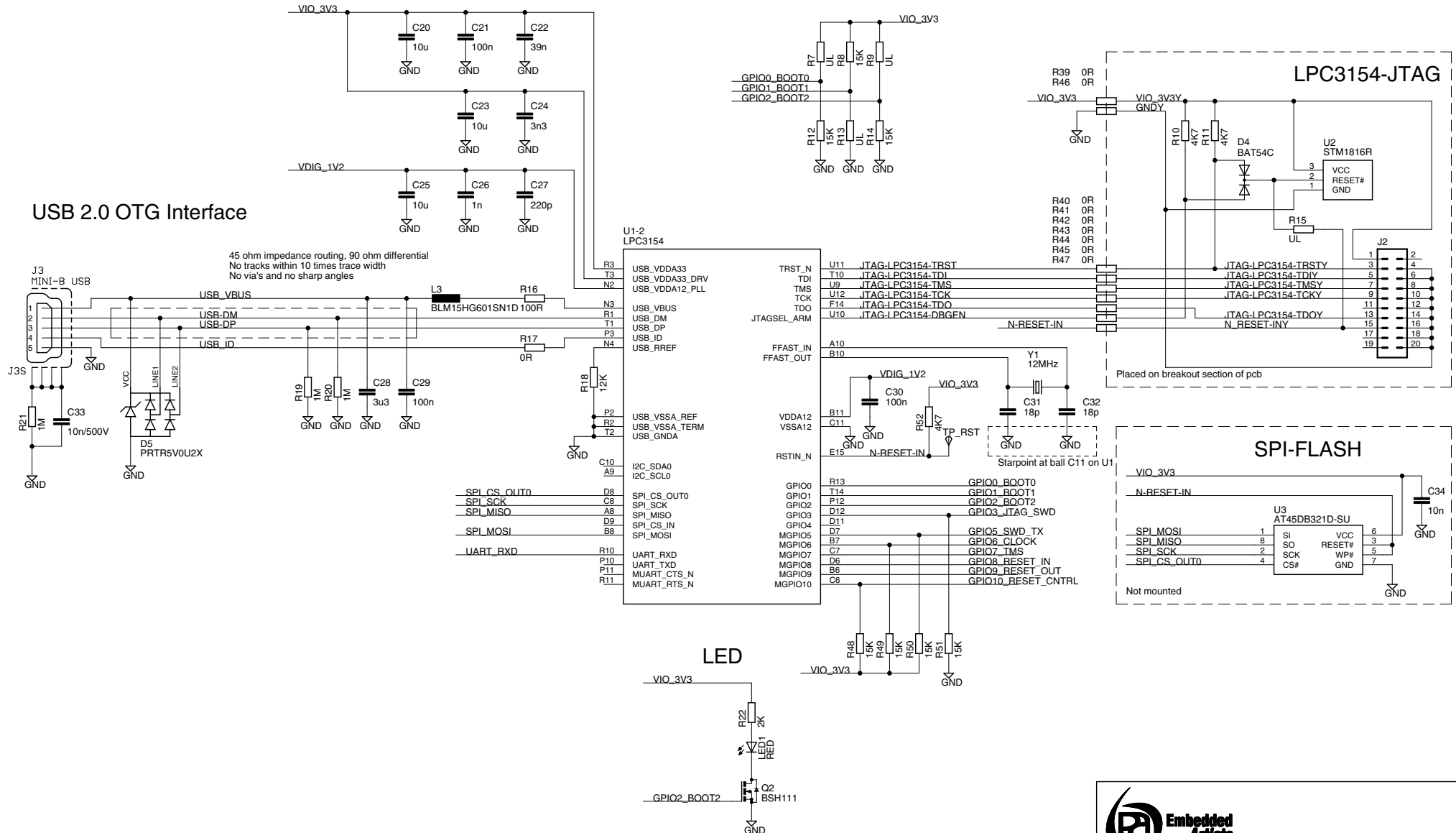
TITLE: LPCXpresso LPC1769 rev B

Document Number:

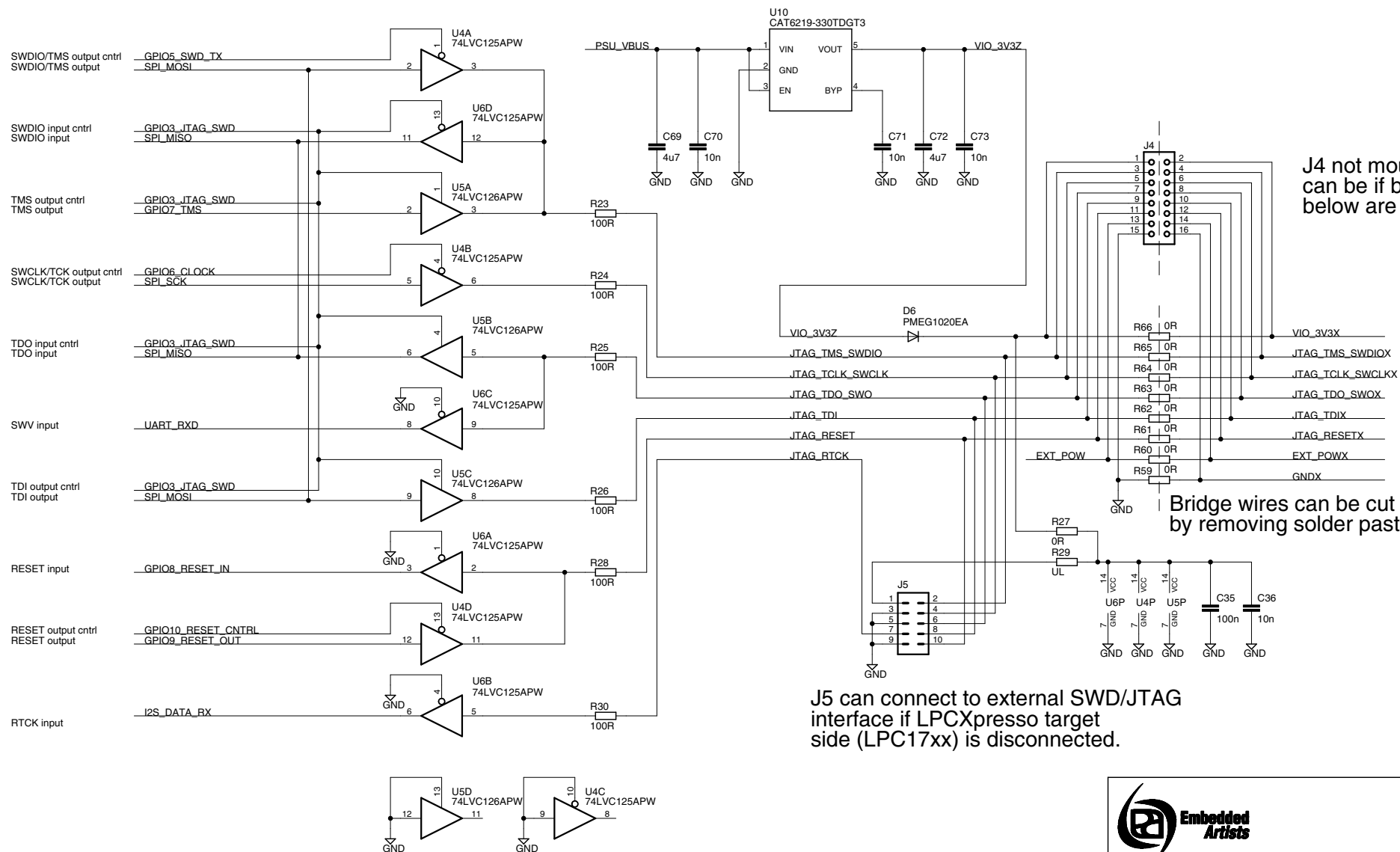
Date: 2010-10-19 14:19:36

Sheet: 2/7

Boot mode - USB via DFU class



SWD/JTAG Interface



J4 not mounted, but
can be if bridge wires
below are cut.

Bridge wires can be cut on pcb
by removing solder paste.

J5 can connect to external SWD/JTAG
interface if LPCXpresso target
side (LPC17xx) is disconnected.



(C) Embedded Artists AB

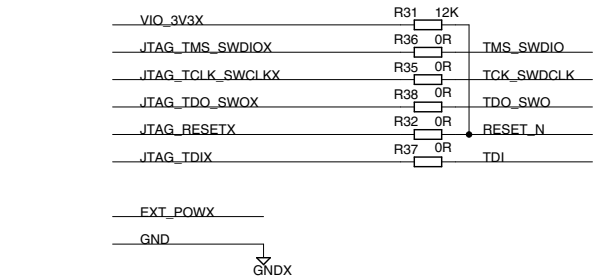
TITLE: LPCXpresso LPC1769 rev B

Document Number:

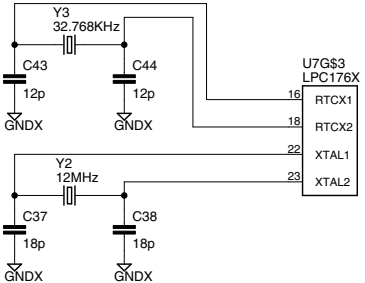
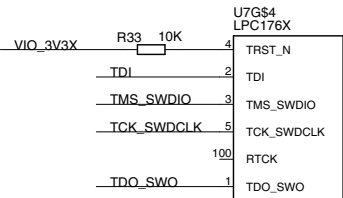
Date: 2010-10-19 14:19:36

Sheet: 4/7

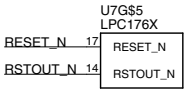
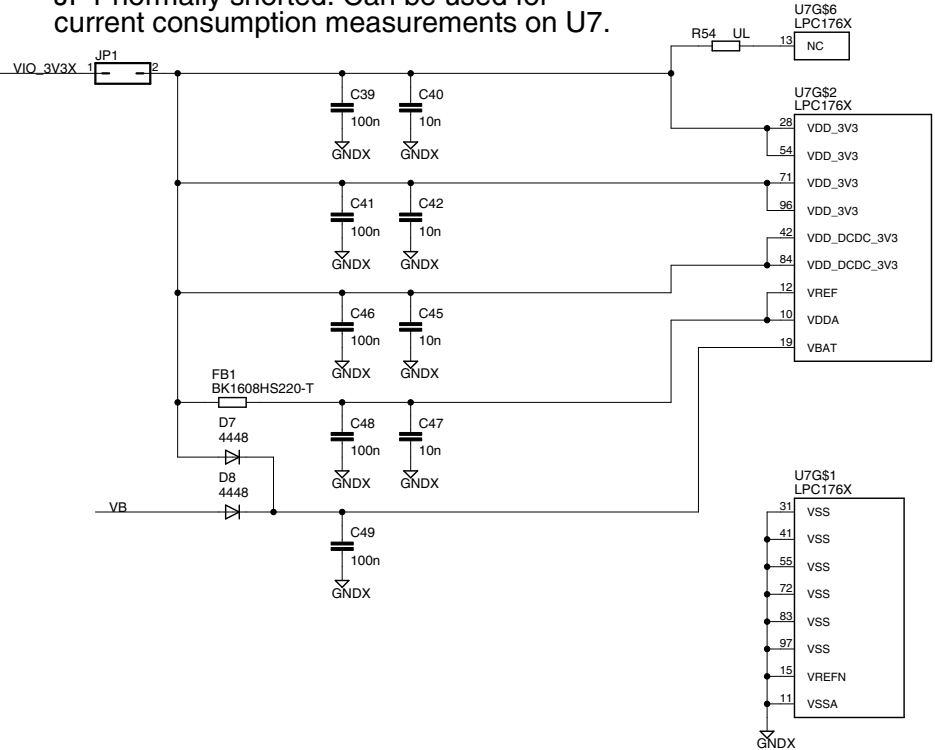
From LPC-LINK Side



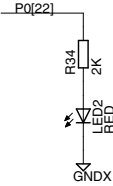
LPC176X Target Side



JP1 normally shorted. Can be used for current consumption measurements on U7.



LED



(C) Embedded Artists AB

TITLE: LPCXpresso LPC1769 rev B

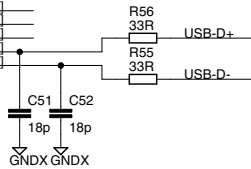
Document Number:

Date: 2010-10-19 14:19:36

Sheet: 5/7

U7G\$7
LPC176X

P0.0_RD1_TXD3_SDA1
P0.1_TD1_RXD3_SCL1
P0.2_TXD0_ADO7
P0.3_RXD0_ADO6
P0.4_I2SRX-CLK_RD2_CAP2.0
P0.5_I2SRX-WS_TD2_CAP2.1
P0.6_I2SRX-SDA_SSEL1_MAT2.0
P0.7_I2STX-CLK_SCK1_MAT2.1
P0.8_I2STX-WS_MISO1_MAT2.2
P0.9_I2STX-SDA_MOSI1_MAT2.3
P0.10_TXD2_SDA2_MAT3.0
P0.11_RXD2_SCL2_MAT3.1
P0.15_TXD1_SCK0_SCK
P0.16_RXD1_SSEL0_SSEL
P0.17_CTS1_MISO0_MISO
P0.18_DCD1_MOSI0_MOSI
P0.19_DSR1_SDA1
P0.20_DTR1_SCL1
P0.21_RI1_RD1
P0.22_RTS1_TD1
P0.23_ADO0_I2SRX-CLK_CAP3.0
P0.24_ADO1_I2SRX-WS_CAP3.1
P0.25_ADO2_I2SRX-SDA_TXD3
P0.26_ADO3_AOUT_RXD3
P0.27_SDA0_USB-SDA1
P0.28_SCL0_USB-SCL1
P0.29_USB-D+
P0.30_USB-D-



U7G\$8
LPC176X

P1.0_ENET-TXD0
P1.1_ENET-TXD1
P1.4_ENET-TX_EN
P1.8_ENET-CRS
P1.9_ENET-RXD0
P1.10_ENET-RXD1
P1.14_ENET-RX_ER
P1.15_ENET-REF_CLK
P1.16_ENET-MDC
P1.17_ENET-MDIO
P1.18_USB-UP-LED_PWM1.1_CAP1.0
P1.19_MCOA_USB-PPWR-N_CAP1.1
P1.20_MCFB0_PWM1.2_SCK0
P1.21_MCAOBORT_PWM1.3_SSEL0
P1.22_MCOB_USB-PWRD_MAT1.0
P1.23_MCFB1_PWM1.4_MISO0
P1.24_MCFB2_PWM1.5_MOSI0
P1.25_MC1A_MAT1.1
P1.26_MC1B_PWM1.6_CAP0.0
P1.27_CLKOUT_USB-OVRCR-N_CAP0.1
P1.28_MC2A_PCAP1.0_MAT0.0
P1.29_MC2B_PCAP1.1_MAT0.1
P1.30_VBUS_ADO4
P1.31_SCK1_ADO5

P1.0 P1[0]
P1.1 P1[1]
P1.4 P1[4]
P1.8 P1[8]
P1.9 P1[9]
P1.10 P1[10]
P1.14 P1[14]
P1.15 P1[15]
P1.16 P1[16]
P1.17 P1[17]
P1.18 P1[18]
P1.19 P1[19]
P1.20 P1[20]
P1.21 P1[21]
P1.22 P1[22]
P1.23 P1[23]
P1.24 P1[24]
P1.25 P1[25]
P1.26 P1[26]
P1.27 P1[27]
P1.28 P1[28]
P1.29 P1[29]
P1.30 P1[30]
P1.31 P1[31]

U7G\$9
LPC176X

P2.0_PWM1.1_TXD1
P2.1_PWM1.2_RXD1
P2.2_PWM1.3_CTS1_TRACEDATA[3]
P2.3_PWM1.4_DCD1_TRACEDATA[2]
P2.4_PWM1.5_DSR1_TRACEDATA[1]
P2.5_PWM1.6_DTR1_TRACEDATA[0]
P2.6_PCAP1.0_RI1_TRACECLK
P2.7_RD2_RTS1
P2.8_TD2_TXD2
P2.9_USB-CONNECT_RXD2
P2.10_EINT0-N_NMI
P2.11_EINT1-N_I2STX-CLK
P2.12_EINT2-N_I2STX-WS
P2.13_EINT3-N_I2STX-SDA

P2.0 P2[0]
P2.1 P2[1]
P2.2 P2[2]
P2.3 P2[3]
P2.4 P2[4]
P2.5 P2[5]
P2.6 P2[6]
P2.7 P2[7]
P2.8 P2[8]
P2.9 P2[9]
P2.10 P2[10]
P2.11 P2[11]
P2.12 P2[12]
P2.13 P2[13]

U7G\$10
LPC176X

P3.25_MAT0.0_PWM1.2
P3.26_STCLK_MAT0.1_PWM1.3

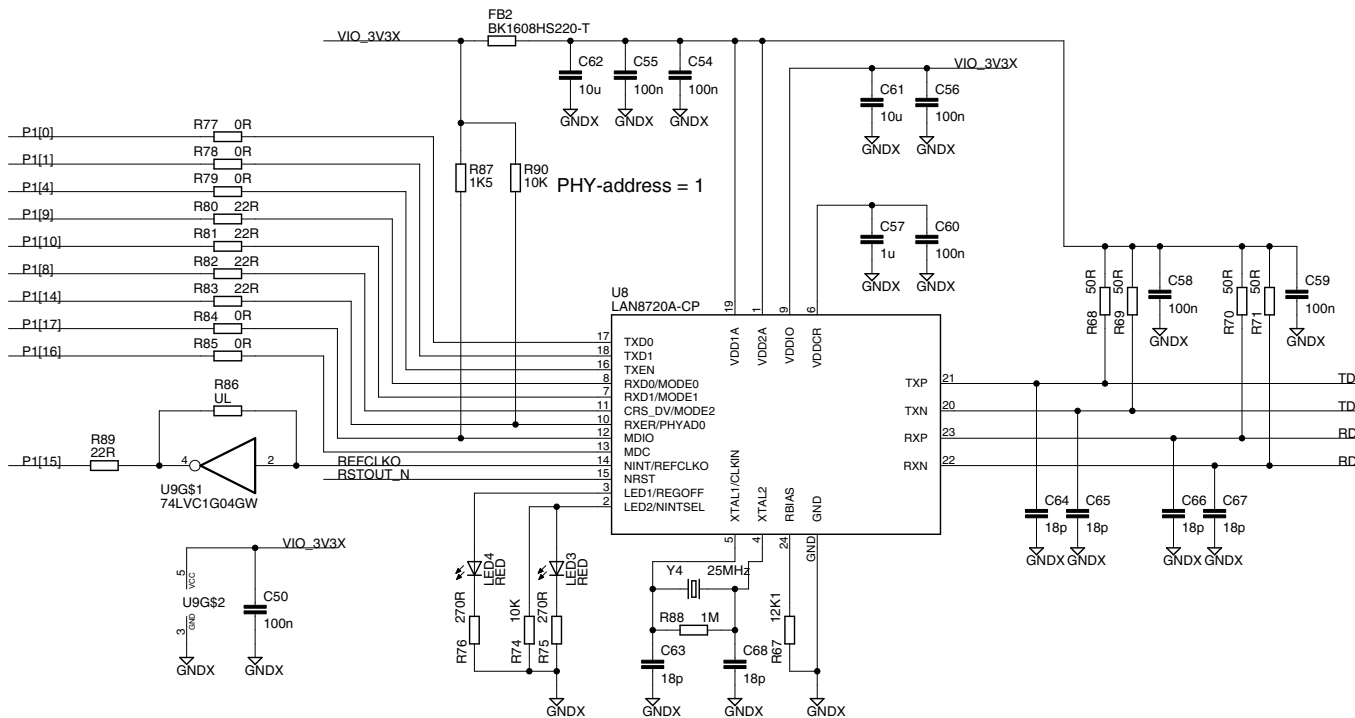
P3.25 P3[25]
P3.26 P3[26]

U7G\$11
LPC176X

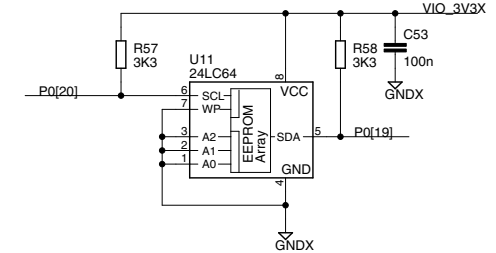
P4.28_RX-MCLK_MAT2.0_TXD3
P4.29_TX-MCLK_MAT2.1_RXD3

P4.28 P4[28]
P4.29 P4[29]

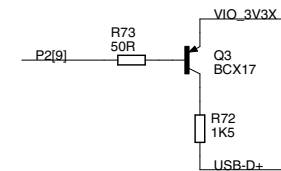
10/100M Ethernet PHY (LAN8720)



I2C-E2PROM



USB pullup for USB Device operation



(C) Embedded Artists AB

TITLE: LPCXpresso LPC1769 rev B

Document Number:

Date: 2010-10-19 14:19:36

Sheet: 6/7

LPC-LINK side

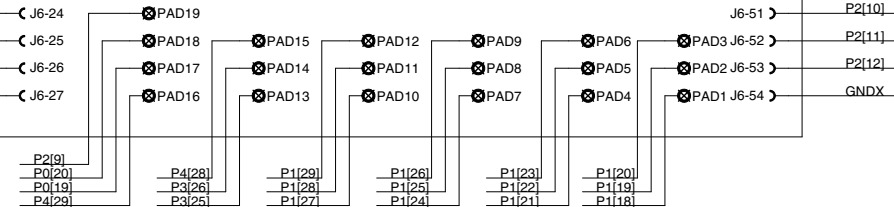
Expansion Connector (superset of mbed pinning)

Dual row holes (2x27), 100 mil spacing

mbed	LPCXpresso
GND	GND
VIN (4.5-14V)	VIN (4.5-5.5V)
VB (battery supply)	VB (battery supply)
nR (reset)	RESET_N
SPI1-MOSI	P0.9 MOSI1
SPI1-MISO	P0.8 MISO1
SPI1-SCK	P0.7 SCK1
GPIO	P0.6 SSEL1
UART1-TX / I2C1-SDA	P0.0 TXD3/SDA1
UART1-RX / I2C1-SCL	P0.1 RXD3/SCL1
SPI2-MOSI	P0.18 MOSI0
SPI2-MISO	P0.17 MISO0
SPI2-SCL / UART2-TX	P0.15 TXD1/SCK0
UART2-RX	P0.16 RXD1/SSEL0
AIN0	P0.23 AD0.0
AIN1	P0.24 AD0.1
AIN2	P0.25 AD0.2
AIN3 / AOUT	P0.26 AD0.3/AOUT
AIN4	P1.30 AD0.4
AIN5	P1.31 AD0.5
	P0.2
	P0.3
	P0.21
	P0.22
	P0.27
	P0.28
	P2.13

GNDX	J6-1
EXT_POWX	J6-2
VB	J6-3
RESET_N	J6-4
P0[9]	J6-5
P0[8]	J6-6
P0[7]	J6-7
P0[6]	J6-8
P0[0]	J6-9
P0[1]	J6-10
P0[18]	J6-11
P0[17]	J6-12
P0[15]	J6-13
P0[16]	J6-14
P0[23]	J6-15
P0[24]	J6-16
P0[25]	J6-17
P0[26]	J6-18
P1[30]	J6-19
P1[31]	J6-20
P0[2]	J6-21
P0[3]	J6-22
P0[21]	J6-23
P0[22]	J6-24
P0[27]	J6-25
P0[28]	J6-26
P2[13]	J6-27

J6-28	VIO_3V3X
J6-29	
J6-30	
J6-31	
J6-32	RD-
J6-33	RD+
J6-34	TD-
J6-35	TD+
J6-36	USB-D-
J6-37	USB-D+
J6-38	P0[4]
J6-39	P0[5]
J6-40	P0[10]
J6-41	P0[11]
J6-42	P2[0]
J6-43	P2[1]
J6-44	P2[2]
J6-45	P2[3]
J6-46	P2[4]
J6-47	P2[5]
J6-48	P2[6]
J6-49	P2[7]
J6-50	P2[8]
J6-51	P2[10]
	P2[11]
	P2[12]
	GNDX



LPCXpresso	mbed
VOUT (+3.3V out) if self powered, else +3.3V input	VOUT (3.3V out)
not used	VU (5.0V USB out)
not used	IF+
not used	IF-
RD-	RD- (Ethernet)
RD+	RD+ (Ethernet)
TD-	TD- (Ethernet)
TD+	TD+ (Ethernet)
USB-D-	D- (USB)
USB-D+	D+ (USB)
P0.4 CAN_RX2	CAN-RD
P0.5 CAN_TX2	CAN-TD
P0.10 TXD2/SDA2	UART3-TX / I2C2-SDA
P0.11 RXD2/SCL2	UART3-RX / I2C2-SCL
P2.0 PWM1.1	PWMOUT0
P2.1 PWM1.2	PWMOUT1
P2.2 PWM1.3	PWMOUT2
P2.3 PWM1.4	PWMOUT3
P2.4 PWM1.5	PWMOUT4
P2.5 PWM1.6	PWMOUT5
P2.6	
P2.7	
P2.8	
P2.10	
P2.11	
P2.12	
GND	



© Embedded Artists AB

TITLE: LPCXpresso LPC1769 rev B

Document Number:

Date: 2010-10-19 14:19:36

Sheet: 7/7

Anexo E

DATASHEET SENSORES Y COMPONENTES



Miniature I²C Digital Barometer

The MPL115A2 is an absolute pressure sensor with a digital I²C output targeting low cost applications. A miniature 5 by 3 by 1.2 mm LGA package is ideally suited for the space constrained requirements of portable electronic devices. Low current consumptions of 5 μ A during Active mode and 1 μ A during Shutdown (Sleep) mode are essential when focusing on low-power applications. The wide operating temperature range spans from -40°C to +105°C to fit demanding environmental conditions.

The MPL115A2 employs a MEMS pressure sensor with a conditioning IC to provide accurate pressure measurements from 50 to 115 kPa. An integrated ADC converts pressure and temperature sensor readings to digitized outputs via a I²C port. Factory calibration data is stored internally in an on-board ROM. Utilizing the raw sensor output and calibration data, the host microcontroller executes a compensation algorithm to render *Compensated Absolute Pressure* with ± 1 kPa accuracy.

The MPL115A2 pressure sensor's small form factor, low power capability, precision, and digital output optimize it for barometric measurement applications.

Features

- Digitized pressure and temperature information together with programmed calibration coefficients for host micro use.
- Factory Calibrated
- 50 kPa to 115 kPa Absolute Pressure
- ± 1 kPa Accuracy
- 2.375V to 5.5V Supply
- Integrated ADC
- I²C Interface (operates up to 400 kHz)
- 7 bit I²C address = 0x60
- Monotonic Pressure and Temperature Data Outputs
- Surface Mount RoHS Compliant Package

Application Examples

- Barometry (portable and desktop)
- Altimeters
- Weather Stations
- Hard Disk-Drives (HDD)
- Industrial Equipment
- Health Monitoring
- Air Control Systems

MPL115A2

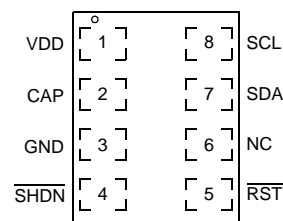
50 to 115 kPa



MPL115A2

5.0 mm by 3.0 mm by 1.2 mm

Top View



Pin Connections

ORDERING INFORMATION									
Device Name	Package Options	Case No.	# of Ports			Pressure Type			Digital Interface
			None	Single	Dual	Gauge	Differential	Absolute	
MPL115A2	Tray	2015	•					•	I ² C
MPL115A2T1	Tape & Reel (1000)	2015	•					•	I ² C
MPL115A2T2	Tape & Reel (5000)	2015	•					•	I ² C

1 Block Diagram and Pin Descriptions

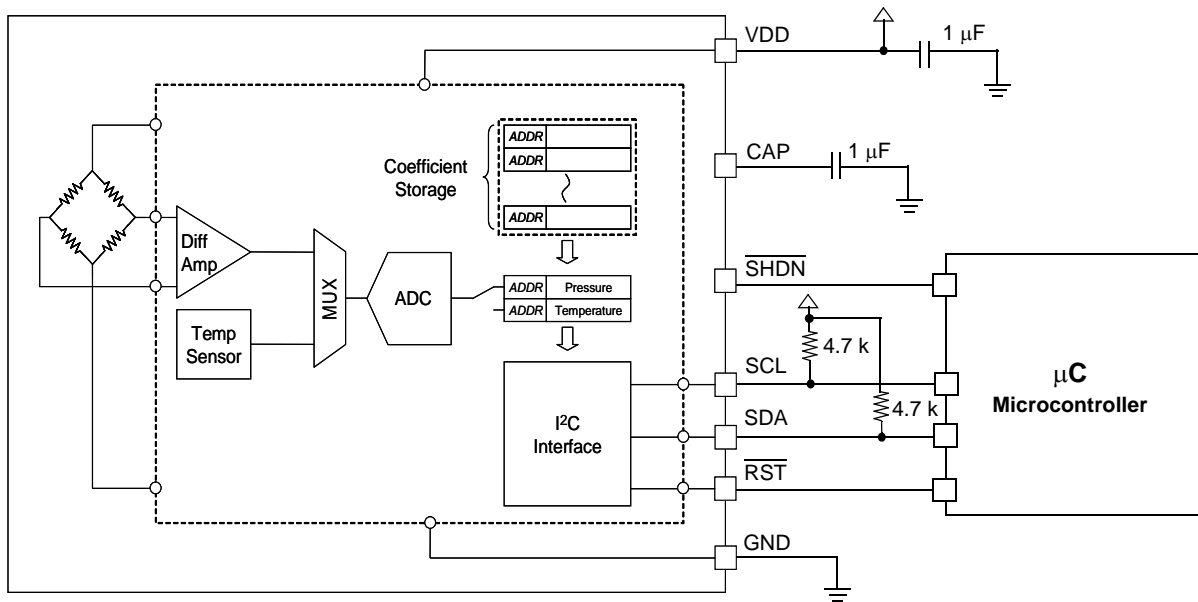


Figure 1. Block Diagram and Pin Connections

Table 1. Pin Description

Pin	Name	Function
1	VDD	VDD Power Supply Connection: VDD range is 2.375V to 5.5V.
2	CAP	1 μ F connected to ground.
3	GND	Ground
4	$\overline{\text{SHDN}}$	Shutdown: Connect to GND to disable the device. When in shutdown, the part draws no more than 1 μ A supply current and all communications pins ($\overline{\text{RST}}$, SCL, SDA) are high impedance. Connect to VDD for normal operation.
5	$\overline{\text{RST}}$	Reset: Connect to ground to disable I ² C communications.
6	NC	NC: No connection
7	SDA ⁽¹⁾	SDA: Serial data I/O line
8	SCL ⁽¹⁾	I ² C Serial Clock Input.

1. Use 4.7k pullup resistors for I²C communication.

2 Mechanical and Electrical Specifications

2.1 Maximum Ratings

Voltage (with respect to GND unless otherwise noted)

V_{DD}	-0.3 V to +5.5 V
\overline{SHDN} , RST, SDA, SCL	-0.3 V to $V_{DD}+0.3$ V
Operating Temperature Range	-40°C to +105°C
Storage Temperature Range	-40°C to +125°C
Overpressure	1000 kPa

2.2 Operating Characteristics

V_{DD} = 2.375 V to 5.5 V, T_A = -40°C to +105°C, unless otherwise noted. Typical values are at V_{DD} = 3.3 V, T_A = +25°C.

Ref	Parameters	Symbol	Conditions	Min	Typ	Max	Units
1	Operating Supply Voltage	V _{DD}		2.375	3.3	5.5	V
2	Supply Current	I _{DD}	Shutdown ($\overline{\text{SHDN}} = \text{GND}$)	—	—	1	μA
			Standby	—	3.5	10	μA
			Average – at one measurement per second	—	5	6	μA
Pressure Sensor							
3	Range			50	—	115	kPa
4	Resolution			—	0.15	—	kPa
5	Accuracy		-20°C to 85°C	—	—	±1	kPa
6	Power Supply Rejection		Typical operating circuit at DC		0.1	—	kPa/V
			100 mV p-p 217 Hz square wave plus 100 mV pseudo random noise with 10 MHz bandwidth		0.1	—	kPa
7	Conversion Time (Start Pressure and Temperature Conversion)	t _c	Time between start convert command and data available in the Pressure and Temperature registers	—	1.6	3	ms
8	Wakeup Time	t _w	Time between leaving Shutdown mode (SHDN goes high) and communicating with the device to issue a command or read data.	—	3	5	ms
I ² C I/O Stages: SCL, SDA							
9	SCL Clock Frequency	f _{SCL}		—	—	400	kHz
10	Low Level Input Voltage	V _{IL}		—	—	0.3V _{DD}	V
11	High Level Input Voltage	V _{IH}		0.7V _{DD}	—	—	V
I ² C Outputs: SDA							
12	Data Setup Time	t _{SU}	Setup time from command receipt to ready to transmit	0	—	0.4	s
I ² C Addressing							
MPL115A2 uses 7-bit addressing, does not acknowledge the general call address 0000000. Slave address has been set to 0x60 or 1100000.							

3 Overview of Functions/Operation

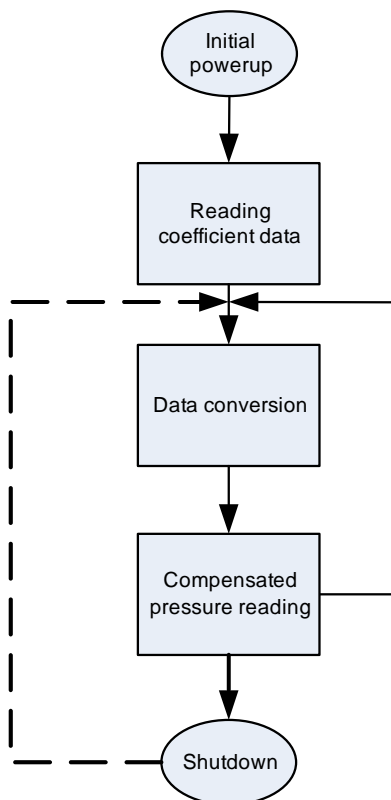


Figure 2. Sequence Flow Chart

The MPL115A interfaces to a host (or system) microcontroller in the user's application. All communications are via I²C. A typical usage sequence is as follows:

Initial Power-up

All circuit elements are active. I²C port pins are high impedance and associated registers are cleared. The device then enters standby mode.

Reading Coefficient Data

The user then typically accesses the part and reads the coefficient data. The main circuits within the slave device are disabled during read activity. The coefficients are usually stored in the host microcontroller local memory but can be re-read at any time.

It is not necessary to read the values stored in the host microcontroller multiple times because the coefficients within a device are constant and do not change. However, note that the coefficients will be different from device to device, and cannot be used for another part.

Data Conversion

This is the first step that is performed each time a new pressure reading is required which is initiated by the host sending the CONVERT command. The main system circuits are activated (wake) in response to the command and after the conversion completes, the result is placed into the Pressure and Temperature ADC output registers.

The conversion completes within the maximum conversion time, t_c (see [Row 7](#), in the Operating Characteristics Table). The device then enters standby mode.

Compensated Pressure Reading

After the conversion has been given sufficient time to complete, the host microcontroller reads the result from the ADC output registers and calculates the Compensated Pressure, a barometric/atmospheric pressure value which is compensated for changes in temperature and pressure sensor linearity. This is done using the coefficient data from the MPL115A and the raw sampled pressure and temperature ADC output values, in a compensation equation (detailed later). Note that this is an absolute pressure measurement with a vacuum as a reference.

From this step the host controller may either wait and then return to the Data Conversion step to obtain the next pressure reading or it may go to the Shutdown step.

Shutdown

For longer periods of inactivity the user may assert the $\overline{\text{SHDN}}$ input by driving this pin low to reduce system power consumption. This removes power from all internal circuits, including any registers. In the shutdown state, the Pressure and Temperature registers will be reset, losing any previous ADC output values.

This step is exited by taking the $\overline{\text{SHDN}}$ pin high. Wait for the maximum wakeup time, t_w (see [Row 8](#), in the Operating Characteristics Table), after which another pressure reading can be taken by transitioning to the data Conversion step.

Table 2. Device Memory Map

Address	Name	Description	Size (bits)
0x00	Padc_MSB	10-bit Pressure ADC output value MSB	8
0x01	Padc_LSB	10-bit Pressure ADC output value LSB	2
0x02	Tadc_MSB	10-bit Temperature ADC output value MSB	8
0x03	Tadc_LSB	10-bit Temperature ADC output value LSB	2
0x04	a0_MSB	a0 coefficient MSB	8
0x05	a0_LSB	a0 coefficient LSB	8
0x06	b1_MSB	b1 coefficient MSB	8
0x07	b1_LSB	b1 coefficient LSB	8
0x08	b2_MSB	b2 coefficient MSB	8
0x09	b2_LSB	b2 coefficient LSB	8
0x0A	c12_MSB	c12 coefficient MSB	8
0x0B	c12_LSB	c12 coefficient LSB	8
0x0C	Reserved*	—	—
0x0D	Reserved*	—	—
0x0E	Reserved*	—	—
0x0F	Reserved*	—	—
0x10	Reserved	—	—
0x11	Reserved	—	—
0x12	CONVERT	Start Pressure and Temperature Conversion	—

*These registers are set to 0x00. These are reserved, and were previously utilized as Coefficient values, c11 and c22, which were always 0x00.

For values with less than 16 bits, the lower LSBs are zero. For example, c12 is 14 bits and is stored into 2 bytes as follows:

c12 MS byte = $c12[13:6] = [c12_{b13}, c12_{b12}, c12_{b11}, c12_{b10}, c12_{b9}, c12_{b8}, c12_{b7}, c12_{b6}]$

c12 LS byte = $c12[5:0] \ \& \ "00" = [c12_{b5}, c12_{b4}, c12_{b3}, c12_{b2}, c12_{b1}, c12_{b0}, 0, 0]$

3.1 Pressure, Temperature and Coefficient Bit-Width Specifications

The table below specifies the initial coefficient bit-width specifications for the compensation algorithm and the specifications for Pressure and Temperature ADC values.

Pressure, Temperature and Compensation Coefficient Specifications						
	a0	b1	b2	c12	Padc	Tadc
Total Bits	16	16	16	14	10	10
Sign Bits	1	1	1	1	0	0
Integer Bits	12	2	1	0	10	10
Fractional Bits	3	13	14	13	0	0
dec pt zero pad	0	0	0	9	0	0

Example Binary Format Definitions:

a0 Signed, Integer Bits = 12, Fractional Bits = 3 :

b1 Signed, Integer Bits = 2, Fractional Bits = 7 :

b2 Signed, Integer Bits = 1, Fractional Bits = 14 :

c12 Signed, Integer Bits = 0, Fractional Bits = 13, dec pt zero pad = 9 :

Padc Unsigned, Integer Bits = 10 :

Tadc Unsigned, Integer Bits = 10 :

Coeff a0 = S I₁₁ I₁₀ I₉ I₈ I₇ I₆ I₅ I₄ I₃ I₂ I₁ I₀ . F₂ F₁ F₀

Coeff b1 = S I₁ I₀ . F₁₂ F₁₀ F₉ F₈ F₇ F₆ F₅ F₄ F₃ F₂ F₁ F₀

Coeff b2 = S I₀ . F₁₃ F₁₂ F₁₀ F₉ F₈ F₇ F₆ F₅ F₄ F₃ F₂ F₁ F₀

Coeff c12 = S 0 . 000 000 000 F₁₂ F₁₀ F₉ F₈ F₇ F₆ F₅ F₄ F₃ F₂ F₁ F₀

Padc U = I₉ I₈ I₇ I₆ I₅ I₄ I₃ I₂ I₁ I₀

Tadc U = I₉ I₈ I₇ I₆ I₅ I₄ I₃ I₂ I₁ I₀

NOTE: Negative coefficients are coded in 2's complement notation.

3.2 Compensation

The 10-bit compensated pressure output, Pcomp, is calculated as follows:

$$P_{comp} = a0 + (b1 + c12 \cdot Tadc) \cdot Padc + b2 \cdot Tadc$$

Eqn. 1

Where:

Padc is the 10-bit pressure ADC output of the MPL115A

Tadc is the 10-bit temperature ADC output of the MPL115A

a0 is the pressure offset coefficient

b1 is the pressure sensitivity coefficient

b2 is the temperature coefficient of offset (TCO)

c12 is the temperature coefficient of sensitivity (TCS)

Pcomp will produce a value of 0 with an input pressure of 50 kPa and will produce a full-scale value of 1023 with an input pressure of 115 kPa.

$$\text{Pressure (kPa)} = P_{comp} \cdot \left[\frac{115 - 50}{1023} \right] + 50$$

Eqn. 2

3.3 Evaluation Sequence, Arithmetic Circuits

The following is an example of the calculation for Pcomp, the compensated pressure output. Input values are in **bold**.

$$c12x2 = \mathbf{c12} \cdot \mathbf{Tadc}$$

$$a1 = \mathbf{b1} + c12x2$$

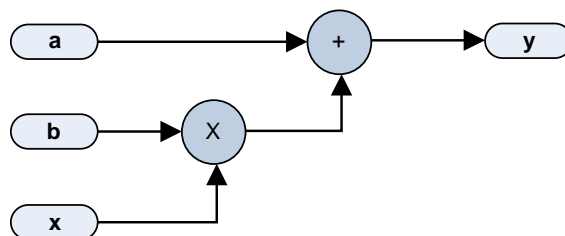
$$a1x1 = a1 \cdot \mathbf{Padc}$$

$$y1 = \mathbf{a0} + a1x1$$

$$a2x2 = \mathbf{b2} \cdot \mathbf{Tadc}$$

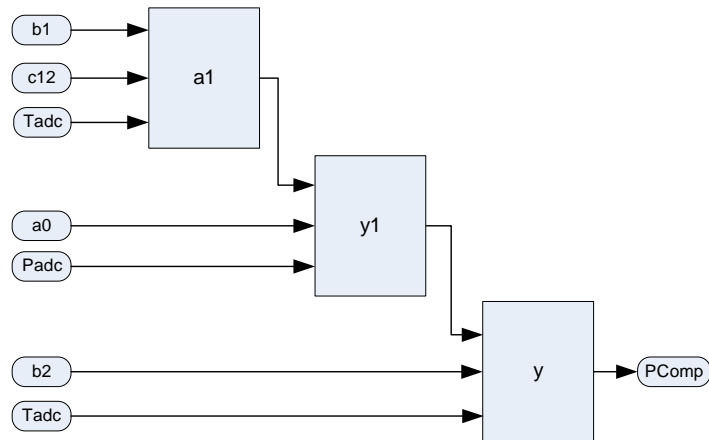
$$P_{comp} = y1 + a2x2$$

This can be calculated as a succession of Multiply Accumulates (MACs) operations of the form $y = a + b \cdot x$:



The polynomial can be evaluated (Equation 1) as a sequence of 3 MACs:

$$P_{comp} = a_0 + (b_1 + c_{12} \cdot T_{adc}) \cdot P_{adc} + b_2 \cdot T_{adc}$$



Please refer to Freescale application note AN3785 for more detailed notes on implementation.

3.4 I²C Device Read/Write Operations

All device read/write operations are memory mapped. Device actions e.g. “Start Conversions” are controlled by writing to the appropriate memory address location.

- For I²C the 7-bit Device Address (from Table 2) has a read/write toggle bit, where the least significant bit is ‘1’ for read operations or ‘0’ for write operations. The Device Address is 0xC0 for a *Write* and the Device Address is 0xC1 for a *Read*.
- The most significant bit in the Command tables below is not used and is don't care (X). In examples given it's set to ‘0’.

Refer to Sensor I²C Setup and FAQ Application Note AN4481 for more information on I²C communication between the sensor and host controller.

Table 3. I²C Write Commands

Command	Binary	HEX ⁽¹⁾
Devices Address + Write bit	1100 0000	0xC0
Start Conversions	X001 0010	0x12

X = Don't care

1 = The command byte needs to be paired with a 0x00 as part of the I²C exchange to complete the passing of Start Conversions.

The actions taken by the part in response to each command are as follows:

Table 4. I²C Write Command Description

Command	Action Taken
Start Conversions	Wake main circuits. Start clock. Allow supply stabilization time. Select pressure sensor input. Apply positive sensor excitation and perform A to D conversion. Select temperature input. Perform A to D conversion. Load the Pressure and Temperature registers with the result. Shut down main circuits and clock.

Table 5. I²C Read Command Description

Command	Binary	HEX ⁽¹⁾
Device Address + Read bit	1100 0001	0xC1
Read Pressure MSB	X000 0000	0x00
Read Pressure LSB	X000 0001	0x01
Read Temperature MSB	X000 0010	0x02
Read Temperature LSB	X000 0011	0x03
Read Coefficient data byte 1	X000 0100	0x04

X = don't care

These are MPL115A2 I²C commands to read coefficients, execute Pressure and Temperature conversions, and to read Pressure and Temperature data. The sequence of the commands for the interaction is given as an example to operate the MPL115A2.

Utilizing this gathered data, an example of the calculating the Compensated Pressure reading is given in floating point notation.

I²C Commands (simplified for communication)

Device Address + write bit "To Write" = 0xC0

Device Address + read bit "To Read" = 0xC1

Command to Write "Convert Pressure and Temperature" = 0x12

Command to Read "Pressure ADC High byte" = 0x00

Command to Read "Pressure ADC Low byte" = 0x01

Command to Read "Temperature ADC High byte" = 0x02

Command to Read "Temperature ADC Low byte" = 0x03

Command to Read "Coefficient data byte 1 High byte" = 0x04

Read Coefficients:

[0xC0], [0x04], [0xC1], [0x3E], [0xCE], [0xB3], [0xF9], [0xC5], [0x17], [0x33], [0xC8]

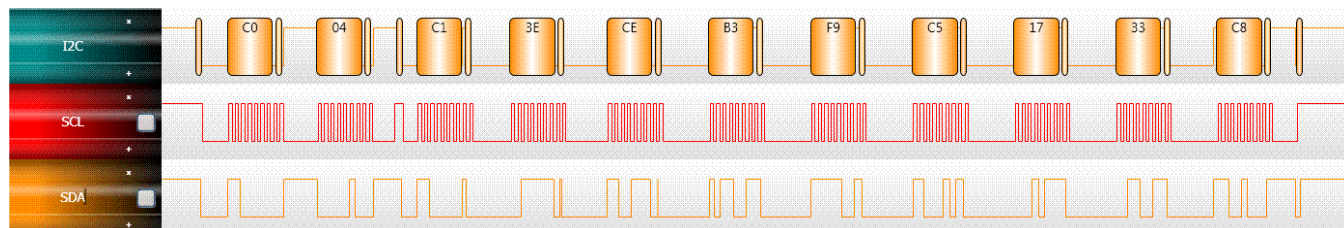


Figure 3. I²C Read Coefficient Datagram

a0 coefficient MSB	=	0x3E			
a0 coefficient LSB	=	0xCE	a0 coefficient	=	0x3ECE = 2009.75
b1 coefficient MSB	=	0xB3			
b1 coefficient LSB	=	0xF9	b1 coefficient	=	0xB3F9 = -2.37585
b2 coefficient MSB	=	0xC5			
b2 coefficient LSB	=	0x17	b2 coefficient	=	0xC517 = -0.92047
c12 coefficient MSB	=	0x33			
c12 coefficient LSB	=	0xC8	c12 coefficient	=	0x33C8 = 0.000790

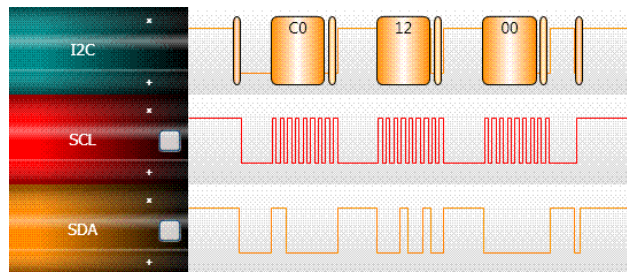


Figure 4. I²C Start Conversion Datagram
Command to I²C Start Conversion, 0x12

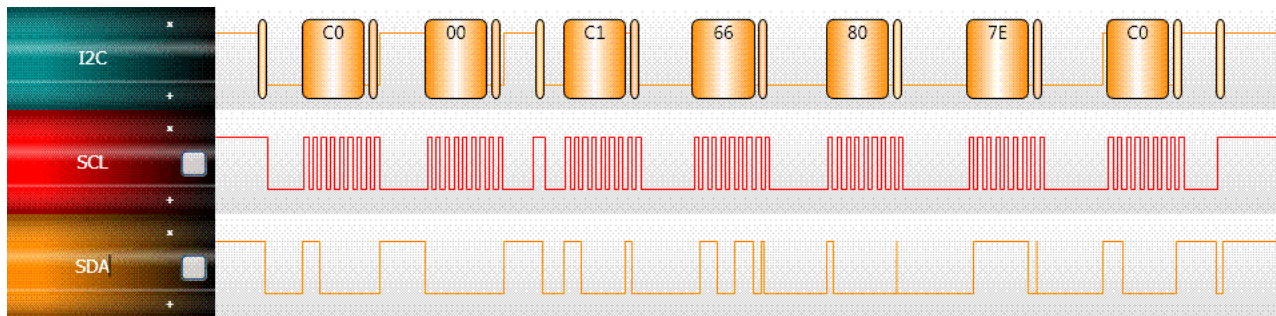


Figure 5. I²C Read Results Datagram

Pressure MSB	= 0x66				
Pressure LSB	= 0x80	Pressure	= 0x6680	= 0110 0110 1100 0000	
				= 410 ADC counts	
Temperature MSB	= 0x7E				
Temperature LSB	= 0xC0	Temperature	= 0x7EC0	= 0111 1110 1100 0000	
				= 507 ADC counts	

3.5 Example of Pressure Compensated Calculation in Floating-point Notation

a0 coefficient = 2009.75
b1 coefficient = -2.37585
b2 coefficient = -0.92047
c12 coefficient = 0.000790

Pressure = 410 ADC counts
Temperature = 507 ADC counts

Pressure Compensation:

$$P_{\text{comp}} = a_0 + (b_1 + c_{12} \cdot T_{\text{adc}}) \cdot P_{\text{adc}} + b_2 \cdot T_{\text{adc}}$$

Using the evaluation sequence shown in Section 3.3:

$$\begin{aligned}c_{12}x_2 &= c_{12} \cdot T_{\text{adc}} = 0.000790 \cdot 507 &= 0.40053 \\a_1 &= b_1 + c_{12}x_2 = -2.37585 + 0.40053 &= -1.97532 \\a_1x_1 &= a_1 \cdot P_{\text{adc}} = -1.97532 \cdot 410 &= -809.8812 \\y_1 &= a_0 + a_1x_1 = 2009.75 + (-809.8812) &= 1199.8688 \\a_2x_2 &= b_2 \cdot T_{\text{adc}} = -0.92047 \cdot 507 &= -466.67829 \\P_{\text{Comp}} &= y_1 + a_2x_2 = 1199.8688 + (-466.67829) &= 733.19051\end{aligned}$$

$$\begin{aligned}\text{Pressure (kPa)} &= P_{\text{comp}} \cdot \left[\frac{115 - 50}{1023} \right] + 50 \\&= 733.19 \cdot \left[\frac{115 - 50}{1023} \right] + 50 \\&= 96.59 \text{ kPa}\end{aligned}$$

4 Solder Recommendations

1. Use SAC solder alloy (i.e., Sn-Ag-Cu) with a melting point of about 217°C. It is recommended to use SAC305 (i.e., Sn-3.0 wt.% Ag-0.5 wt.% Cu).
2. Reflow
 - Ramp up rate: 2 to 3°C/s.
 - Preheat flat (soak): 110 to 130s.
 - Reflow peak temperature: 250°C to 260°C (depends on exact SAC alloy composition).
 - Time above 217°C: 40 to 90s (depends on board type, thermal mass of the board/quantities in the reflow).
 - Ramp down: 5 to 6°C/s.
 - Using an inert reflow environment (with O₂ level about 5 to 15 ppm).

NOTE: The stress level and signal offset of the device also depends on the board type, board core material, board thickness and metal finishing of the board.

Please refer to Freescale application note AN3150, Soldering Recommendations for Pressure Sensor Devices for any additional information.

5 Handling Recommendations

It is recommended to handle the MPL115A pressure sensor with a vacuum pick and place tool. Sharp objects utilized to move the MPL115A pressure sensor increase the possibility of damage via a foreign object/tool into the small exposed port.

The sensor die is sensitive to light exposure. Direct light exposure through the port hole can lead to varied accuracy of pressure measurement. Avoid such exposure to the port during normal operation.

Please note that the Pin 1 designator is on the bottom of the package. Do not use the port as a orientation reference in production.

6 Soldering/Landing Pad Information

The LGA package is compliant with the RoHS standard. It is recommended to use a no-clean solder paste to reduce cleaning exposure to high pressure and chemical agents that can damage or reduce life span of the Pressure sensing element.

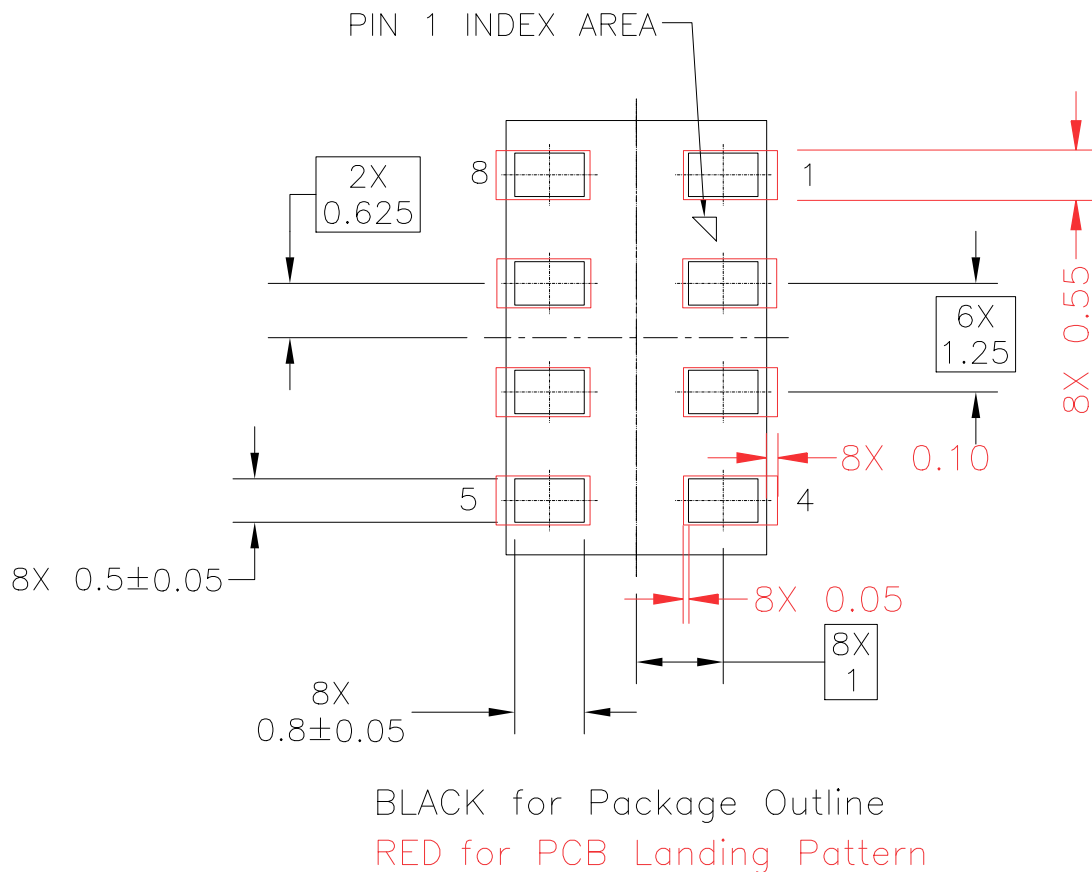


Figure 6. MPL115A2 Recommended PCB Landing Pattern

7 Tape and Reel Specifications

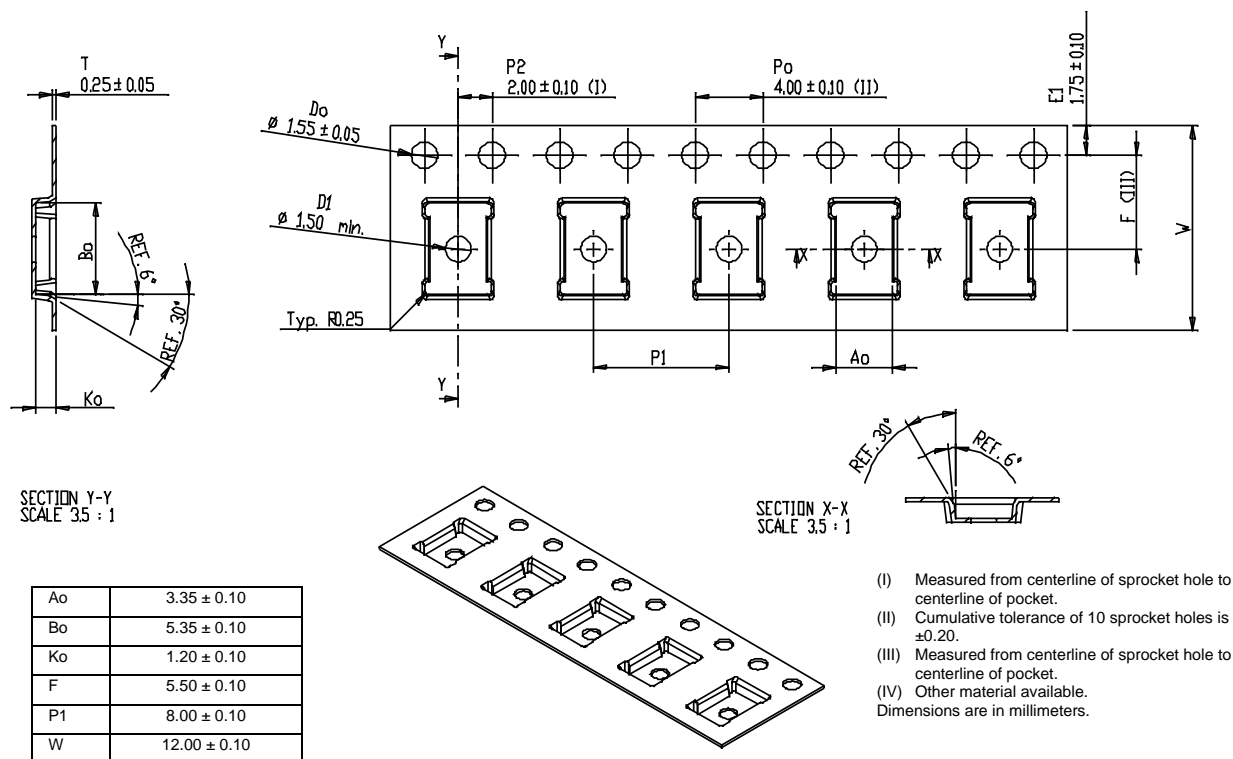


Figure 7. LGA (3 by 5) Embossed Carrier Tape Dimensions

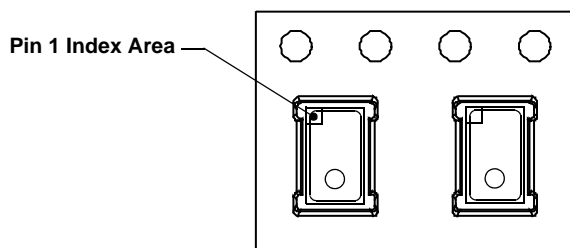
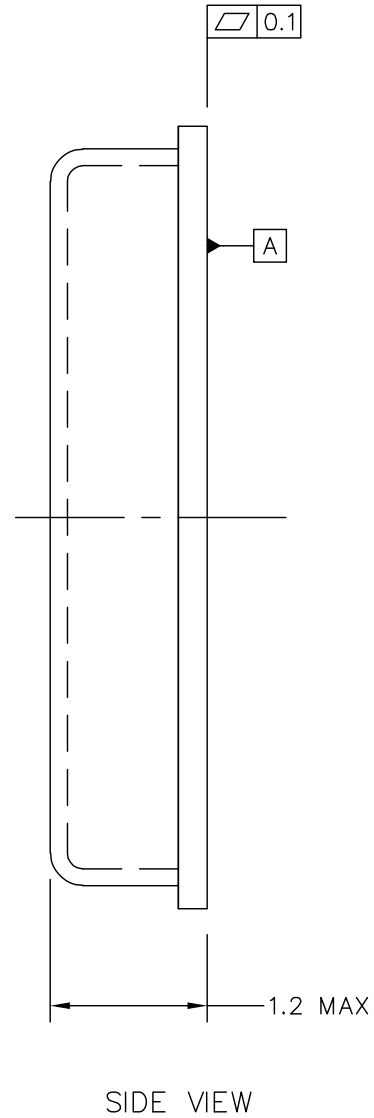
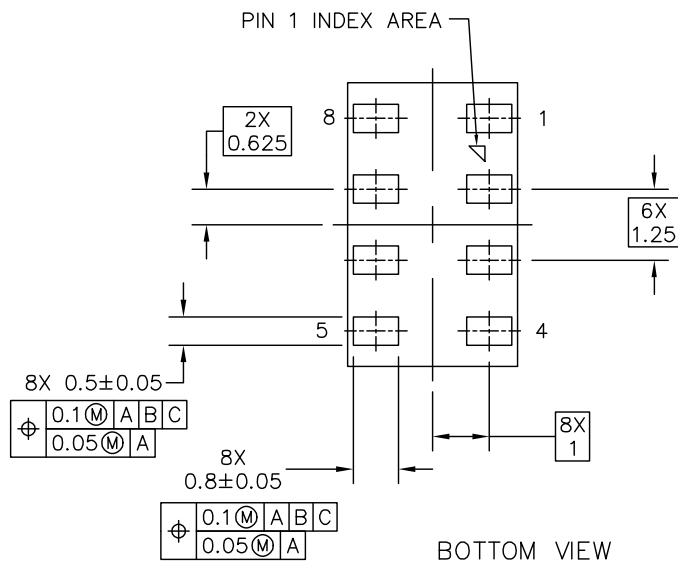
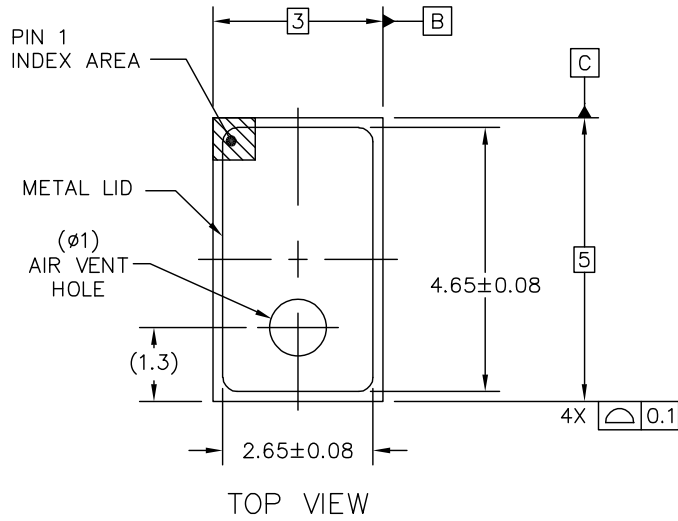


Figure 8. Device Orientation in Chip Carrier

PACKAGE DIMENSIONS



NOTES:

1. ALL DIMENSIONS IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: LGA 8 I/O, 3 X 5 X 1.25 PITCH, SENSOR 1.2MAX MM PKG	DOCUMENT NO: 98ASA10829D		REV: A
	CASE NUMBER: 2015-02		10 MAR 2010
	STANDARD: NON-JEDEC		

**CASE 2015-02
ISSUE A
LGA PACKAGE**

Related Documentation

The MPL115A2 device features and operations are described in a variety of reference manuals, user guides, and application notes. To find the most-current versions of these documents:

1. Go to the Freescale homepage at:
<http://www.freescale.com/>
2. In the Keyword search box at the top of the page, enter the device number MPL115A2.
3. In the Refine Your Result pane on the left, click on the Documentation link.

Table 6. Revision History

Revision number	Revision date	Description of changes
8	06/2012	<ul style="list-style-type: none">Updated graphic on page 1, Section 2.2 Operating Characteristics: Ref 7: Conversion Time: changed Typ from 3.0 to 1.6, Section 3.0 Overview of Functions/Operation: Reading Coefficient Data deleted statement that reading of coefficients may be executed only once, Table 2: added Size (bits) column in table, added new Section 3.4 I²C Device Read/Write Operations
		<ul style="list-style-type: none">
		<ul style="list-style-type: none">

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2012 Freescale Semiconductor, Inc. All rights reserved.



Digital Temperature Sensor with I²C™ Interface

FEATURES

- **DIGITAL OUTPUT:** I²C Serial 2-Wire
- **RESOLUTION:** 9- to 12-Bits, User-Selectable
- **ACCURACY:**
 $\pm 2.0^{\circ}\text{C}$ from -25°C to $+85^{\circ}\text{C}$ (max)
 $\pm 3.0^{\circ}\text{C}$ from -55°C to $+125^{\circ}\text{C}$ (max)
- **LOW QUIESCENT CURRENT:**
 $45\mu\text{A}$, $0.1\mu\text{A}$ Standby
- **WIDE SUPPLY RANGE:** 2.7V to 5.5V
- **TINY SOT23-6 PACKAGE**

APPLICATIONS

- **POWER-SUPPLY TEMPERATURE MONITORING**
- **COMPUTER PERIPHERAL THERMAL PROTECTION**
- **NOTEBOOK COMPUTERS**
- **CELL PHONES**
- **BATTERY MANAGEMENT**
- **OFFICE MACHINES**
- **THERMOSTAT CONTROLS**
- **ENVIRONMENTAL MONITORING AND HVAC**
- **ELECTROMECHANICAL DEVICE TEMPERATURE**

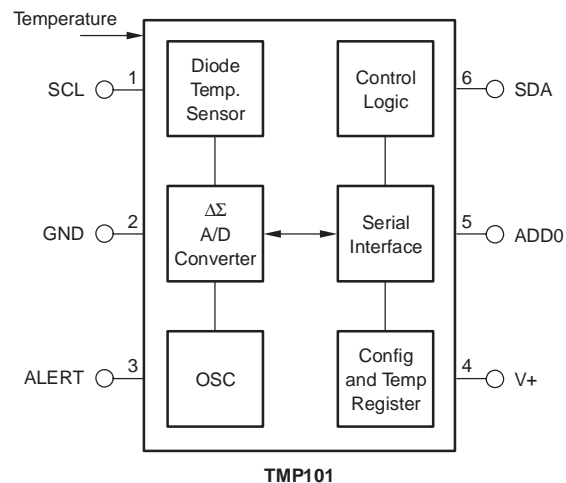
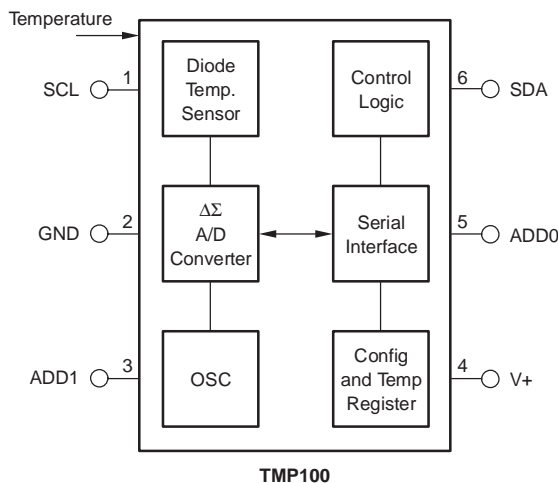
DESCRIPTION

The TMP100 and TMP101 are two-wire, serial output temperature sensors available in SOT23-6 packages. Requiring no external components, the TMP100 and TMP101 are capable of reading temperatures with a resolution of 0.0625°C .

The TMP100 and TMP101 feature SMBus and I²C interface compatibility, with the TMP100 allowing up to eight devices on one bus. The TMP101 offers SMBus alert function with up to three devices per bus.

The TMP100 and TMP101 are ideal for extended temperature measurement in a variety of communication, computer, consumer, environmental, industrial, and instrumentation applications.

The TMP100 and TMP101 are specified for operation over a temperature range of -55°C to $+125^{\circ}\text{C}$.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

I²C is a trademark of NXP Semiconductors. All other trademarks are the property of their respective owners.

ABSOLUTE MAXIMUM RATINGS⁽¹⁾

Power Supply, V+	7.5V
Input Voltage ⁽²⁾	–0.5V to 7.5V
Operating Temperature Range	–55°C to +125°C
Storage Temperature Range	–60°C to +150°C
Junction Temperature (T _J max)	+150°C
ESD Rating, Human Body Model	2000V
Machine Model	200V

(1) Stresses above these ratings may cause permanent damage. Exposure to absolute maximum conditions for extended periods may degrade device reliability. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those specified is not supported.

(2) Input voltage rating applies to all TMP100 and TMP101 input voltages.



This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

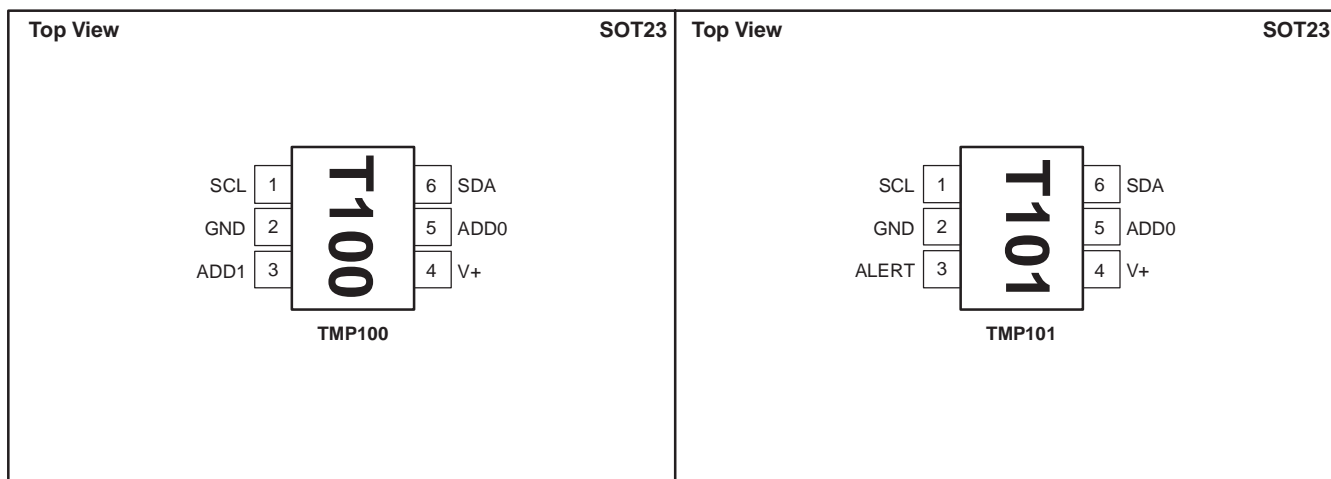
ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

ORDERING INFORMATION⁽¹⁾

PRODUCT	PACKAGE-LEAD	PACKAGE DESIGNATOR	PACKAGE MARKING
TMP100	SOT23-6	DBV	T100
TMP101	SOT23-6	DBV	T101

(1) For the most current package and ordering information, see the Package Option Addendum at the end of this document, or see the TI web site at www.ti.com.

PIN CONFIGURATION



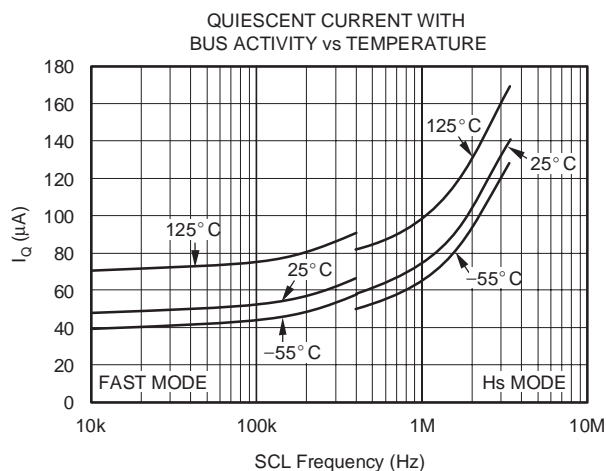
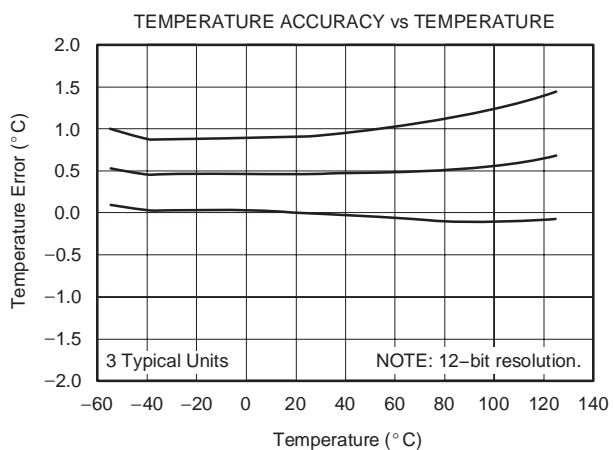
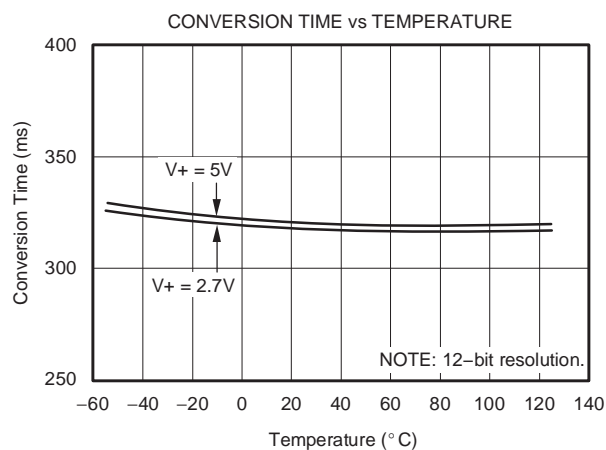
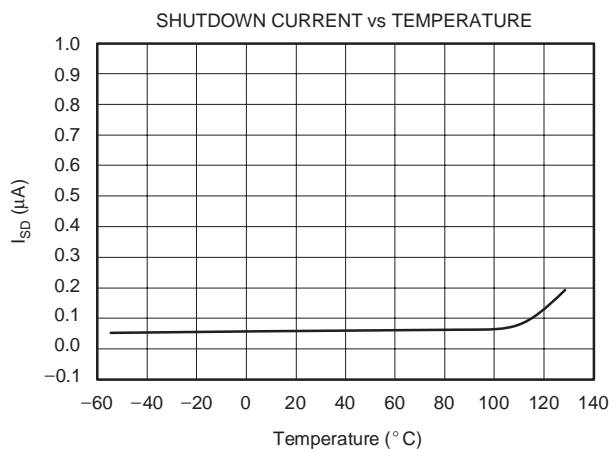
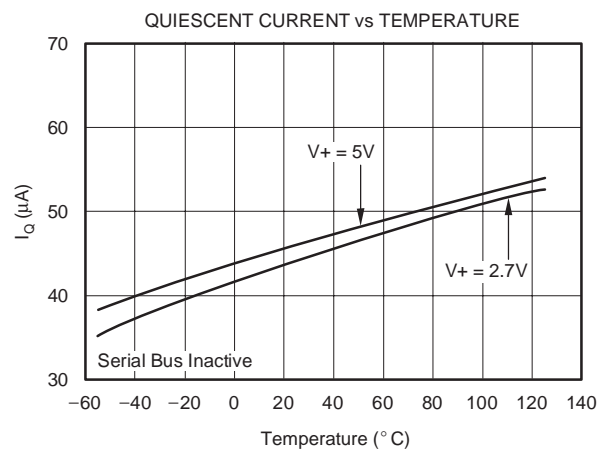
ELECTRICAL CHARACTERISTICS

At $T_A = -55^{\circ}\text{C}$ to $+125^{\circ}\text{C}$ and $V_+ = 2.7\text{V}$ to 5.5V , unless otherwise noted.

PARAMETER	TEST CONDITIONS	TMP100, TMP101			UNIT
		MIN	TYP	MAX	
TEMPERATURE INPUT					
Range		-55		+125	$^{\circ}\text{C}$
Accuracy (temperature error)	-25°C to $+85^{\circ}\text{C}$		± 0.5	± 2.0	$^{\circ}\text{C}$
	-55°C to $+125^{\circ}\text{C}$		± 1.0	± 3.0	$^{\circ}\text{C}$
Resolution	Selectable		± 0.0625		$^{\circ}\text{C}$
DIGITAL INPUT/OUTPUT					
Input Logic Levels:					
V_{IH}		$0.7(V_+)$		6.0	V
V_{IL}		-0.5		$0.3(V_+)$	V
Input Current, I_{IN}	$0\text{V} \leq V_{IN} \leq 6\text{V}$			1	μA
Output Logic Levels:					
V_{OL} SDA	$I_{OL} = 3\text{mA}$	0	0.15	0.4	V
V_{OL} ALERT	$I_{OL} = 4\text{mA}$	0	0.15	0.4	V
Resolution	Selectable		9 to 12		Bits
Conversion Time	9-Bit		40	75	ms
	10-Bit		80	150	ms
	11-Bit		160	300	ms
	12-Bit		320	600	ms
Conversion Rate	9-Bit		25		s/s
	10-Bit		12		s/s
	11-Bit		6		s/s
	12-Bit		3		s/s
POWER SUPPLY					
Operating Range		2.7		5.5	V
Quiescent Current	I_Q				
	Serial Bus Inactive		45	75	μA
	Serial Bus Active, SCL Frequency = 400kHz		70		μA
	Serial Bus Active, SCL Frequency = 3.4MHz		150		μA
Shutdown Current	I_{SD}				
	Serial Bus Inactive		0.1	1	μA
	Serial Bus Active, SCL Frequency = 400kHz		20		μA
	Serial Bus Active, SCL Frequency = 3.4MHz		100		μA
TEMPERATURE RANGE					
Specified Range		-55		+125	$^{\circ}\text{C}$
Storage Range		-60		+150	$^{\circ}\text{C}$
Thermal Resistance	θ_{JA}		200		$^{\circ}\text{C/W}$
	SOT23-6 Surface-Mount				

TYPICAL CHARACTERISTICS

At $T_A = +25^\circ\text{C}$ and $V_+ = 5.0\text{V}$, unless otherwise noted.



APPLICATIONS INFORMATION

The TMP100 and TMP101 are digital temperature sensors optimal for thermal management and thermal protection applications. The TMP100 and TMP101 are I²C and SMBus interface-compatible and are specified over a temperature range of –55°C to +125°C.

The TMP100 and TMP101 require no external components for operation except for pull-up resistors on SCL, SDA, and ALERT, although a 0.1µF bypass capacitor is recommended, as shown in Figure 1 and Figure 2.

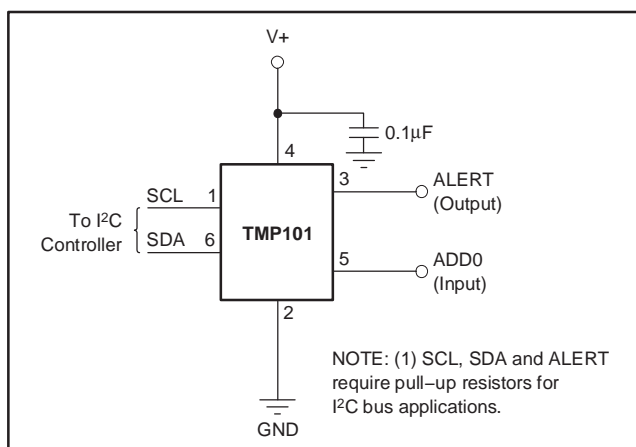


Figure 1. Typical Connections of the TMP101

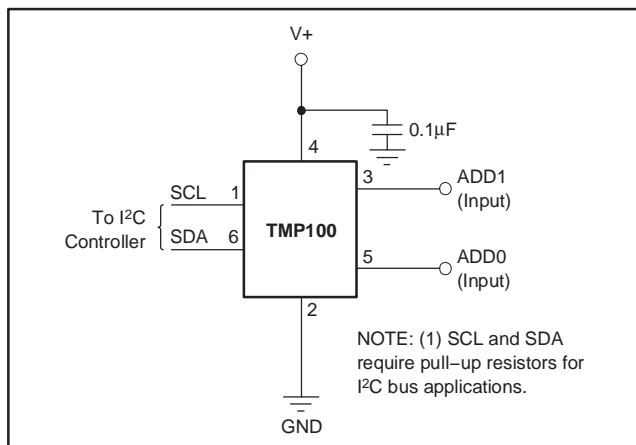


Figure 2. Typical Connections of the TMP100

The die flag of the lead frame is connected to pin 2. The sensing device of the TMP100 and TMP101 is the chip itself. Thermal paths run through the package leads as well as the plastic package. The lower thermal resistance of metal causes the leads to provide the primary thermal path. The GND pin of the TMP100 or TMP101 is directly connected to the metal lead frame, and is the best choice for thermal input.

To maintain the accuracy in applications requiring air or surface temperature measurement, care should be taken to isolate the package and leads from ambient air temperature. A thermally-conductive adhesive will assist in achieving accurate surface temperature measurement.

POINTER REGISTER

Figure 3 shows the internal register structure of the TMP100 and TMP101. The 8-bit Pointer Register of the TMP100 and TMP101 is used to address a given data register. The Pointer Register uses the two LSBs to identify which of the data registers should respond to a read or write command. Table 1 identifies the bits of the Pointer Register byte. Table 2 describes the pointer address of the registers available in the TMP100 and TMP101. Power-up Reset value of P1/P0 is 00.

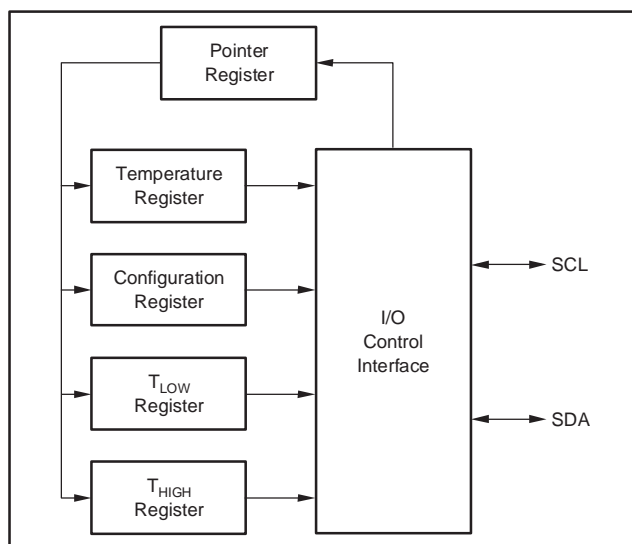


Figure 3. Internal Register Structure of the TMP100 and TMP101

Table 1. Pointer Register Type

P7	P6	P5	P4	P3	P2	P1	P0
0	0	0	0	0	0	Register Bits	

Table 2. Pointer Addresses of the TMP100 and TMP101 Registers

P1	P0	REGISTER
0	0	Temperature Register (READ Only)
0	1	Configuration Register (READ/WRITE)
1	0	T _{LOW} Register (READ/WRITE)
1	1	T _{HIGH} Register (READ/WRITE)

TEMPERATURE REGISTER

The Temperature Register of the TMP100 or TMP101 is a 12-bit read-only register that stores the output of the most recent conversion. Two bytes must be read to obtain data and are described in Table 3 and Table 4. The first 12 bits are used to indicate temperature with all remaining bits

equal to zero. Data format for temperature is summarized in Table 5. Following power-up or reset, the Temperature Register will read 0°C until the first conversion is complete.

Table 3. Byte 1 of Temperature Register

D7	D6	D5	D4	D3	D2	D1	D0
T11	T10	T9	T8	T7	T6	T5	T4

Table 4. Byte 2 of Temperature Register

D7	D6	D5	D4	D3	D2	D1	D0
T3	T2	T1	T0	0	0	0	0

Table 5. Temperature Data Format

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	HEX
128	0111 1111 1111	7FF
127.9375	0111 1111 1111	7FF
100	0110 0100 0000	640
80	0101 0000 0000	500
75	0100 1011 0000	4B0
50	0011 0010 0000	320
25	0001 1001 0000	190
0.25	0000 0000 0100	004
0.0	0000 0000 0000	000
-0.25	1111 1111 1100	FFC
-25	1110 0111 0000	E70
-55	1100 1001 0000	C90
-128	1000 0000 0000	800

The user can obtain 9, 10, 11, or 12 bits of resolution by addressing the Configuration Register and setting the resolution bits accordingly. For 9-, 10-, or 11-bit resolution, the most significant bits in the Temperature Register are used with the unused LSBs set to zero.

CONFIGURATION REGISTER

The Configuration Register is an 8-bit read/write register used to store bits that control the operational modes of the temperature sensor. Read/write operations are performed MSB first. The format of the Configuration Register for the TMP100 and TMP101 is shown in Table 6, followed by a breakdown of the register bits. The power-up/reset value of the Configuration Register is all bits equal to 0. The OS/ALERT bit will read as 1 after power-up/reset.

Table 6. Configuration Register Format

BYTE	D7	D6	D5	D4	D3	D2	D1	D0
1	OS/ALERT	R1	R0	F1	F0	POL	TM	SD

SHUTDOWN MODE (SD)

The Shutdown Mode of the TMP100 and TMP101 allows the user to save maximum power by shutting down all device circuitry other than the serial interface, which reduces current consumption to less than 1µA. For the TMP100 and TMP101, Shutdown Mode is enabled when

the SD bit is 1. The device will shutdown once the current conversion is completed. For SD equal to 0, the device will maintain continuous conversion.

THERMOSTAT MODE (TM)

The Thermostat Mode bit of the TMP101 indicates to the device whether to operate in Comparator Mode (TM = 0) or Interrupt Mode (TM = 1). For more information on comparator and interrupt modes, see the *HIGH and LOW Limit Registers* section.

POLARITY (POL)

The Polarity Bit of the TMP101 allows the user to adjust the polarity of the ALERT pin output. If POL = 0, the ALERT pin will be active LOW, as shown in Figure 4. For POL = 1 the ALERT pin will be active HIGH, and the state of the ALERT pin is inverted.

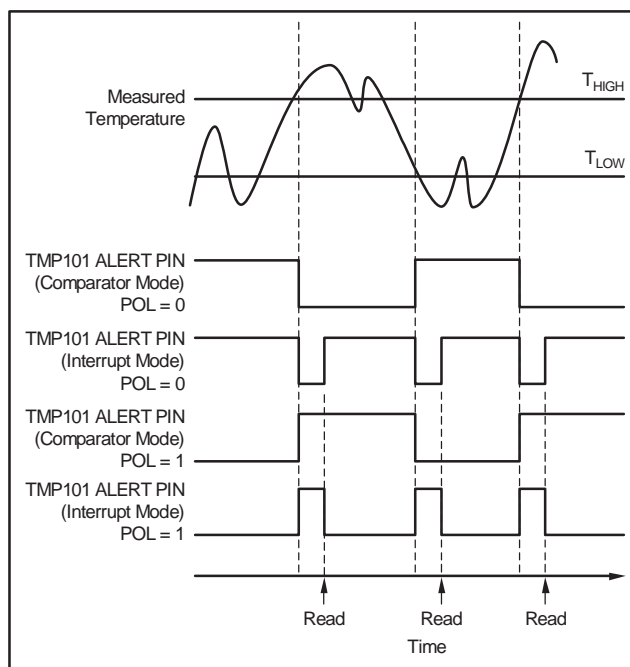


Figure 4. Output Transfer Function Diagrams

FAULT QUEUE (F1/F0)

A fault condition occurs when the measured temperature exceeds the user-defined limits set in the T_{HIGH} and T_{LOW} Registers. Additionally, the number of fault conditions required to generate an alert may be programmed using the Fault Queue. The Fault Queue is provided to prevent a false alert due to environmental noise. The Fault Queue requires consecutive fault measurements in order to trigger the alert function. If the temperature falls below T_{LOW}, prior to reaching the number of programmed consecutive faults limit, the count is reset to 0. Table 7 defines the number of measured faults that may be programmed to trigger an alert condition in the device.

Table 7. Fault Settings of the TMP100 and TMP101

F1	F0	CONSECUTIVE FAULTS
0	0	1
0	1	2
1	0	4
1	1	6

CONVERTER RESOLUTION (R1/R0)

The Converter Resolution Bits control the resolution of the internal Analog-to-Digital (A/D) converter. This allows the user to maximize efficiency by programming for higher resolution or faster conversion time. Table 8 identifies the Resolution Bits and relationship between resolution and conversion time.

Table 8. Resolution of the TMP100 and TMP101

R1	R0	RESOLUTION	CONVERSION TIME (typical)
0	0	9 Bits (0.5°C)	40ms
0	1	10 Bits (0.25°C)	80ms
1	0	11 Bits (0.125°C)	160ms
1	1	12 Bits (0.0625°C)	320ms

OS/ALERT (OS)

The TMP100 and TMP101 feature a One-Shot Temperature Measurement Mode. When the device is in Shutdown Mode, writing a 1 to the OS/ALERT bit will start a single temperature conversion. The device will return to the shutdown state at the completion of the single conversion. This is useful to reduce power consumption in the TMP100 and TMP101 when continuous monitoring of temperature is not required.

Reading the OS/ALERT bit will provide information about the Comparator Mode status. The state of the POL bit will invert the polarity of data returned from the OS/ALERT bit. For POL = 0, the OS/ALERT will read as 1 until the temperature equals or exceeds T_{HIGH} for the programmed number of consecutive faults, causing the OS/ALERT bit to read as 0. The OS/ALERT bit will continue to read as 0 until the temperature falls below T_{LOW} for the programmed number of consecutive faults when it will again read as 1. The status of the TM bit does not affect the status of the OS/ALERT bit.

HIGH AND LOW LIMIT REGISTERS

In Comparator Mode (TM = 0), the ALERT pin of the TMP101 becomes active when the temperature equals or exceeds the value in T_{HIGH} and generates a consecutive number of faults according to fault bits F1 and F0. The ALERT pin will remain active until the temperature falls below the indicated T_{LOW} value for the same number of faults.

In Interrupt Mode (TM = 1) the ALERT Pin becomes active when the temperature equals or exceeds T_{HIGH} for a consecutive number of fault conditions. The ALERT pin remains active until a read operation of any register occurs or the device successfully responds to the SMBus Alert Response Address. The ALERT pin will also be cleared if the device is placed in Shutdown Mode. Once the ALERT pin is cleared, it will only become active again by the temperature falling below T_{LOW} . When the temperature falls below T_{LOW} , the ALERT pin will become active and remain active until cleared by a read operation of any register or a successful response to the SMBus Alert Response Address. Once the ALERT pin is cleared, the above cycle will repeat with the ALERT pin becoming active when the temperature equals or exceeds T_{HIGH} . The ALERT pin can also be cleared by resetting the device with the General Call Reset command. This will also clear the state of the internal registers in the device returning the device to Comparator Mode (TM = 0).

Both operational modes are represented in Figure 4. Table 9 and Table 10 describe the format for the T_{HIGH} and T_{LOW} registers. Power-up Reset values for T_{HIGH} and T_{LOW} are: T_{HIGH} = 80°C and T_{LOW} = 75°C. The format of the data for T_{HIGH} and T_{LOW} is the same as for the Temperature Register.

Table 9. Bytes 1 and 2 of T_{HIGH} Register

BYTE	D7	D6	D5	D4	D3	D2	D1	D0
1	H11	H10	H9	H8	H7	H6	H5	H4
BYTE	D7	D6	D5	D4	D3	D2	D1	D0
2	H3	H2	H1	H0	0	0	0	0

Table 10. Bytes 1 and 2 of T_{LOW} Register

BYTE	D7	D6	D5	D4	D3	D2	D1	D0
1	L11	L10	L9	L8	L7	L6	L5	L4
BYTE	D7	D6	D5	D4	D3	D2	D1	D0
2	L3	L2	L1	L0	0	0	0	0

All 12 bits for the Temperature, T_{HIGH} , and T_{LOW} registers are used in the comparisons for the ALERT function for all converter resolutions. The three LSBs in T_{HIGH} and T_{LOW} can affect the ALERT output even if the converter is configured for 9-bit resolution.

SERIAL INTERFACE

The TMP100 and TMP101 operate only as slave devices on the I²C bus and SMBus. Connections to the bus are made via the open-drain I/O lines SDA and SCL. The TMP100 and TMP101 support the transmission protocol for fast (up to 400kHz) and high-speed (up to 3.4MHz) modes. All data bytes are transmitted most significant bit first.

SERIAL BUS ADDRESS

To program the TMP100 and TMP101, the master must first address slave devices via a slave address byte. The slave address byte consists of seven address bits, and a direction bit indicating the intent of executing a read or write operation.

The TMP100 features two address pins to allow up to eight devices to be addressed on a single I²C interface. Table 11 describes the pin logic levels used to properly connect up to eight devices. *Float* indicates the pin is left unconnected. The state of pins ADD0 and ADD1 is sampled on the first I²C bus communication and should be set prior to any activity on the interface.

Table 11. Address Pins and Slave Addresses for the TMP100

ADD1	ADD0	SLAVE ADDRESS
0	0	1001000
0	Float	1001001
0	1	1001010
1	0	1001100
1	Float	1001101
1	1	1001110
Float	0	1001011
Float	1	1001111

The TMP101 features one address pin and an ALERT pin, allowing up to three devices to be connected per bus. Pin logic levels are described in Table 12. The address pins of the TMP100 and TMP101 are read after reset or in response to an I²C address acquire request. Following reading, the state of the address pins is latched to minimize power dissipation associated with detection.

Table 12. Address Pins and Slave Addresses for the TMP101

ADD0	SLAVE ADDRESS
0	1001000
Float	1001001
1	1001010

BUS OVERVIEW

The device that initiates the transfer is called a *master*, and the devices controlled by the master are *slaves*. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions.

To address a specific device, a START condition is initiated, indicated by pulling the data-line (SDA) from a HIGH to LOW logic level while SCL is HIGH. All slaves on the bus shift in the slave address byte, with the last bit indicating whether a read or write operation is intended. During the ninth clock pulse, the slave being addressed responds to the master by generating an Acknowledge and pulling SDA LOW.

Data transfer is then initiated and sent over eight clock pulses followed by an Acknowledge Bit. During data transfer SDA must remain stable while SCL is HIGH, as any change in SDA while SCL is HIGH will be interpreted as a control signal.

Once all data have been transferred, the master generates a STOP condition indicated by pulling SDA from LOW to HIGH, while SCL is HIGH.

WRITING/READING TO THE TMP100 AND TMP101

Accessing a particular register on the TMP100 and TMP101 is accomplished by writing the appropriate value to the Pointer Register. The value for the Pointer Register is the first byte transferred after the I²C slave address byte with the R/ \overline{W} bit LOW. Every write operation to the TMP100 and TMP101 requires a value for the Pointer Register. (Refer to Figure 6.)

When reading from the TMP100 and TMP101, the last value stored in the Pointer Register by a write operation is used to determine which register is read by a read operation. To change the register pointer for a read operation, a new value must be written to the Pointer Register. This is accomplished by issuing an I²C slave address byte with the R/ \overline{W} bit LOW, followed by the Pointer Register Byte. No additional data are required. The master can then generate a START condition and send the I²C slave address byte with the R/ \overline{W} bit HIGH to initiate the read command. See Figure 7 for details of this sequence. If repeated reads from the same register are desired, it is not necessary to continually send the Pointer Register bytes as the TMP100 and TMP101 will remember the Pointer Register value until it is changed by the next write operation.

SLAVE MODE OPERATIONS

The TMP100 and TMP101 can operate as slave receivers or slave transmitters.

Slave Receiver Mode:

The first byte transmitted by the master is the slave address, with the R/ \overline{W} bit LOW. The TMP100 or TMP101 then acknowledges reception of a valid address. The next byte transmitted by the master is the Pointer Register. The TMP100 or TMP101 then acknowledges reception of the Pointer Register byte. The next byte or bytes are written to the register addressed by the Pointer Register. The TMP100 and TMP101 will acknowledge reception of each data byte. The master may terminate data transfer by generating a START or STOP condition.

Slave Transmitter Mode:

The first byte is transmitted by the master and is the slave address, with the R/ \overline{W} bit HIGH. The slave acknowledges reception of a valid slave address. The next byte is transmitted by the slave and is the most significant byte of the register indicated by the Pointer Register. The master

acknowledges reception of the data byte. The next byte transmitted by the slave is the least significant byte. The master acknowledges reception of the data byte. The master may terminate data transfer by generating a Not-Acknowledge on reception of any data byte, or generating a START or STOP condition.

SMBus ALERT FUNCTION

The TMP101 supports the SMBus Alert function. When the TMP101 is operating in Interrupt Mode (TM = 1), the ALERT pin of the TMP101 may be connected as an SMBus Alert signal. When a master senses that an ALERT condition is present on the ALERT line, the master sends an SMBus Alert command (00011001) on the bus. If the ALERT pin of the TMP101 is active, the TMP101 will acknowledge the SMBus Alert command and respond by returning its slave address on the SDA line. The eighth bit (LSB) of the slave address byte will indicate if the temperature exceeding T_{HIGH} or falling below T_{LOW} caused the ALERT condition. For POL = 0, this bit will be LOW if the temperature is greater than or equal to T_{HIGH} . This bit will be HIGH if the temperature is less than T_{LOW} . The polarity of this bit will be inverted if POL = 1. Refer to Figure 8 for details of this sequence.

If multiple devices on the bus respond to the SMBus Alert command, arbitration during the slave address portion of the SMBus alert command will determine which device will clear its ALERT status. If the TMP101 wins the arbitration, its ALERT pin will become inactive at the completion of the SMBus Alert command. If the TMP101 loses the arbitration, its ALERT pin will remain active.

The TMP100 will also respond to the SMBus ALERT command if its TM bit is set to 1. Since it does not have an ALERT pin, the master needs to periodically poll the device by issuing an SMBus Alert command. If the TMP100 has generated an ALERT, it will acknowledge the SMBus Alert command and return its slave address in the next byte.

GENERAL CALL

The TMP100 and TMP101 respond to the I²C General Call address (0000000) if the eighth bit is 0. The device will acknowledge the General Call address and respond to commands in the second byte. If the second byte is 00000100, the TMP100 and TMP101 will latch the status of their address pins, but will not reset. If the second byte is 00000110, the TMP100 and TMP101 will latch the status of their address pins and reset their internal registers.

POR (POWER-ON RESET)

The TMP100 and TMP101 both have on-chip power-on reset circuits that reset the device to default settings when the device is powered on. This circuit activates when the power supply is less than 0.3V for more than 100ms. If the TMP100 and TMP101 are powered down by removing supply voltage from the device, but the supply voltage is not assured to be less than 0.3V, it is recommended to issue a General Call reset command on the I²C interface bus to ensure that the TMP100 and TMP101 are completely reset.

HIGH-SPEED MODE

In order for the I²C bus to operate at frequencies above 400kHz, the master device must issue an Hs-mode master code (00001XXX) as the first byte after a START condition to switch the bus to high-speed operation. The TMP100 and TMP101 will not acknowledge this byte as required by the I²C specification, but will switch their input filters on SDA and SCL and their output filters on SDA to operate in Hs-mode, allowing transfers at up to 3.4MHz. After the Hs-mode master code has been issued, the master will transmit an I²C slave address to initiate a data transfer operation. The bus will continue to operate in Hs-mode until a STOP condition occurs on the bus. Upon receiving the STOP condition, the TMP100 and TMP101 will switch their input and output filters back to fast-mode operation.

TIMING DIAGRAMS

The TMP100 and TMP101 are I²C and SMBus compatible. Figure 5 to Figure 8 describe the various operations on the TMP100 and TMP101. Bus definitions are given below. Parameters for Figure 5 are defined in Table 13.

Bus Idle: Both SDA and SCL lines remain HIGH.

Start Data Transfer: A change in the state of the SDA line, from HIGH to LOW, while the SCL line is HIGH, defines a START condition. Each data transfer is initiated with a START condition.

Stop Data Transfer: A change in the state of the SDA line from LOW to HIGH while the SCL line is HIGH defines a STOP condition. Each data transfer is terminated with a repeated START or STOP condition.

Data Transfer: The number of data bytes transferred between a START and a STOP condition is not limited and is determined by the master device. The receiver acknowledges the transfer of data.

Acknowledge: Each receiving device, when addressed, is obliged to generate an Acknowledge bit. A device that acknowledges must pull down the SDA line during the Acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the Acknowledge clock pulse. Setup and hold times must be taken into account. On a master receive, the termination of the data transfer can be signaled by the master generating a Not-Acknowledge on the last byte that has been transmitted by the slave.

Table 13. Timing Diagram Definitions

PARAMETER		FAST MODE		HIGH-SPEED MODE		UNITS
		MIN	MAX	MIN	MAX	
SCLK Operating Frequency	f(SCLK)		0.4		3.4	MHz
Bus Free Time Between STOP and START Conditions	t(BUF)	600		160		ns
Hold time after repeated START condition. After this period, the first clock is generated.	t(HDSTA)	600		160		ns
Repeated START Condition Setup Time	t(SUSTA)	600		160		ns
STOP Condition Setup Time	t(SUSTO)	600		160		ns
Data HOLD Time	t(HDDAT)	0		0		ns
Data Setup Time	t(SUDAT)	100		10		ns
SCLK Clock LOW Period	t(LOW)	1300		160		ns
SCLK Clock HIGH Period	t(HIGH)	600		60		ns
Clock/Data Fall Time	t _F		300		160	ns
Clock/Data Rise Time	t _R		300		160	ns
for SCLK ≤ 100kHz	t _R		1000			ns

I²C TIMING DIAGRAMS

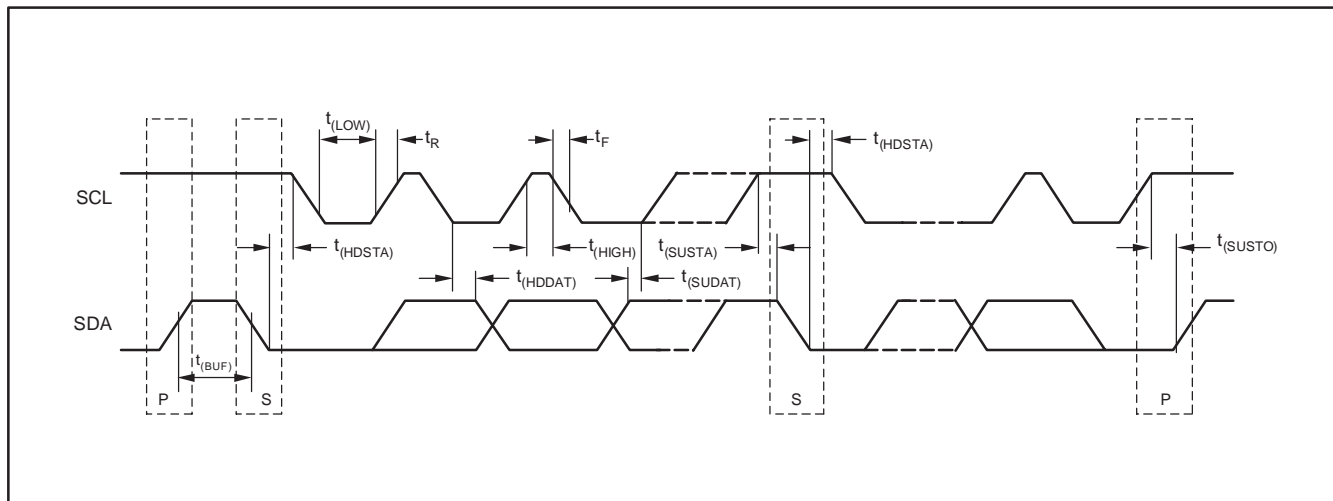


Figure 5. I²C Timing Diagram

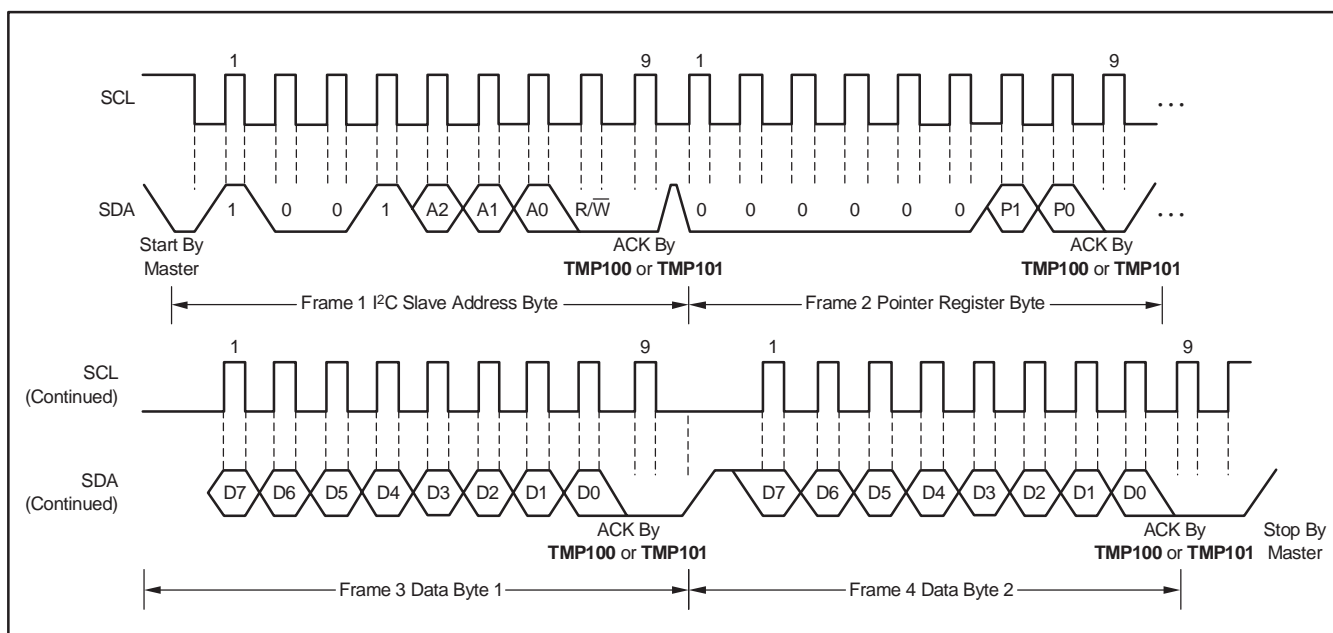


Figure 6. I²C Timing Diagram for Write Word Format

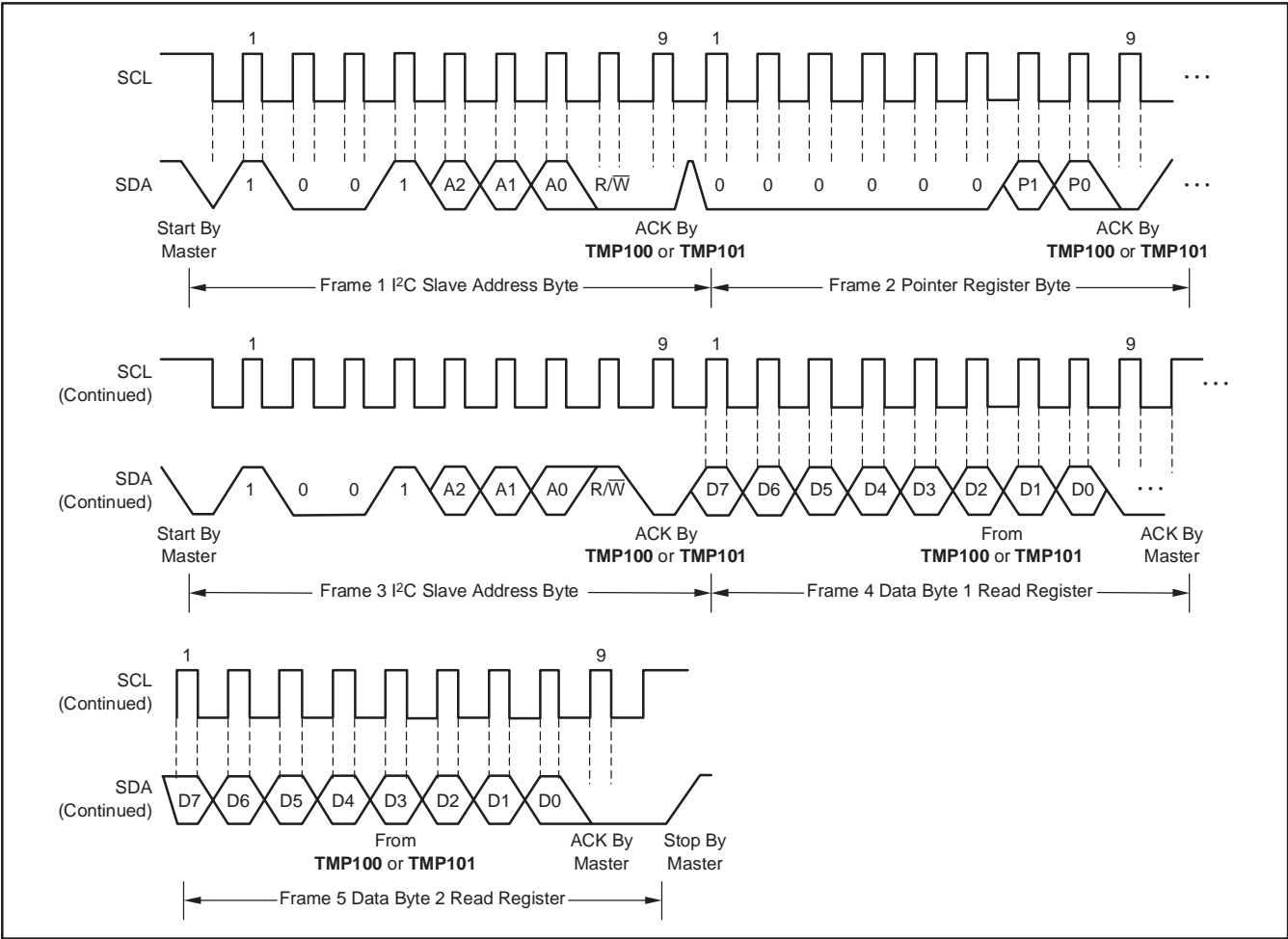


Figure 7. I2C Timing Diagram for Read Word Format

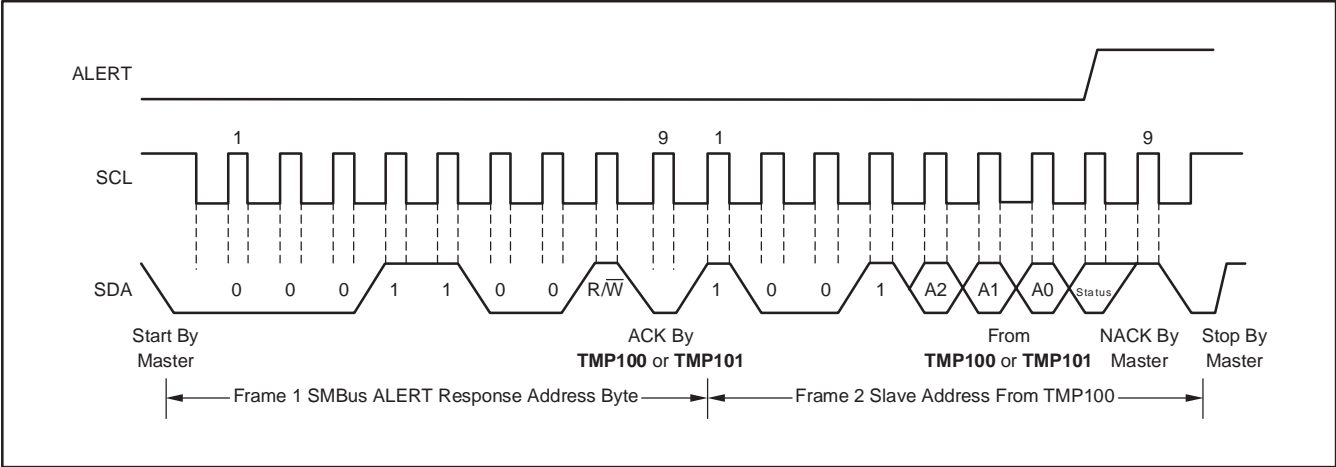


Figure 8. Timing Diagram for SMBus ALERT

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish	MSL Peak Temp (3)	Op Temp (°C)	Top-Side Markings (4)	Samples
SN0312100DBVR	ACTIVE	SOT-23	DBV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-55 to 125	T100	Samples
TMP100NA/250	ACTIVE	SOT-23	DBV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR		T100	Samples
TMP100NA/250G4	ACTIVE	SOT-23	DBV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR		T100	Samples
TMP100NA/3K	ACTIVE	SOT-23	DBV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-55 to 125	T100	Samples
TMP100NA/3KG4	ACTIVE	SOT-23	DBV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-55 to 125	T100	Samples
TMP101NA/250	ACTIVE	SOT-23	DBV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-55 to 125	T101	Samples
TMP101NA/250G4	ACTIVE	SOT-23	DBV	6	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-55 to 125	T101	Samples
TMP101NA/3K	ACTIVE	SOT-23	DBV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-55 to 125	T101	Samples
TMP101NA/3KG4	ACTIVE	SOT-23	DBV	6	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR	-55 to 125	T101	Samples

(1) The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.

LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

PREVIEW: Device has been announced but is not in production. Samples may or may not be available.

OBSELETE: TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check <http://www.ti.com/productcontent> for the latest availability information and additional product content details.

TBD: The Pb-Free/Green conversion plan has not been defined.

Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.

Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.

Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

⁽³⁾ MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

⁽⁴⁾ Multiple Top-Side Markings will be inside parentheses. Only one Top-Side Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Top-Side Marking for that device.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

OTHER QUALIFIED VERSIONS OF TMP100, TMP101 :

- Automotive: [TMP101-Q1](#)
- Enhanced Product: [TMP100-EP](#)

NOTE: Qualified Version Definitions:

- Automotive - Q100 devices qualified for high-reliability automotive applications targeting zero defects
- Enhanced Product - Supports Defense, Aerospace and Medical Applications

TAPE AND REEL INFORMATION


*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
TMP100NA/250	SOT-23	DBV	6	250	178.0	9.0	3.23	3.17	1.37	4.0	8.0	Q3
TMP100NA/3K	SOT-23	DBV	6	3000	178.0	9.0	3.23	3.17	1.37	4.0	8.0	Q3

TAPE AND REEL BOX DIMENSIONS

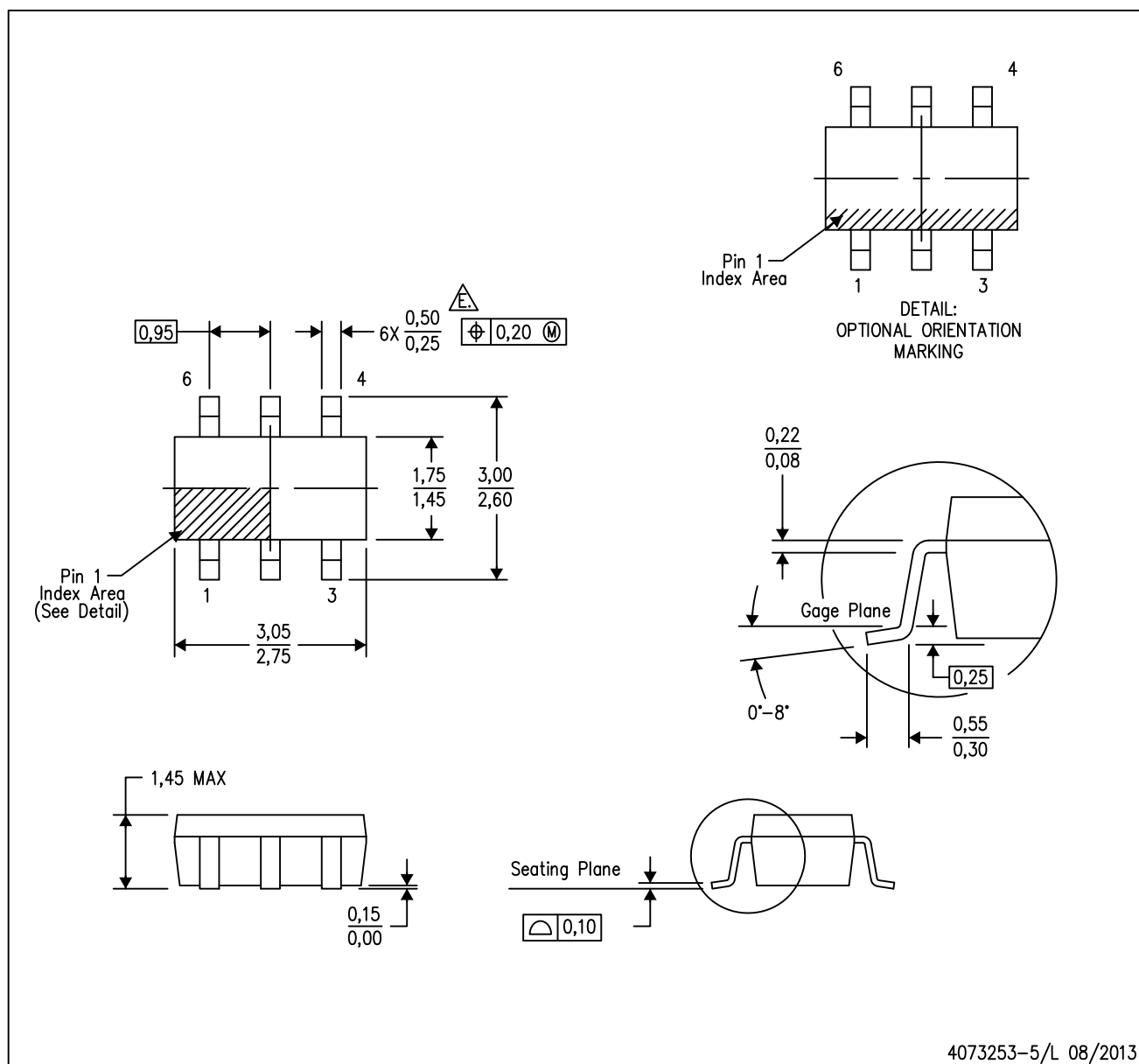


*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
TMP100NA/250	SOT-23	DBV	6	250	180.0	180.0	18.0
TMP100NA/3K	SOT-23	DBV	6	3000	180.0	180.0	18.0

DBV (R-PDSO-G6)

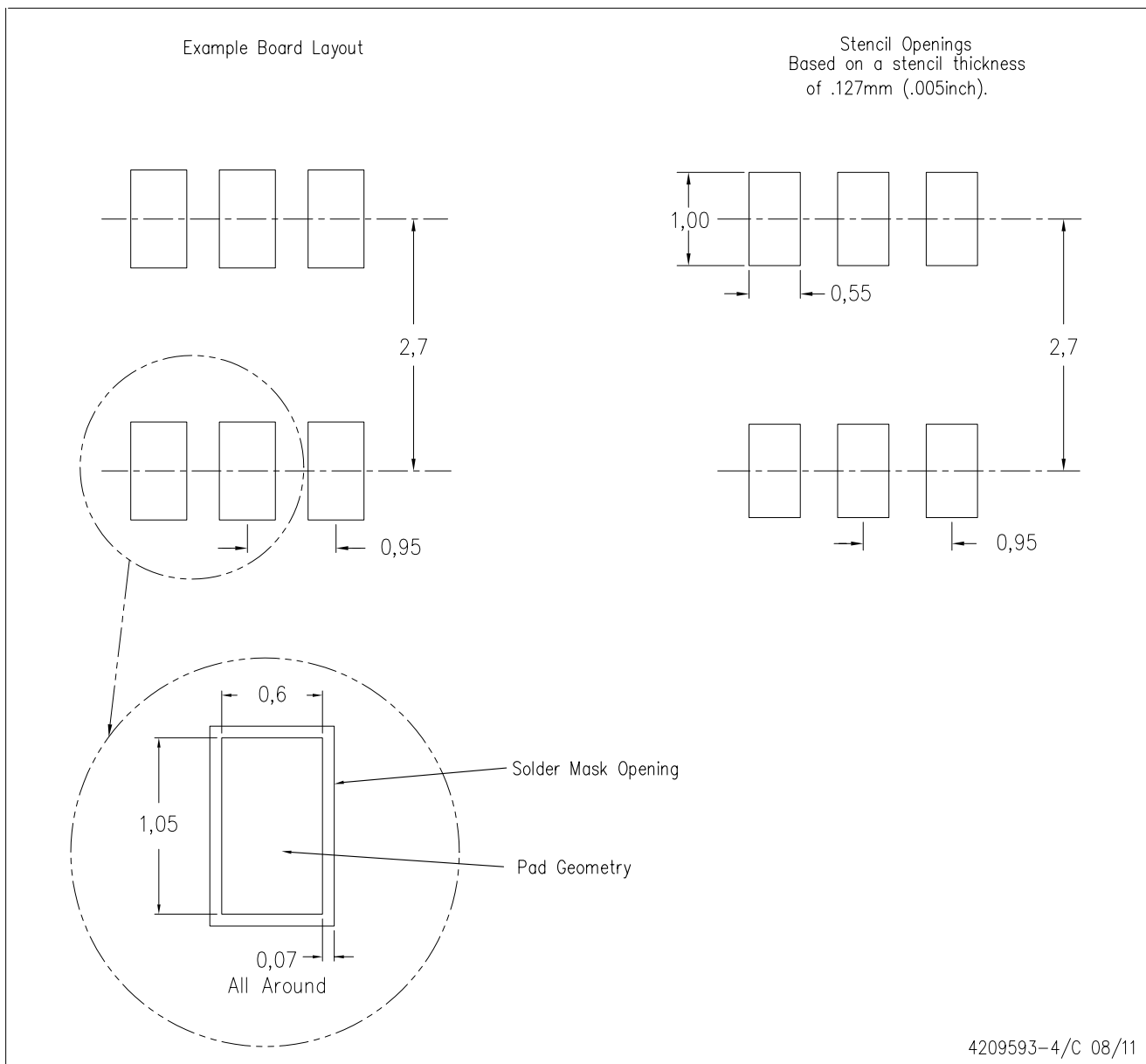
PLASTIC SMALL-OUTLINE PACKAGE



- NOTES:
- A. All linear dimensions are in millimeters.
 - B. This drawing is subject to change without notice.
 - C. Body dimensions do not include mold flash or protrusion. Mold flash and protrusion shall not exceed 0.15 per side.
 - D. Leads 1,2,3 may be wider than leads 4,5,6 for package orientation.
 - E. Falls within JEDEC MO-178 Variation AB, except minimum lead width.

DBV (R-PDSO-G6)

PLASTIC SMALL OUTLINE



- NOTES:
- A. All linear dimensions are in millimeters.
 - B. This drawing is subject to change without notice.
 - C. Customers should place a note on the circuit board fabrication drawing not to alter the center solder mask defined pad.
 - D. Publication IPC-7351 is recommended for alternate designs.
 - E. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Example stencil design based on a 50% volumetric metal load solder paste. Refer to IPC-7525 for other stencil recommendations.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

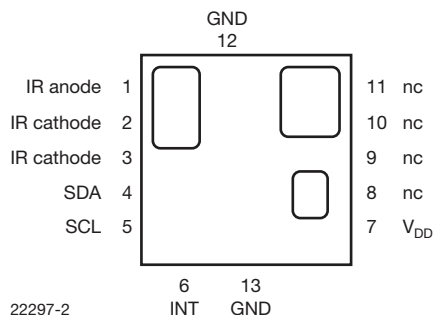
Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community

e2e.ti.com

Fully Integrated Proximity and Ambient Light Sensor with Infrared Emitter, I²C Interface, and Interrupt Function



DESCRIPTION

The VCNL4010 is a fully integrated proximity and ambient light sensor. Fully integrated means that the infrared emitter is included in the package. It has 16 bit resolution. It includes a signal processing IC and features standard I²C communication interface. It features an interrupt function.

APPLICATIONS

- Proximity sensor for mobile devices (e.g. smart phones, touch phones, PDA, GPS) for touch screen locking, power saving, etc.
- Integrated ambient light function for display/keypad contrast control and dimming of mobile devices
- Proximity/optical switch for consumer, computing and industrial devices and displays
- Dimming control for consumer, computing and industrial displays

FEATURES

- Package type: surface mount
- Dimensions (L x W x H in mm): 3.95 x 3.95 x 0.75
- Integrated infrared emitter, ambient light sensor, proximity sensor, and signal conditioning IC
- Interrupt function
- Supply voltage range V_{DD}: 2.5 V to 3.6 V
- Supply voltage range IR anode: 2.5 V to 5 V
- Communication via I²C interface
- I²C Bus H-level range: 1.7 V to 5 V
- Floor life: 168 h, MSL 3, acc. J-STD-020
- Low stand by current consumption: 1.5 μ A
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912



PROXIMITY FUNCTION

- Built-in infrared emitter and photo-pin-diode for proximity function
- 16 bit effective resolution for proximity detection range ensures excellent cross talk immunity
- Programmable LED drive current from 10 mA to 200 mA in 10 mA steps
- Excellent ambient light suppression by modulating the infrared signal
- Proximity distance up to 200 mm

AMBIENT LIGHT FUNCTION

- Built-in ambient light photo-pin-diode with close-to-human-eye sensitivity
- 16 bit dynamic range from 0.25 lx to 16 klx
- 100 Hz and 120 Hz flicker noise rejection

PRODUCT SUMMARY

PART NUMBER	OPERATING RANGE (mm)	OPERATING VOLTAGE RANGE (V)	I ² C BUS VOLTAGE RANGE (V)	LED PULSE CURRENT ⁽¹⁾ (mA)	AMBIENT LIGHT RANGE (lx)	AMBIENT LIGHT RESOLUTION (lx)	OUTPUT CODE
VCNL4010	1 to 200	2.5 to 3.6	1.7 to 5	10 to 200	0.25 to 16 383	0.25	16 bit, I ² C

Note

⁽¹⁾ Adjustable through I²C interface

**ORDERING INFORMATION**

ORDERING CODE	PACKAGING	VOLUME ⁽¹⁾	REMARKS
VCNL4010-GS08	Tape and reel	MOQ: 1800 pcs	3.95 mm x 3.95 mm x 0.75 mm
VCNL4010-GS18		MOQ: 7000 pcs	
VCNL4000Demokit ⁽²⁾	-	MOQ: 1 pc	-

Notes

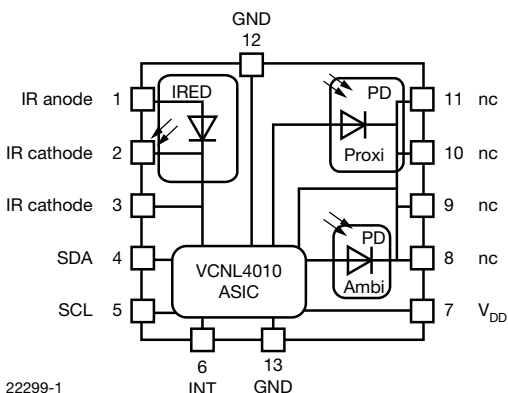
- (1) MOQ: minimum order quantity
- (2) VCNL4000 Demokit provides USB dongle, basic software including Vishay licence. The VCNL4010 sensor board could be ordered free of charge by contacting sensorstechsupport@vishay.com. Software updates for VCNL4010 can be downloaded from our web site: www.vishay.com/???/

ABSOLUTE MAXIMUM RATINGS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)

PARAMETER	TEST CONDITION	SYMBOL	MIN.	MAX.	UNIT
Supply voltage		V_{DD}	- 0.3	5.5	V
Operation temperature range		T_{amb}	- 25	+ 85	$^{\circ}\text{C}$
Storage temperature range		T_{stg}	- 40	+ 85	$^{\circ}\text{C}$
Total power dissipation	$T_{amb} \leq 25\text{ }^{\circ}\text{C}$	P_{tot}		50	mW
Junction temperature		T_j		100	$^{\circ}\text{C}$

BASIC CHARACTERISTICS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)

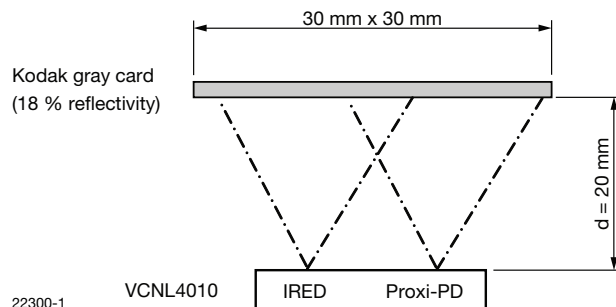
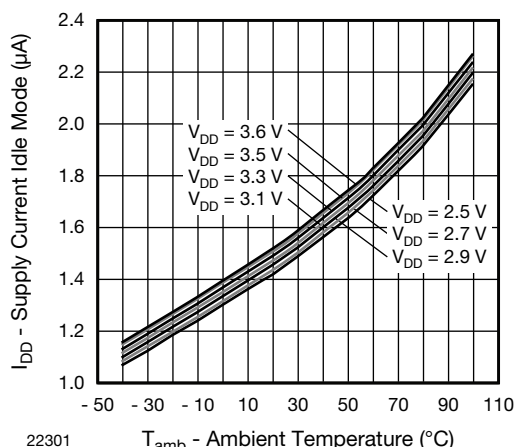
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Supply voltage V_{DD}			2.5		3.6	V
Supply voltage IR anode			2.5		5	V
I ² C Bus H-level range			1.7		5	V
INT H-level range			1.7		5	V
INT low voltage	3 mA sink current				0.4	V
Current consumption	Standby current, no IRED-operation			1.5	2	μA
Current consumption proximity mode incl. IRED (averaged)	2 measurements per second, IRED current 20 mA			5		μA
	250 measurements per second, IRED current 20 mA			520		μA
	2 measurements per second, IRED current 200 mA			35		μA
	250 measurements per second, IRED current 200 mA			4.0		mA
Current consumption ambient light mode	2 measurements per second averaging = 1			2.5		μA
	8 measurements per second averaging = 1			10		μA
	2 measurements per second averaging = 64			160		μA
	8 measurements per second averaging = 64			640		μA
Ambient light resolution	Digital resolution (LSB count)			0.25		lx
Ambient light output	$E_V = 100\text{ lx}$ averaging = 64			400		counts
I ² C clock rate range		f_{SCL}			3400	kHz

CIRCUIT BLOCK DIAGRAM


22299-1

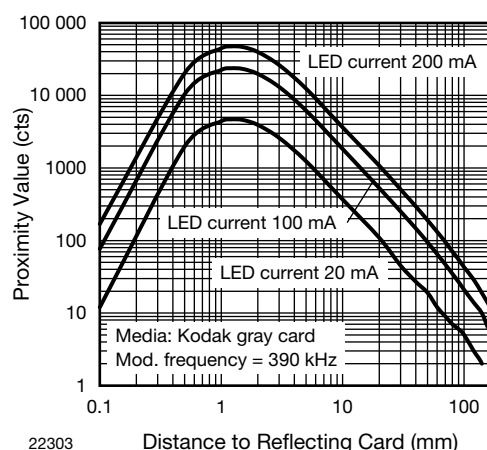
Note

- nc must not be electrically connected
Pads 8 to 11 are only considered as solder pads

TEST CIRCUIT

BASIC CHARACTERISTICS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)


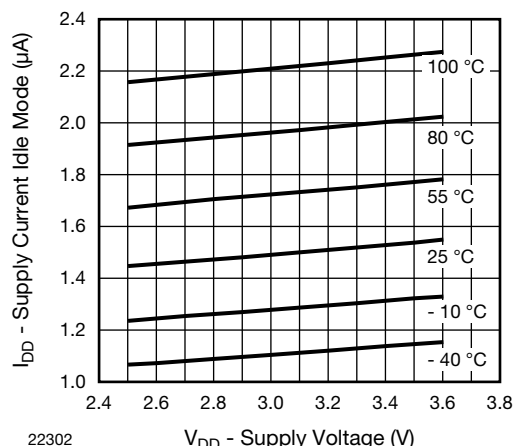
22301

Fig. 1 - Idle Current vs. Ambient Temperature

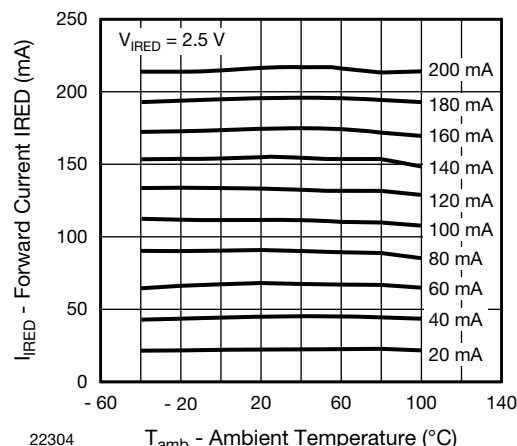


22303

Fig. 3 - Proximity Value vs. Distance



22302

Fig. 2 - Idle Current vs. V_{DD}


22304

Fig. 4 - Forward Current vs. Temperature

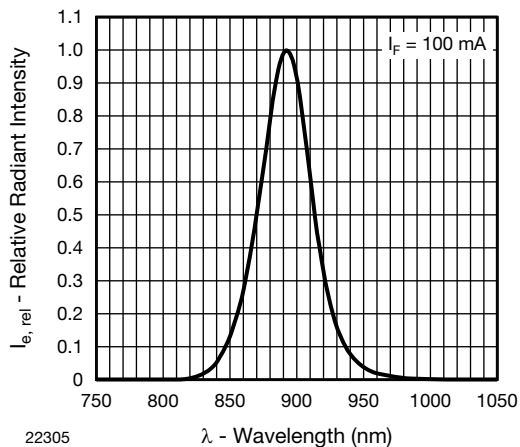


Fig. 5 - Relative Radiant Intensity vs. Wavelength

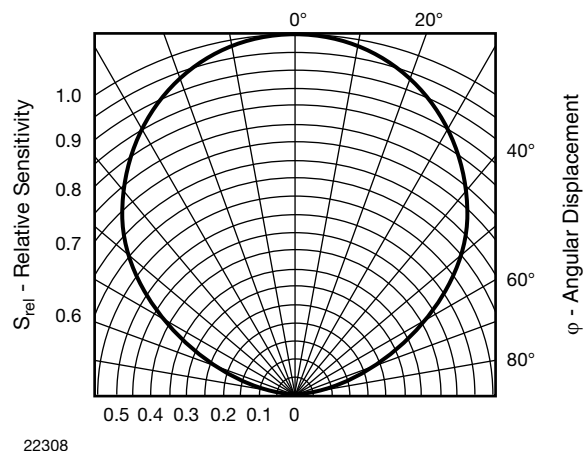


Fig. 8 - Relative Radiant Sensitivity vs. Angular Displacement (Proximity Sensor)

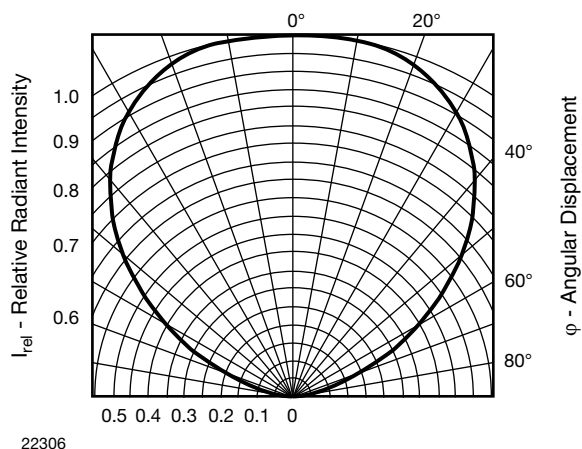


Fig. 6 - Relative Radiant Intensity vs. Angular Displacement

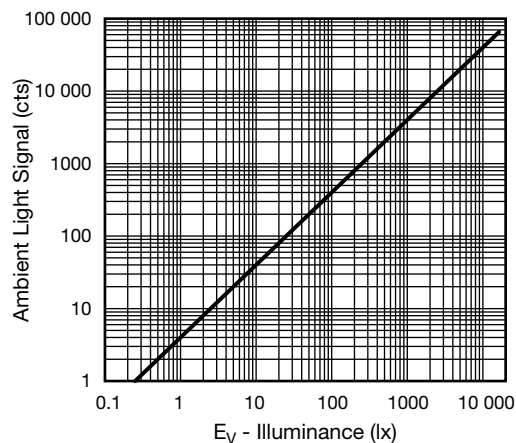


Fig. 9 - Ambient Light Value vs. Illuminance

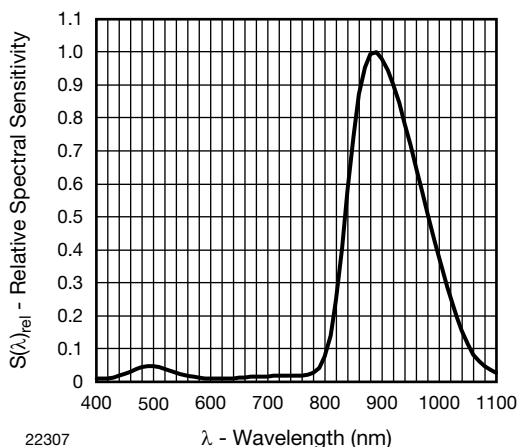


Fig. 7 - Relative Spectral Sensitivity vs. Wavelength (Proximity Sensor)

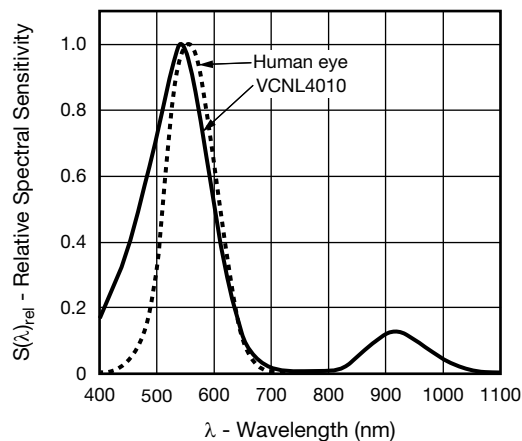
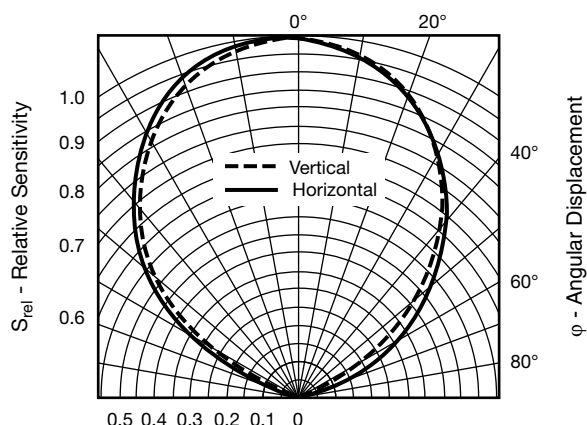


Fig. 10 - Relative Spectral Sensitivity vs. Wavelength (Ambient Light Sensor)



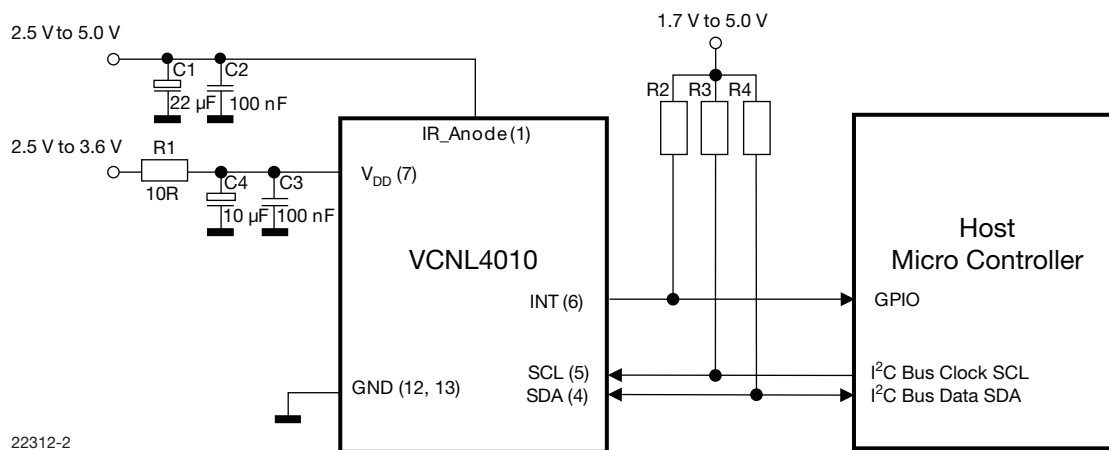
22311

Fig. 11 - Relative Radiant Sensitivity vs. Angular Displacement
(Ambient Light Sensor)

APPLICATION INFORMATION

VCNL4010 is a cost effective solution of proximity and ambient light sensor with I²C bus interface. The standard serial digital interface is easy to access “Proximity Signal” and “Light Intensity” without complex calculation and programming by external controller. Beside the digital output also a flexible programmable interrupt pin is available.

1. Application Circuit



22312-2

Fig. 12 - Application Circuit
(x) = Pin Number

Note

- The interrupt pin is an open drain output. The needed pull-up resistor may be connected to the same supply voltage as the application controller and the pull-up resistors at SDA/SCL. Proposed value R2 should be >1 kΩ, e.g. 10 kΩ to 100 kΩ. Proposed value for R3 and R4, e.g. 2.2 kΩ to 4.7 kΩ, depend also on the I²C bus speed. For detailed description about set-up and use of the interrupt as well as more application related information see AN: “Designing VCNL4010 into an Application”.

2. I²C Interface

The VCNL4010 contains seventeen 8 bit registers for operation control, parameter setup and result buffering. All registers are accessible via I²C communication. Figure 13 shows the basic I²C communication with VCNL4010.

The built in I²C interface is compatible with all I²C modes (standard, fast and high speed).

I²C H-level range = 1.7 V to 5 V.

Please refer to the I²C specification from NXP for details.

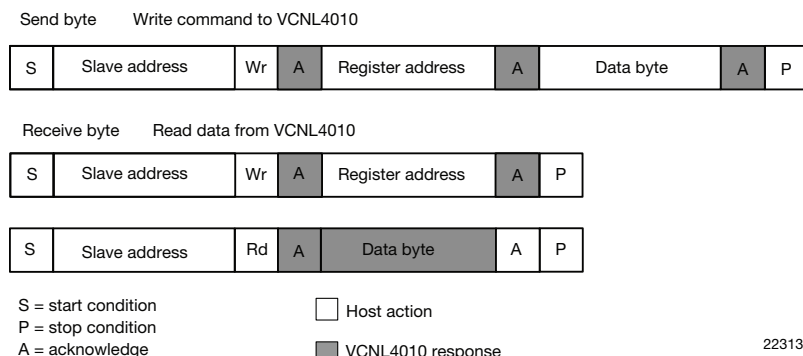


Fig. 13 - Send Byte/Receive Byte Protocol

Device Address

The VCNL4010 has a fix slave address for the host programming and accessing selection. The predefined 7 bit I²C bus address is set to 0010 011 = 13h. The least significant bit (LSB) defines read or write mode. Accordingly the bus address is set to 0010 011x = 26h for write, 27h for read.

Register Addresses

VCNL4010 has seventeen user accessible 8 bit registers. The register addresses are 80h (register #0) to 90h (register #16).

REGISTER FUNCTIONS

Register #0 Command Register

Register address = 80h

The register #0 is for starting ambient light or proximity measurements. This register contains 2 flag bits for data ready indication.

TABLE 1 - COMMAND REGISTER #0							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
config_lock	als_data_rdy	prox_data_rdy	als_od	prox_od	als_en	prox_en	selftimed_en
Description							
config_lock	Read only bit. Value = 1						
als_data_rdy	Read only bit. Value = 1 when ambient light measurement data is available in the result registers. This bit will be reset when one of the corresponding result registers (reg #5, reg #6) is read.						
prox_data_rdy	Read only bit. Value = 1 when proximity measurement data is available in the result registers. This bit will be reset when one of the corresponding result registers (reg #7, reg #8) is read.						
als_od	R/W bit. Starts a single on-demand measurement for ambient light. If averaging is enabled, starts a sequence of readings and stores the averaged result. Result is available at the end of conversion for reading in the registers #5(HB) and #6(LB).						
prox_od	R/W bit. Starts a single on-demand measurement for proximity. Result is available at the end of conversion for reading in the registers #7(HB) and #8(LB).						
als_en	R/W bit. Enables periodic als measurement						
prox_en	R/W bit. Enables periodic proximity measurement						
selftimed_en	R/W bit. Enables state machine and LP oscillator for self timed measurements; no measurement is performed until the corresponding bit is set						

Note

- With setting bit 3 and bit 4 at the same write command, a simultaneously measurement of ambient light and proximity is done. Beside als_en and/or prox_en first selftimed_en needs to be set. On-demand measurement modes are disabled if selftimed_en bit is set. For the selftimed_en mode changes in reading rates (reg #4 and reg #2) can be made only when b0 (selftimed_en bit) = 0. For the als_od mode changes to the reg #4 can be made only when b4 (als_od bit) = 0; this is to avoid synchronization problems and undefined states between the clock domains. In effect this means that it is only reasonable to change rates while no selftimed conversion is ongoing.

**Register #1 Product ID Revision Register**

Register address = 81h. This register contains information about product ID and product revision.

Register data value of current revision = 21h.

TABLE 2 - PRODUCT ID REVISION REGISTER #1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Product ID				Revision ID			
Description							
Product ID		Read only bits. Value = 2					
Revision ID		Read only bits. Value = 1					

Register #2 Rate of Proximity Measurement

Register address = 82h.

TABLE 3 - PROXIMITY RATE REGISTER #2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a					Rate of Proximity Measurement (no. of measurements per second)		
Description							
Proximity rate		R/W bits. 000 - 1.95 measurements/s (DEFAULT) 001 - 3.90625 measurements/s 010 - 7.8125 measurements/s 011 - 16.625 measurements/s 100 - 31.25 measurements/s 101 - 62.5 measurements/s 110 - 125 measurements/s 111 - 250 measurements/s					

Note

- If self_timed measurement is running, any new value written in this register will not be taken over until the mode is actually cycled.

Register #3 LED Current Setting for Proximity Mode

Register address = 83h. This register is to set the LED current value for proximity measurement.

The value is adjustable in steps of 10 mA from 0 mA to 200 mA.

This register also contains information about the used device fuse program ID.

TABLE 4 - IR LED CURRENT REGISTER #3

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Fuse prog ID		IR LED current value					
Description							
Fuse prog ID		Read only bits. Information about fuse program revision used for initial setup/calibration of the device.					
IR LED current value		R/W bits. IR LED current = Value (dec.) x 10 mA. Valid Range = 0 to 20d. e.g. 0 = 0 mA , 1 = 10 mA, ..., 20 = 200 mA (2 = 20 mA = DEFAULT) LED Current is limited to 200 mA for values higher as 20d.					

**Register #4 Ambient Light Parameter Register**

Register address = 84h.

TABLE 5 - AMBIENT LIGHT PARAMETER REGISTER #4

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Cont. conv. mode	als_rate			Auto offset compensation	Averaging function (number of measurements per run)		
Description							
Cont. conversion mode		R/W bit. Continuous conversion mode. Enable = 1; Disable = 0 = DEFAULT This function can be used for performing faster ambient light measurements. Please refer to the application information chapter 3.3 for details about this function.					
Ambient light measurement rate		R/W bits. Ambient light measurement rate 000 - 1 samples/s 001 - 2 samples/s = DEFAULT 010 - 3 samples/s 011 - 4 samples/s 100 - 5 samples/s 101 - 6 samples/s 110 - 8 samples/s 111 - 10 samples/s					
Auto offset compensation		R/W bit. Automatic offset compensation. Enable = 1 = DEFAULT; Disable = 0 In order to compensate a technology, package or temperature related drift of the ambient light values there is a built in automatic offset compensation function. With active auto offset compensation the offset value is measured before each ambient light measurement and subtracted automatically from actual reading.					
Averaging function		R/W bits. Averaging function. Bit values sets the number of single conversions done during one measurement cycle. Result is the average value of all conversions. Number of conversions = 2 ^{decimal_value} e.g. 0 = 1 conv., 1 = 2 conv, 2 = 4 conv.,7 = 128 conv. DEFAULT = 32 conv. (bit 2 to bit 0: 101)					

Note

- If self_timed measurement is running, any new value written in this register will not be taken over until the mode is actually cycled.

Register #5 and #6 Ambient Light Result Register

Register address = 85h and 86h. These registers are the result registers for ambient light measurement readings.

The result is a 16 bit value. The high byte is stored in register #5 and the low byte in register #6.

TABLE 6 - AMBIENT LIGHT RESULT REGISTER #5

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description							
Read only bits. High byte (15:8) of ambient light measurement result							

TABLE 7 - AMBIENT LIGHT RESULT REGISTER #6

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description							
Read only bits. Low byte (7:0) of ambient light measurement result							

**Register #7 and #8 Proximity Measurement Result Register**

Register address = 87h and 88h. These registers are the result registers for proximity measurement readings.

The result is a 16 bit value. The high byte is stored in register #7 and the low byte in register #8.

TABLE 8 - PROXIMITY RESULT REGISTER #7

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description							
Read only bits. High byte (15:8) of proximity measurement result							

TABLE 9 - PROXIMITY RESULT REGISTER #8

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description							
Read only bits. Low byte (7:0) of proximity measurement result							

Register #9 Interrupt Control Register

Register address = 89h.

TABLE 10 - INTERRUPT CONTROL REGISTER #9

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Int count exceed			n/a	INT_PROX_ready_EN	INT_ALS_ready_EN	INT_THRES_EN	INT_THRES_SEL
Description							
Int count exceed		R/W bits. These bits contain the number of consecutive measurements needed above/below the threshold 000 - 1 count = DEFAULT 001 - 2 count 010 - 4 count 011 - 8 count 100 - 16 count 101 - 32 count 110 - 64 count 111 - 128 count					
INT_PROX_ready_EN		R/W bit. Enables interrupt generation at proximity data ready					
INT_ALS_ready_EN		R/W bit. Enables interrupt generation at ambient data ready					
INT_THRES_EN		R/W bit. Enables interrupt generation when high or low threshold is exceeded					
INT_THRES_SEL		R/W bit. If 0: thresholds are applied to proximity measurements If 1: thresholds are applied to als measurements					

**Register #10 and #11 Low Threshold**

Register address = 8Ah and 8Bh. These registers contain the low threshold value. The value is a 16 bit word. The high byte is stored in register #10 and the low byte in register #11.

TABLE 11 - LOW THRESHOLD REGISTER #10

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description							
R/W bits. High byte (15:8) of low threshold value							

TABLE 12 - LOW THRESHOLD REGISTER #11

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description							
R/W bits. Low byte (7:0) of low threshold value							

Register #12 and #13 High Threshold

Register address = 8Ch and 8Dh. These registers contain the high threshold value. The value is a 16 bit word. The high byte is stored in register #12 and the low byte in register #13.

TABLE 13 - HIGH THRESHOLD REGISTER #12

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description							
R/W bits. High byte (15:8) of high threshold value							

TABLE 14 - HIGH THRESHOLD REGISTER #13

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Description							
R/W bits. Low byte (7:0) of high threshold value							

Register #14 Interrupt Status Register

Register address = 8Eh. This register contains information about the interrupt status for either proximity or ALS function and indicates if high or low going threshold exceeded.

TABLE 15 - INTERRUPT STATUS REGISTER #14

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a				int_prox_ready	int_als_ready	int_th_low	int_th_hi
Description							
int_prox_ready		R/W bit. Indicates a generated interrupt for proximity					
int_als_ready		R/W bit. Indicates a generated interrupt for als					
int_th_low		R/W bit. Indicates a low threshold exceed					
int_th_hi		R/W bit. Indicates a high threshold exceed					

Note

- Once an interrupt is generated the corresponding status bit goes to 1 and stays there unless it is cleared by writing a 1 in the corresponding bit. The int pad will be pulled down while at least one of the status bit is 1.

Register #15 Proximity Modulator Timing Adjustment

Register address = 8Fh.

TABLE 16 - PROXIMITY MODULATOR TIMING ADJUSTMENT #15							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Modulation delay time			Proximity frequency		Modulation dead time		
Description							
Modulation delay time		R/W bits. Setting a delay time between IR LED signal and IR input signal evaluation. This function is for compensation of delays from IR LED and IR photo diode. Also in respect to the possibility for setting different proximity signal frequency. Correct adjustment is optimizing measurement signal level. (DEFAULT = 0)					
Proximity frequency		R/W bits. Setting the proximity IR test signal frequency The proximity measurement is using a square IR signal as measurement signal. Four different values are possible: 00 = 390.625 kHz (DEFAULT) 01 = 781.25 kHz 10 = 1.5625 MHz 11 = 3.125 MHz					
Modulation dead time		R/W bits. Setting a dead time in evaluation of IR signal at the slopes of the IR signal. (DEFAULT = 1) This function is for reducing of possible disturbance effects. This function is reducing signal level and should be used carefully.					

Note

- The settings for best performance will be provided by Vishay. With first samples this is evaluated to:
Delay Time = 0 ; Dead Time = 1 and Prox Frequency = 0 . With that register#15 should be programmed with 1 (= default value).

Register #16 Ambient IR Light Level Register

Register address = 90h.

This register is not intended to be used by customer.

3. IMPORTANT APPLICATION HINTS AND EXAMPLES
3.1 Receiver standby mode

In standby mode the receiver has the lowest current consumption of about 1.5 μ A. In this mode only the I²C interface is active. This is always valid, when there are no measurement demands for proximity and ambient light executed. Also the current sink for the IR-LED is inactive, so there is no need for changing register #3 (IR LED current).

3.2 Data Read

In order to get a certain register value, the register has to be addressed without data like shown in the following scheme. After this register addressing, the data from the addressed register is written after a subsequent read command.

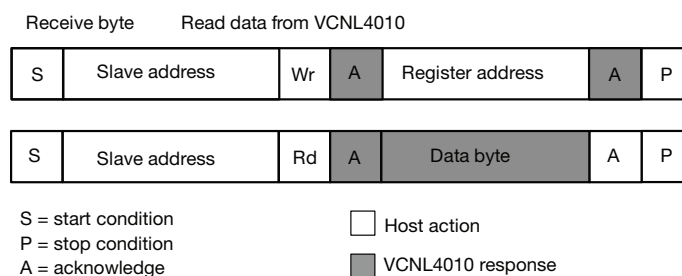


Fig. 14 - Send Byte/Receive Byte Protocol

The stop condition between these write and read sequences is not mandatory. It works also with a repeated start condition.

Note

- For reading out 2 (or more) subsequent registers like the result registers, it is not necessary to address each of the registers separately. After one read command the internal register counter is increased automatically and any subsequent read command is accessing the next register.

Example: read register “Ambient Light Result Register” #5 and #6:

Addressing:command: 26h, 85h (VCNL4010_I²C_Bus_Write_Adr., Ambient Light Result Register #5 [85])

Read register #5:command: 27h, data (VCNL4010_I²C_Bus_Read_Adr., {High Byte Data of Ambient Light Result register #5 [85]})

Read register #6:command: 27h, data (VCNL4010_I²C_Bus_Read_Adr., {Low Byte Data of Ambient Light Result register #6 [86]})

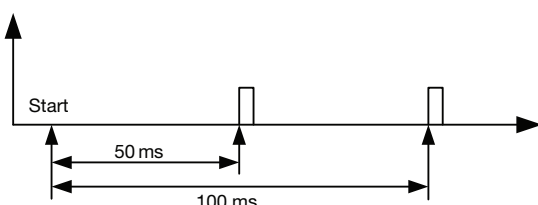
3.3 Continuous Conversion Mode in Ambient Light Measurement

In the following is a detail description of the function “continuous conversion” (bit 7 of register #4)

Standard mode (bit 7 of reg #4 = 0):

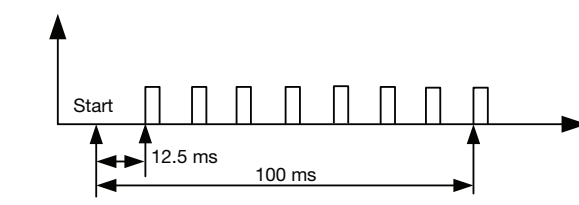
In standard mode the ambient light measurement is done during a fixed time frame of 100 ms. The single measurement itself takes actually only appr. 300 μ s.

The following figures show examples of this measurement timing in standard mode using averaging function 2 and 8 as examples for illustration (possible values up to 128).



22315

Fig. 15 - Ambient Light Measurement with Averaging = 2;
Final Measurement Result = Average of these 2 Measurements



22316

Fig. 16 - Ambient Light Measurement with Averaging = 8;
Final Measurement Result = Average of these 8 Measurements

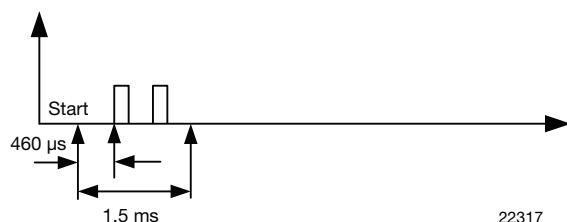
Note

- \geq Independent of setting of averaging the result is available only after 100 ms.

Continuous conversion mode (bit 7 of register #4 = 1):

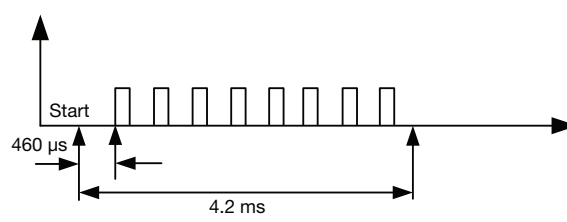
In continuous conversion mode the single measurements are done directly subsequent after each other.

See following examples in figure 17 and 18



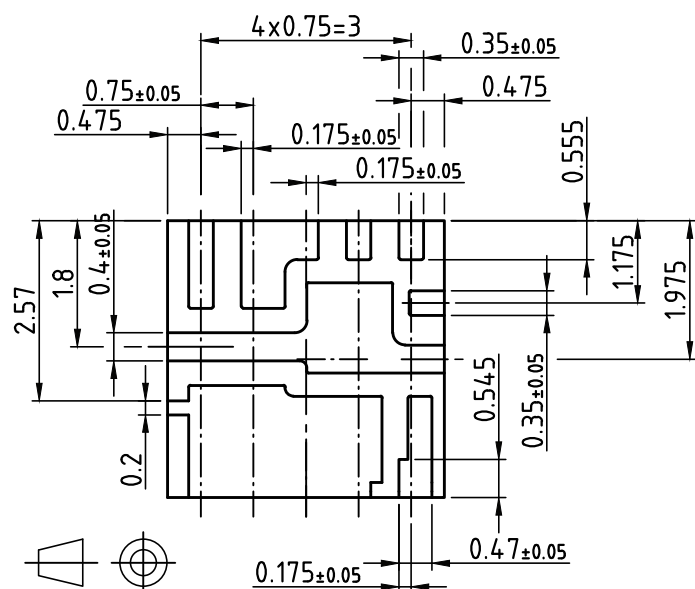
22317

Fig. 17 - Ambient Light Measurement with Averaging = 2;
using Continuous Conversion Mode

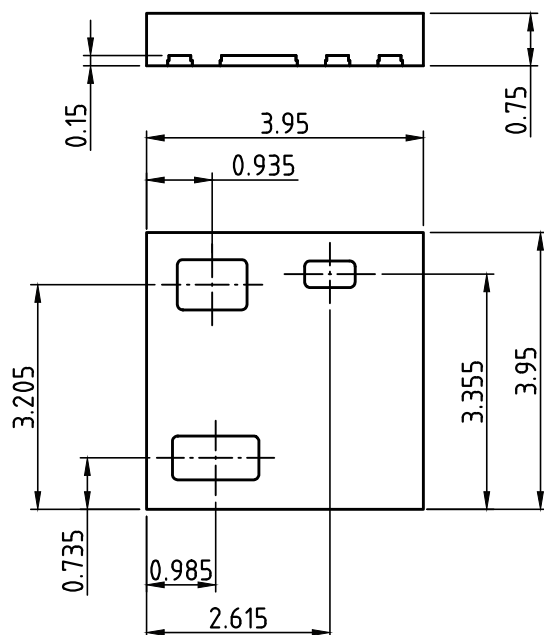


22318

Fig. 18 - Ambient Light Measurement with Averaging = 8;
using Continuous Conversion Mode

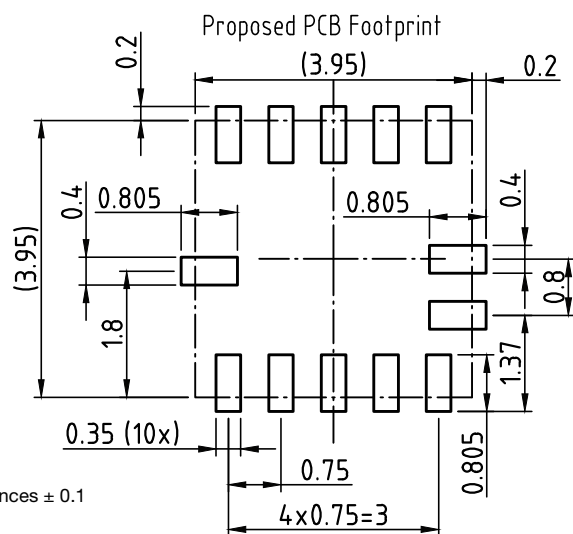
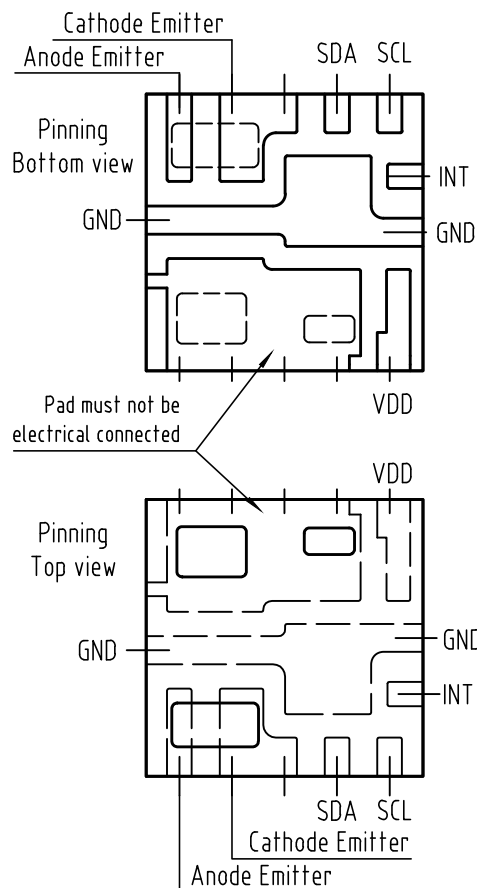
PACKAGE DIMENSIONS in millimeters


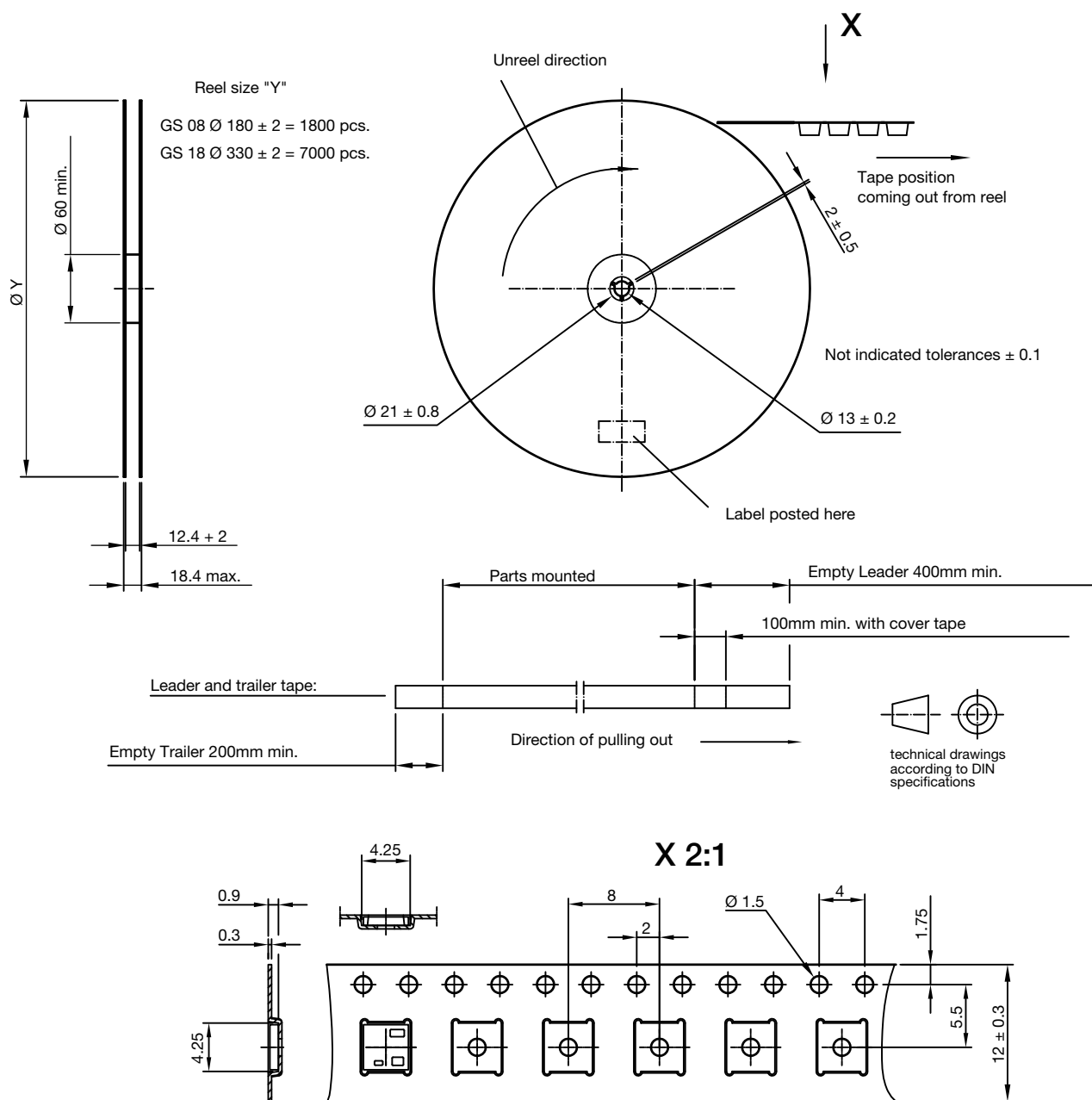
technical drawings
according to DIN
specifications



Drawing-No.: 6.550-5317.01-4

Not indicated tolerances ± 0.1



TAPE AND REEL DIMENSIONS in millimeters


Drawing-No.: 9.800-5103.01-4

SOLDER PROFILE

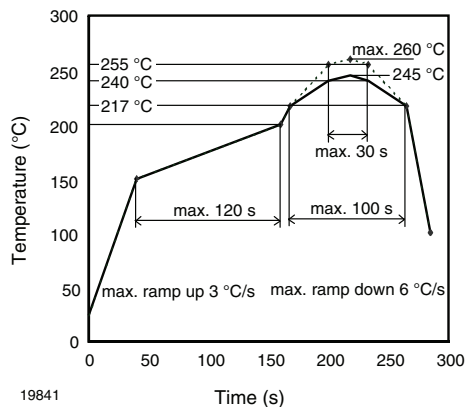


Fig. 19 - Lead (Pb)-free Reflow Solder Profile acc. J-STD-020

DRYPACK

Devices are packed in moisture barrier bags (MBB) to prevent the products from moisture absorption during transportation and storage. Each bag contains a desiccant.

FLOOR LIFE

Floor life (time between soldering and removing from MBB) must not exceed the time indicated on MBB label:

Floor life: 168 h

Conditions: $T_{amb} < 30\text{ °C}$, $RH < 60\%$

Moisture sensitivity level 3, according to J-STD-020

DRYING

In case of moisture absorption devices should be baked before soldering. Conditions see J-STD-020 or label. Devices taped on reel dry using recommended conditions 192 h at 40 °C (+ 5 °C), $RH < 5\%$.



Disclaimer

ALL PRODUCT, PRODUCT SPECIFICATIONS AND DATA ARE SUBJECT TO CHANGE WITHOUT NOTICE TO IMPROVE RELIABILITY, FUNCTION OR DESIGN OR OTHERWISE.

Vishay Intertechnology, Inc., its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, "Vishay"), disclaim any and all liability for any errors, inaccuracies or incompleteness contained in any datasheet or in any other disclosure relating to any product.

Vishay makes no warranty, representation or guarantee regarding the suitability of the products for any particular purpose or the continuing production of any product. To the maximum extent permitted by applicable law, Vishay disclaims (i) any and all liability arising out of the application or use of any product, (ii) any and all liability, including without limitation special, consequential or incidental damages, and (iii) any and all implied warranties, including warranties of fitness for particular purpose, non-infringement and merchantability.

Statements regarding the suitability of products for certain types of applications are based on Vishay's knowledge of typical requirements that are often placed on Vishay products in generic applications. Such statements are not binding statements about the suitability of products for a particular application. It is the customer's responsibility to validate that a particular product with the properties described in the product specification is suitable for use in a particular application. Parameters provided in datasheets and/or specifications may vary in different applications and performance may vary over time. All operating parameters, including typical parameters, must be validated for each customer application by the customer's technical experts. Product specifications do not expand or otherwise modify Vishay's terms and conditions of purchase, including but not limited to the warranty expressed therein.

Except as expressly indicated in writing, Vishay products are not designed for use in medical, life-saving, or life-sustaining applications or for any other application in which the failure of the Vishay product could result in personal injury or death. Customers using or selling Vishay products not expressly indicated for use in such applications do so at their own risk. Please contact authorized Vishay personnel to obtain written terms and conditions regarding products designed for such applications.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of Vishay. Product names and markings noted herein may be trademarks of their respective owners.

Material Category Policy

Vishay Intertechnology, Inc. hereby certifies that all its products that are identified as RoHS-Compliant fulfill the definitions and restrictions defined under Directive 2011/65/EU of The European Parliament and of the Council of June 8, 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment (EEE) - recast, unless otherwise specified as non-compliant.

Please note that some Vishay documentation may still make reference to RoHS Directive 2002/95/EC. We confirm that all the products identified as being compliant to Directive 2002/95/EC conform to Directive 2011/65/EU.

Vishay Intertechnology, Inc. hereby certifies that all its products that are identified as Halogen-Free follow Halogen-Free requirements as per JEDEC JS709A standards. Please note that some Vishay documentation may still make reference to the IEC 61249-2-21 definition. We confirm that all the products identified as being compliant to IEC 61249-2-21 conform to JEDEC JS709A standards.

±1.5g, ±6g Three Axis Low-g Micromachined Accelerometer

The MMA7361L is a low power, low profile capacitive micromachined accelerometer featuring signal conditioning, a 1-pole low pass filter, temperature compensation, self test, 0g-Detect which detects linear freefall, and g-Select which allows for the selection between 2 sensitivities. Zero-g offset and sensitivity are factory set and require no external devices. The MMA7361L includes a Sleep Mode that makes it ideal for handheld battery powered electronics.

Features

- 3mm x 5mm x 1.0mm LGA-14 Package
- Low Current Consumption: 400 μ A
- Sleep Mode: 3 μ A
- Low Voltage Operation: 2.2 V – 3.6 V
- High Sensitivity (800 mV/g @ 1.5g)
- Selectable Sensitivity (± 1.5 g, ± 6 g)
- Fast Turn On Time (0.5 ms Enable Response Time)
- Self Test for Freefall Detect Diagnosis
- 0g-Detect for Freefall Protection
- Signal Conditioning with Low Pass Filter
- Robust Design, High Shocks Survivability
- RoHS Compliant
- Environmentally Preferred Product
- Low Cost

Typical Applications

- 3D Gaming: Tilt and Motion Sensing, Event Recorder
- HDD MP3 Player: Freefall Detection
- Laptop PC: Freefall Detection, Anti-Theft
- Cell Phone: Image Stability, Text Scroll, Motion Dialing, E-Compass
- Pedometer: Motion Sensing
- PDA: Text Scroll
- Navigation and Dead Reckoning: E-Compass Tilt Compensation
- Robotics: Motion Sensing

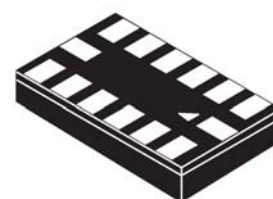
ORDERING INFORMATION

Part Number	Temperature Range	Package Drawing	Package	Shipping
MMA7361LT	-40 to +85°C	1977-01	LGA-14	Tray
MMA7361LR1	-40 to +85°C	1977-01	LGA-14	7" Tape & Reel
MMA7361LR2	-40 to +85°C	1977-01	LGA-14	13" Tape & Reel

MMA7361L

MMA7361L: XYZ AXIS ACCELEROMETER ± 1.5 g, ± 6 g

Bottom View



14 LEAD
LGA
CASE 1977-01

Top View

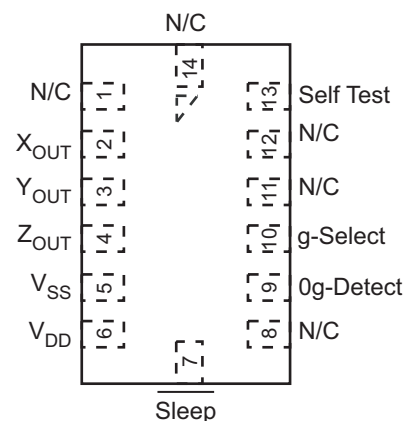


Figure 1. Pin Connections

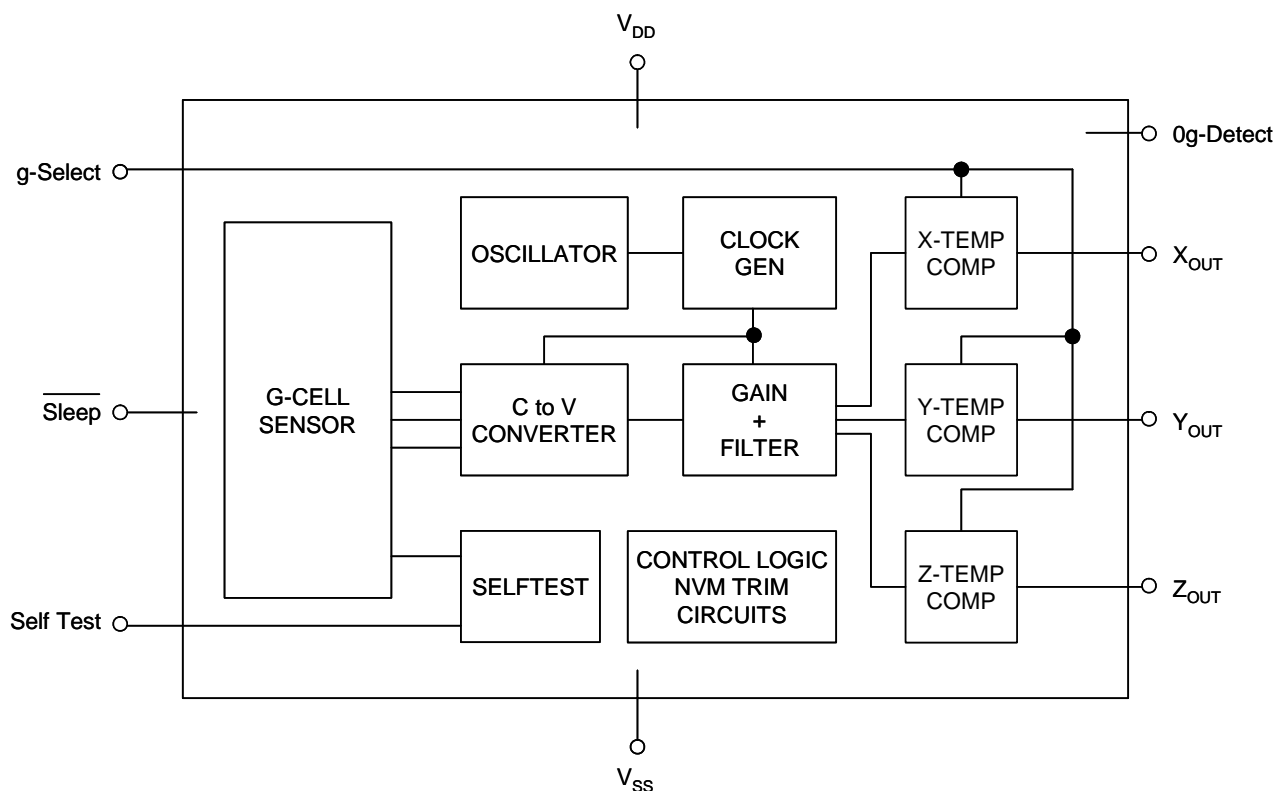


Figure 2. Simplified Accelerometer Functional Block Diagram

Table 1. Maximum Ratings

(Maximum ratings are the limits to which the device can be exposed without causing permanent damage.)

Rating	Symbol	Value	Unit
Maximum Acceleration (all axis)	g_{max}	± 5000	g
Supply Voltage	V_{DD}	-0.3 to +3.6	V
Drop Test ⁽¹⁾	D_{drop}	1.8	m
Storage Temperature Range	T_{stg}	-40 to +125	°C

1. Dropped onto concrete surface from any axis.

ELECTRO STATIC DISCHARGE (ESD)

WARNING: This device is sensitive to electrostatic discharge.

Although the Freescale accelerometer contains internal 2000 V ESD protection circuitry, extra precaution must be taken by the user to protect the chip from ESD. A charge of over 2000 volts can accumulate on the human body or associated test equipment. A charge of this magnitude can

alter the performance or cause failure of the chip. When handling the accelerometer, proper ESD precautions should be followed to avoid exposing the device to discharges which may be detrimental to its performance.

Table 2. Operating CharacteristicsUnless otherwise noted: $-40^{\circ}\text{C} \leq T_A \leq 85^{\circ}\text{C}$, $2.2\text{ V} \leq V_{DD} \leq 3.6\text{ V}$, Acceleration = $0g$, Loaded output⁽¹⁾

Characteristic	Symbol	Min	Typ	Max	Unit
Operating Range ⁽²⁾					
Supply Voltage ⁽³⁾	V_{DD}	2.2	3.3	3.6	V
Supply Current ⁽⁴⁾	I_{DD}	—	400	600	μA
Supply Current at Sleep Mode ⁽⁴⁾	I_{DD}	—	3	10	μA
Operating Temperature Range	T_A	-40	—	+85	$^{\circ}\text{C}$
Acceleration Range, X-Axis, Y-Axis, Z-Axis					
g-Select: 0	g_{FS}	—	± 1.5	—	g
g-Select: 1	g_{FS}	—	± 6.0	—	g
Output Signal					
Zero-g ($T_A = 25^{\circ}\text{C}$, $V_{DD} = 3.3\text{ V}$) ^{(5), (6)}	V_{OFF}	1.485	1.65	1.815	V
Zero-g ⁽⁴⁾	V_{OFF}, T_A	-2.0	± 0.5	+2.0	$\text{mg}/^{\circ}\text{C}$
Sensitivity ($T_A = 25^{\circ}\text{C}$, $V_{DD} = 3.3\text{ V}$)					
1.5g	$S_{1.5g}$	740	800	860	mV/g
6g	S_{6g}	190.6	206	221.5	mV/g
Sensitivity ⁽⁴⁾	S, T_A	-0.0075	± 0.002	+0.0075	$\%/^{\circ}\text{C}$
Bandwidth Response					
XY	f_{-3dBXY}	—	400	—	Hz
Z	f_{-3dBZ}	—	300	—	Hz
Output Impedance	Z_O	—	32	—	$\text{k}\Omega$
0g-Detect	$0g_{detect}$	-0.4	0	+0.4	g
Self Test					
Output Response					
X_{OUT}, Y_{OUT}	Δg_{STXY}	+0.05	-0.1	—	g
Z_{OUT}	Δg_{STZ}	+0.8	+1.0	+1.2	g
Input Low	V_{IL}	V_{SS}	—	$0.3 V_{DD}$	V
Input High	V_{IH}	$0.7 V_{DD}$	—	V_{DD}	V
Noise					
Power Spectral Density RMS (0.1 Hz – 1 kHz) ⁽⁴⁾	n_{PSD}	—	350	—	$\mu\text{g}/\sqrt{\text{Hz}}$
Control Timing					
Power-Up Response Time ⁽⁷⁾	$t_{RESPONSE}$	—	1.0	2.0	ms
Enable Response Time ⁽⁸⁾	t_{ENABLE}	—	0.5	2.0	ms
Self Test Response Time ⁽⁹⁾	t_{ST}	—	2.0	5.0	ms
Sensing Element Resonant Frequency					
XY	$f_{GCELLXY}$	—	6.0	—	kHz
Z	f_{GCELLZ}	—	3.4	—	kHz
Internal Sampling Frequency	f_{CLK}	—	11	—	kHz
Output Stage Performance					
Full-Scale Output Range ($I_{OUT} = 3\text{ }\mu\text{A}$)	V_{FSO}	$V_{SS}+0.1$	—	$V_{DD}-0.1$	V
Nonlinearity, $X_{OUT}, Y_{OUT}, Z_{OUT}$	NL_{OUT}	-1.0	—	+1.0	%FSO
Cross-Axis Sensitivity ⁽¹⁰⁾	$V_{XY, XZ, YZ}$	-5.0	—	+5.0	%

- For a loaded output, the measurements are observed after an RC filter consisting of an internal $32\text{ k}\Omega$ resistor and an external 3.3 nF capacitor (recommended as a minimum to filter clock noise) on the analog output for each axis and a $0.1\text{ }\mu\text{F}$ capacitor on $V_{DD} - \text{GND}$. The output sensor bandwidth is determined by the Capacitor added on the output. $f = 1/2\pi * (32 \times 10^3) * C$. $C = 3.3\text{ nF}$ corresponds to $BW = 1507\text{ HZ}$, which is the minimum to filter out internal clock noise.
- These limits define the range of operation for which the part will meet specification.
- Within the supply range of 2.2 and 3.6 V, the device operates as a fully calibrated linear accelerometer. Beyond these supply limits the device may operate as a linear device but is not guaranteed to be in calibration.
- This value is measured with g-Select in 1.5g mode.
- The device can measure both + and – acceleration. With no input acceleration the output is at midsupply. For positive acceleration the output will increase above $V_{DD}/2$. For negative acceleration, the output will decrease below $V_{DD}/2$.
- For optimal 0g offset performance, adhere to AN3484 and AN3447
- The response time between 10% of full scale V_{DD} input voltage and 90% of the final operating output voltage.
- The response time between 10% of full scale Sleep Mode input voltage and 90% of the final operating output voltage.
- The response time between 10% of the full scale self test input voltage and 90% of the self test output voltage.
- A measure of the device's ability to reject an acceleration applied 90° from the true axis of sensitivity.

PRINCIPLE OF OPERATION

The Freescale accelerometer is a surface-micromachined integrated-circuit accelerometer.

The device consists of a surface micromachined capacitive sensing cell (g-cell) and a signal conditioning ASIC contained in a single package. The sensing element is sealed hermetically at the wafer level using a bulk micromachined cap wafer.

The g-cell is a mechanical structure formed from semiconductor materials (polysilicon) using semiconductor processes (masking and etching). It can be modeled as a set of beams attached to a movable central mass that move between fixed beams. The movable beams can be deflected from their rest position by subjecting the system to an acceleration (Figure 3).

As the beams attached to the central mass move, the distance from them to the fixed beams on one side will increase by the same amount that the distance to the fixed beams on the other side decreases. The change in distance is a measure of acceleration.

The g-cell beams form two back-to-back capacitors (Figure 3). As the center beam moves with acceleration, the distance between the beams changes and each capacitor's value will change, ($C = A\epsilon/D$). Where A is the area of the beam, ϵ is the dielectric constant, and D is the distance between the beams.

The ASIC uses switched capacitor techniques to measure the g-cell capacitors and extract the acceleration data from the difference between the two capacitors. The ASIC also signal conditions and filters (switched capacitor) the signal, providing a high level output voltage that is ratiometric and proportional to acceleration.

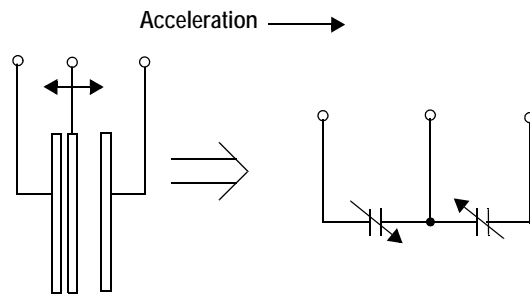


Figure 3. Simplified Transducer Physical Model

SPECIAL FEATURES

0g-Detect

The sensor offers a 0g-Detect feature that provides a logic high signal when all three axes are at 0g. This feature enables the application of Linear Freefall protection if the signal is connected to an interrupt pin or a poled I/O pin on a microcontroller.

Self Test

The sensor provides a self test feature that allows the verification of the mechanical and electrical integrity of the accelerometer at any time before or after installation. This feature is critical in applications such as hard disk drive

protection where system integrity must be ensured over the life of the product. Customers can use self test to verify the solderability to confirm that the part was mounted to the PCB correctly. To use this feature to verify the 0g-Detect function, the accelerometer should be held upside down so that the z-axis experiences -1g. When the self test function is initiated, an electrostatic force is applied to each axis to cause it to deflect. The x- and y-axis are deflected slightly while the z-axis is trimmed to deflect 1g. This procedure assures that both the mechanical (g-cell) and electronic sections of the accelerometer are functioning.

g-Select

The g-Select feature allows for the selection between two sensitivities. Depending on the logic input placed on pin 10, the device internal gain will be changed allowing it to function with a 1.5g or 6g sensitivity (Table 3). This feature is ideal when a product has applications requiring two different sensitivities for optimum performance. The sensitivity can be changed at anytime during the operation of the product. The g-Select pin can be left unconnected for applications requiring only a 1.5g sensitivity as the device has an internal pull-down to keep it at that sensitivity (800mV/g).

Table 3. g-Select Pin Description

g-Select	g-Range	Sensitivity
0	1.5g	800 mV/g
1	6g	206 mV/g

Sleep Mode

The 3 axis accelerometer provides a Sleep Mode that is ideal for battery operated products. When Sleep Mode is active, the device outputs are turned off, providing significant reduction of operating current. A low input signal on pin 7 (Sleep Mode) will place the device in this mode and reduce the current to 3 μ A typ. For lower power consumption, it is recommended to set g-Select to 1.5g mode. By placing a high input signal on pin 7, the device will resume to normal mode of operation.

Filtering

The 3 axis accelerometer contains an onboard single-pole switched capacitor filter. Because the filter is realized using switched capacitor techniques, there is no requirement for external passive components (resistors and capacitors) to set the cut-off frequency.

Ratiometricity

Ratiometricity simply means the output offset voltage and sensitivity will scale linearly with applied supply voltage. That is, as supply voltage is increased, the sensitivity and offset increase linearly; as supply voltage decreases, offset and sensitivity decrease linearly. This is a key feature when interfacing to a microcontroller or an A/D converter because it provides system level cancellation of supply induced errors in the analog to digital conversion process.

BASIC CONNECTIONS

Pin Descriptions

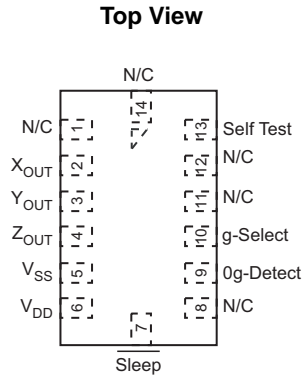


Figure 4. Pinout Description

Table 4. Pin Descriptions

Pin No.	Pin Name	Description
1	N/C	No internal connection Leave unconnected
2	X _{OUT}	X direction output voltage
3	Y _{OUT}	Y direction output voltage
4	Z _{OUT}	Z direction output voltage
5	V _{SS}	Power Supply Ground
6	V _{DD}	Power Supply Input
7	Sleep	Logic input pin to enable product or Sleep Mode
8	NC	No internal connection Leave unconnected
9	0g-Detect	Linear Freefall digital logic output signal
10	g-Select	Logic input pin to select g level
11	N/C	Unused for factory trim Leave unconnected
12	N/C	Unused for factory trim Leave unconnected
13	Self Test	Input pin to initiate Self Test
14	N/C	Unused for factory trim Leave unconnected

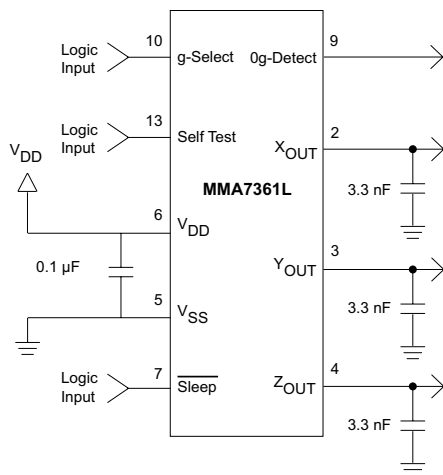


Figure 5. Accelerometer with Recommended Connection Diagram

PCB Layout

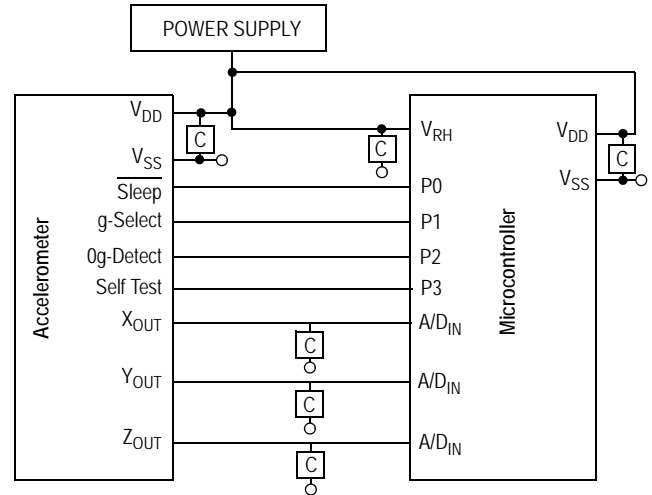
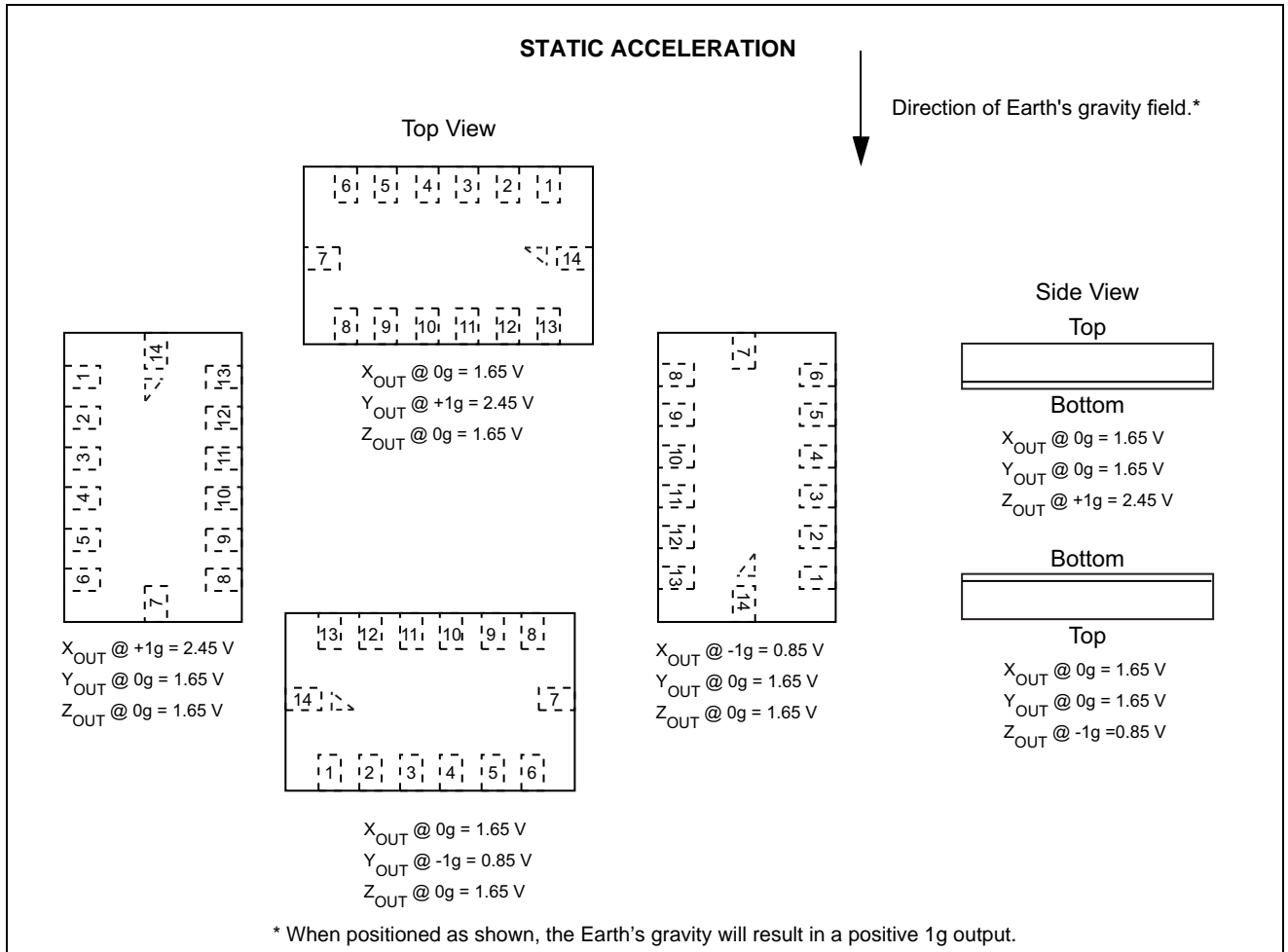
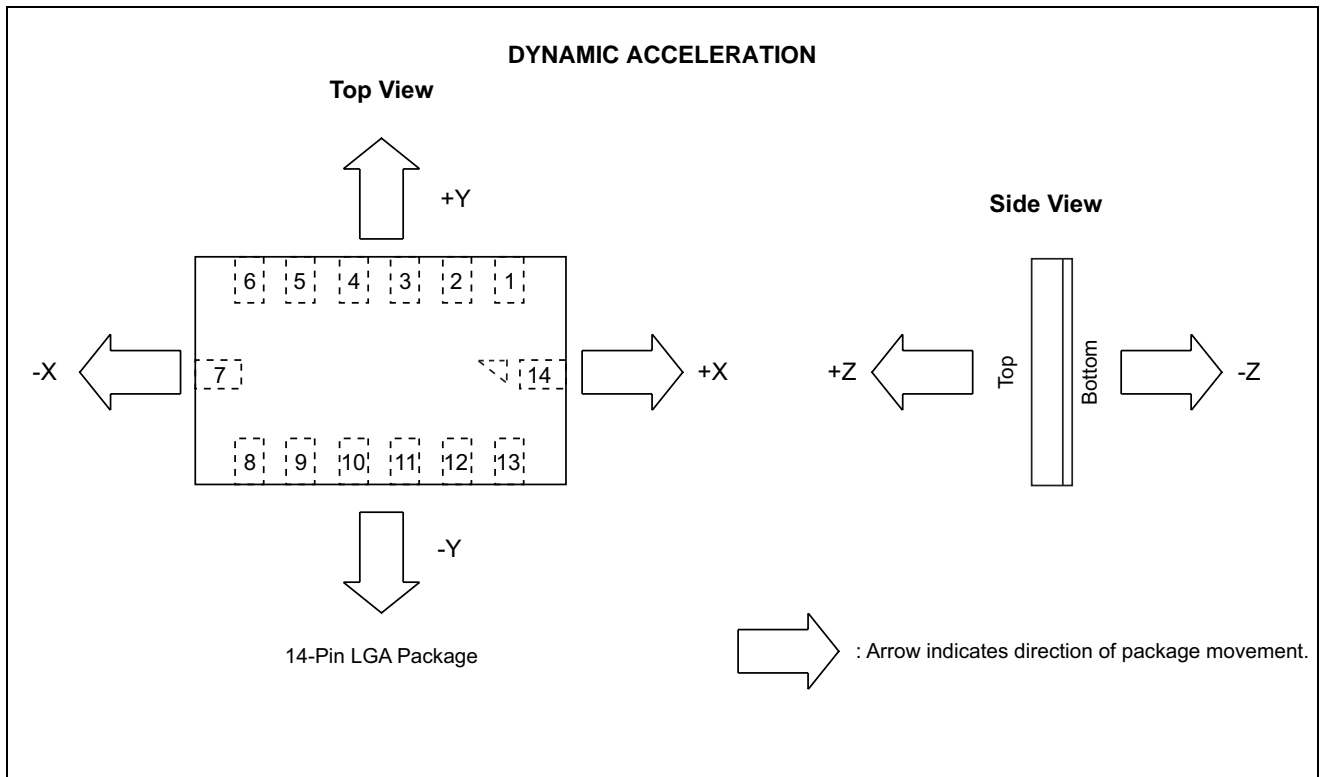


Figure 6. Recommended PCB Layout for Interfacing Accelerometer to Microcontroller

NOTES:

1. Use 0.1 μF capacitor on V_{DD} to decouple the power source.
2. Physical coupling distance of the accelerometer to the microcontroller should be minimal.
3. Place a ground plane beneath the accelerometer to reduce noise, the ground plane should be attached to all of the open ended terminals shown in [Figure 6](#).
4. Use a 3.3nF capacitor on the outputs of the accelerometer to minimize clock noise (from the switched capacitor filter circuit).
5. PCB layout of power and ground should not couple power supply noise.
6. Accelerometer and microcontroller should not be a high current path.
7. A/D sampling rate and any external power supply switching frequency should be selected such that they do not interfere with the internal accelerometer sampling frequency (11 kHz for the sampling frequency). This will prevent aliasing errors.
8. 10M Ω or higher is recommended on X_{OUT}, Y_{OUT} and Z_{OUT} to prevent loss due to the voltage divider relationship between the internal 32 k Ω resistor and the measurement input impedance.



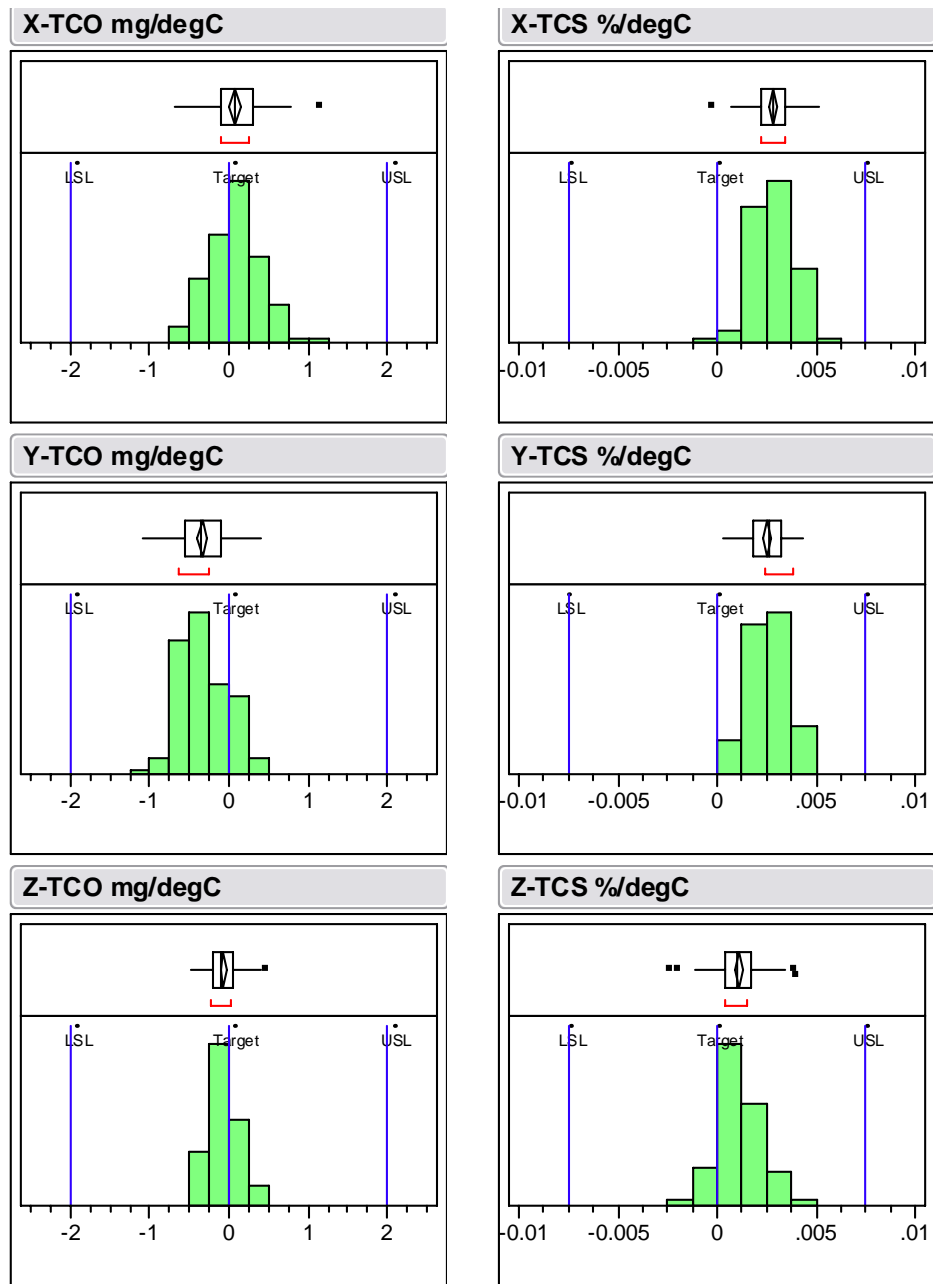


Figure 7. MMA7361L Temperature Coefficient of Offset (TCO) and Temperature Coefficient of Sensitivity (TCS) Distribution Charts

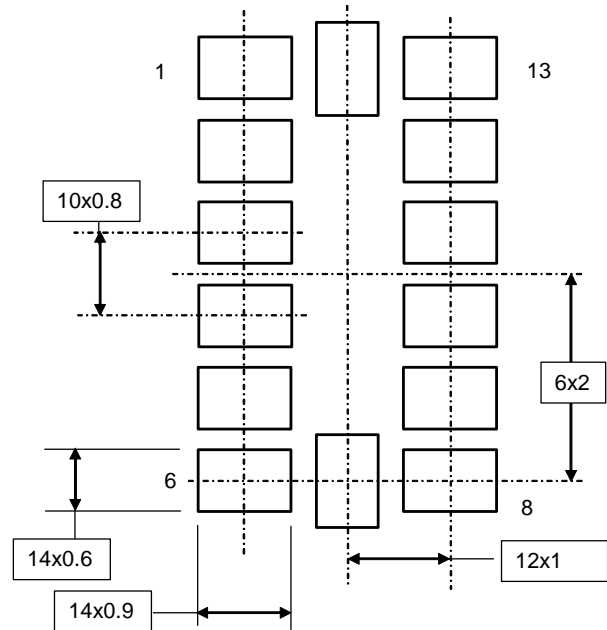
MINIMUM RECOMMENDED FOOTPRINT FOR SURFACE MOUNTED APPLICATIONS

PCB Mounting Recommendations

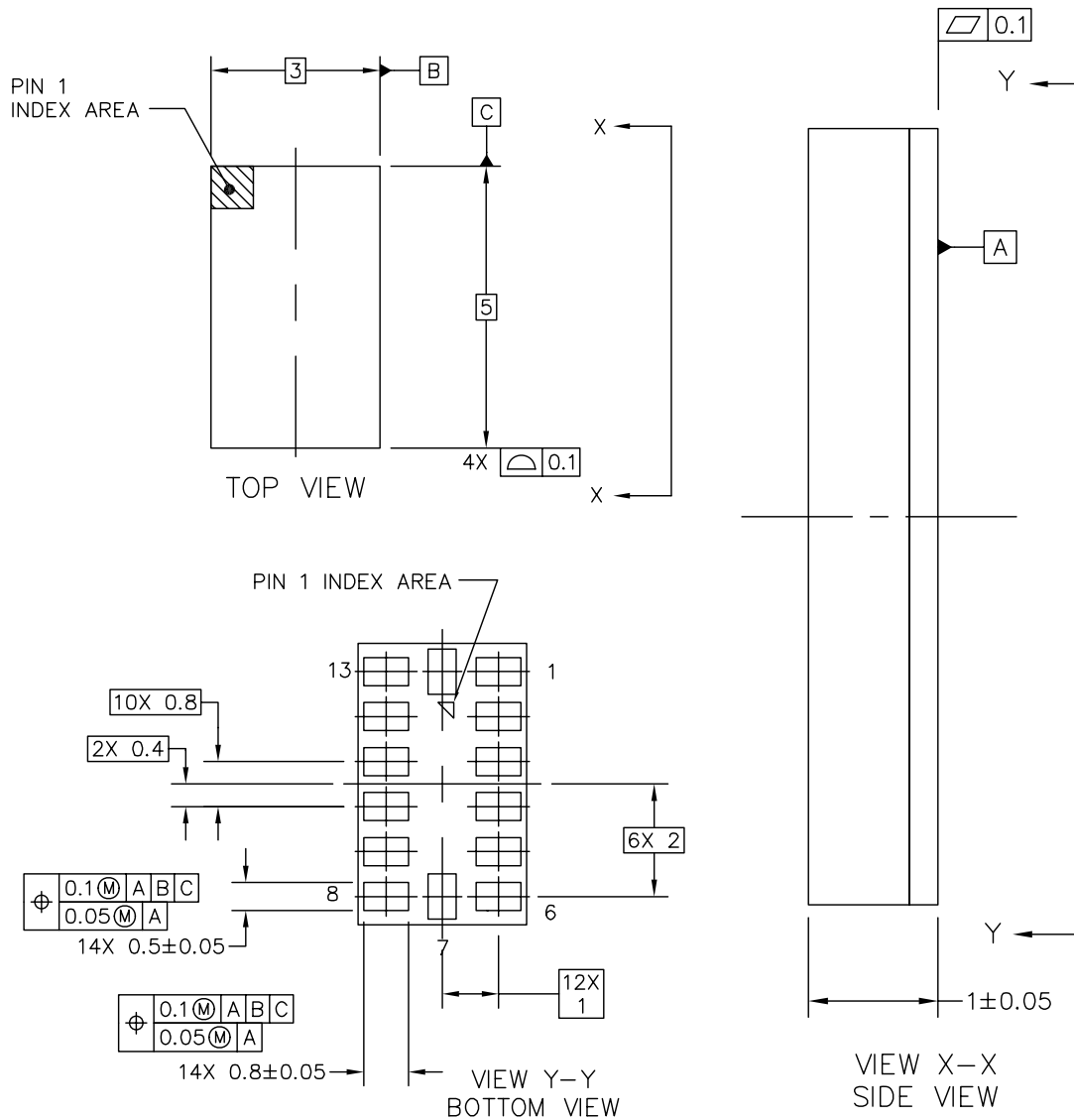
MEMS based sensors are sensitive to Printed Circuit Board (PCB) reflow processes. For optimal zero-g offset after PCB mounting, care must be taken to PCB layout and reflow conditions. Reference application note AN3484 for best practices to minimize the zero-g offset shift after PCB mounting.

Surface mount board layout is a critical portion of the total design. The footprint for the surface mount packages must be the correct size to ensure proper solder connection interface between the board and the package.

With the correct footprint, the packages will self-align when subjected to a solder reflow process. It is always recommended to design boards with a solder mask layer to avoid bridging and shorting between solder pads.



PACKAGE DIMENSIONS



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.		MECHANICAL OUTLINE		PRINT VERSION NOT TO SCALE	
TITLE: LGA 14 I/O, 3 X 5 X 1 PITCH 0.8, SENSOR 1.0MM PKG		DOCUMENT NO: 98ASA10801D		REV: A	
		CASE NUMBER: 1977-01		09 JAN 2008	
		STANDARD: NON-JEDEC			

**CASE 1977-01
ISSUE A
14-LEAD LGA**

MMA7361L

PACKAGE DIMENSIONS

NOTES:

1. ALL DIMENSIONS IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE		PRINT VERSION NOT TO SCALE	
TITLE: LGA 14 I/O, 3 X 5 X 1 PITCH 0.8, SENSOR 1.0MM PKG	DOCUMENT NO: 98ASA10801D		REV: A	
	CASE NUMBER: 1977-01		09 JAN 2008	
	STANDARD: NON-JEDEC			

**CASE 1977-01
ISSUE A
14-LEAD LGA**

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2008. All rights reserved.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.



Digital relative humidity & temperature sensor RHT03

1. Feature & Application:

- *High precision
- *Capacitive type
- *Full range temperature compensated
- *Relative humidity and temperature measurement
- *Calibrated digital signal
- *Outstanding long-term stability
- *Extra components not needed
- *Long transmission distance, up to 100 meters
- *Low power consumption
- *4 pins packaged and fully interchangeable

2. Description:

RHT03 output calibrated digital signal. It applies exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

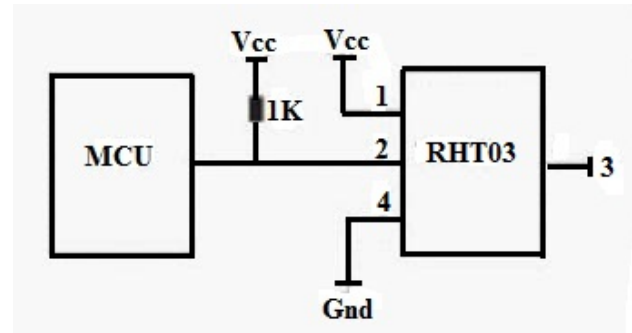
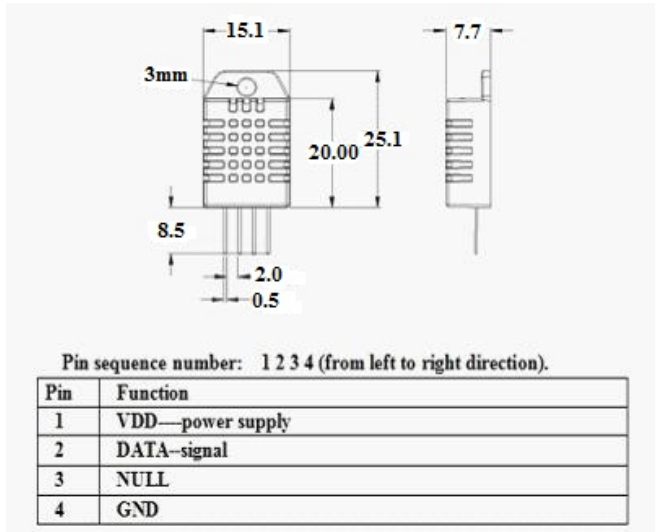
Small size & low consumption & long transmission distance(100m) enable RHT03 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	RHT03	
Power supply	3.3-6V DC	
Output signal	digital signal via MaxDetect 1-wire bus	
Sensing element	Polymer humidity capacitor	
Operating range	humidity 0-100%RH;	temperature -40~80Celsius
Accuracy	humidity +2%RH (Max +-5%RH);	temperature +-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH;	temperature 0.1Celsius
Repeatability	humidity +-1%RH;	temperature +-0.2Celsius

Humidity hysteresis	+/-0.3%RH
Long-term Stability	+/-0.5%RH/year
Interchangeability	fully interchangeable

4. Dimensions: (unit----mm)



5. Electrical connection diagram:

6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3.3-6V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for wave filtering.

(2) Communication and signal

MaxDetect 1-wire bus is used for communication between MCU and RHT03. (MaxDetect 1-wire bus is specially designed by MaxDetect Technology Co., Ltd. , it's different from Maxim/Dallas 1-wire bus, so it's incompatible with Dallas 1-wire bus.)

Illustration of MaxDetect 1-wire bus:

Data is comprised of integral and decimal part, the following is the formula for data.

DATA=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data+8 bit check-sum

If the data transmission is right, check-sum should be:

Check sum=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data

Example: MCU has received 40 bits data from RHT03 as

0000 0010 1000 1100 0000 0001 0101 1111 1110 1110

16 bits RH data 16 bits T data check sum

Check sum=0000 0010+1000 1100+0000 0001+0101 1111=1110 1110

RH= (0000 0010 1000 1100)/10=65.2%RH

T=(0000 0001 0101 1111)/10=35.1□

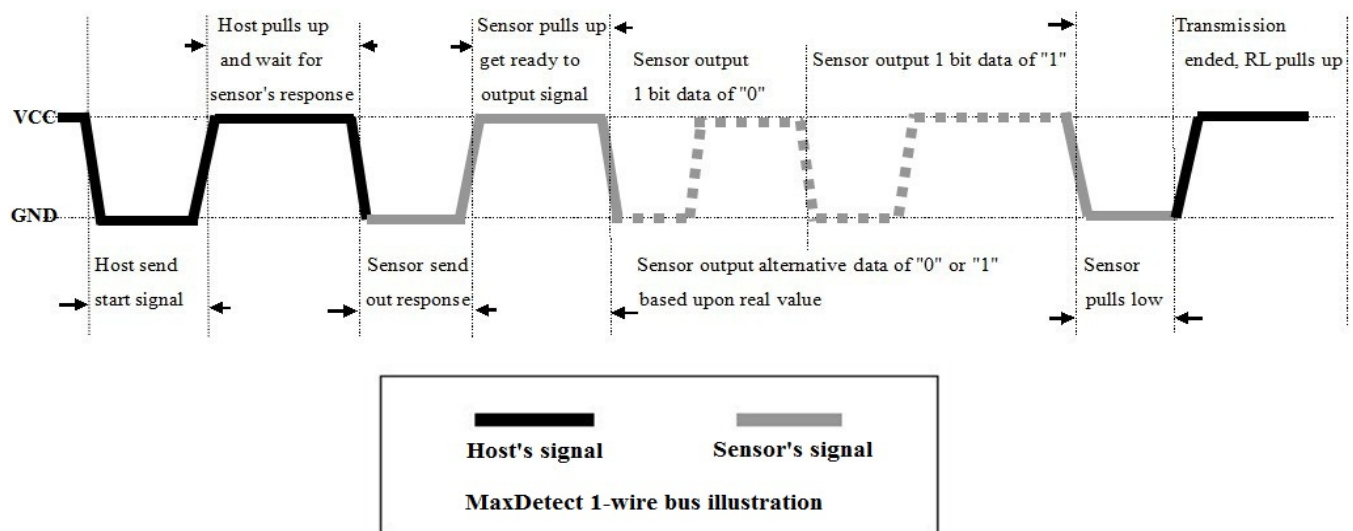
When highest bit of temperature is 1, it means the temperature is below 0 degree Celsius.

Example: 1000 0000 0110 0101, T= minus 10.1□

16 bits T data

When MCU send start signal, RHT03 change from standby-status to running-status. When MCU finishes sending the start signal, RHT03 will send response signal of 40-bit data that reflect the relative humidity and temperature to MCU. Without start signal from MCU, RHT03 will not give response signal to MCU. One start signal for one response data from RHT03 that reflect the relative humidity and temperature. RHT03 will change to standby status when data collecting finished if it don't receive start signal from MCU again.

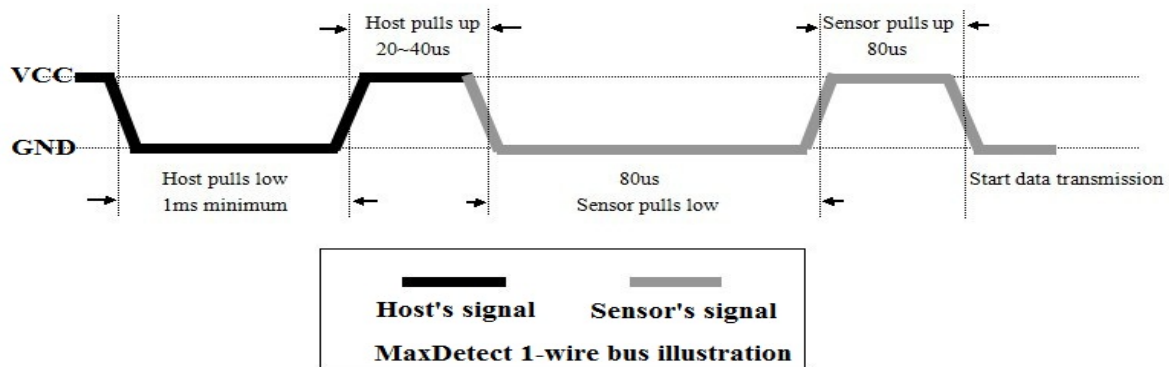
See below figure for overall communication process, the interval of whole process must beyond 2 seconds.



1) Step 1: MCU send out start signal to RHT03 and RHT03 send response signal to MCU

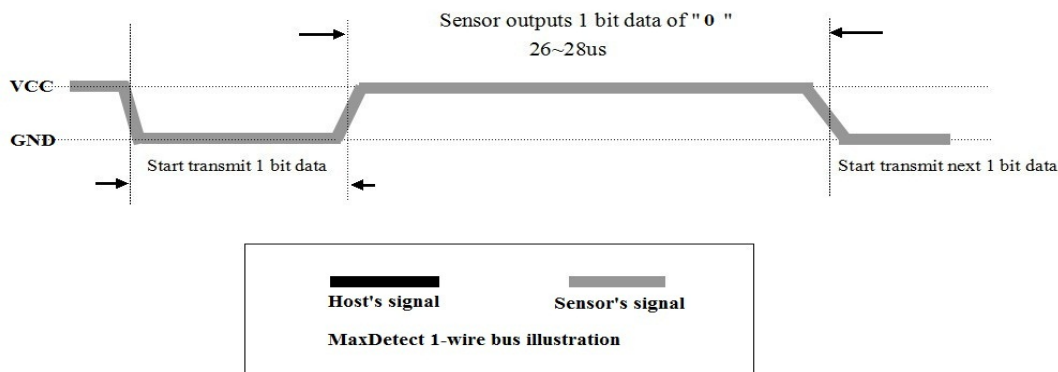
Data-bus's free status is high voltage level. When communication between MCU and RHT03 begins, MCU will pull low data-bus and this process must beyond at least 1~10ms to ensure RHT03 could detect MCU's signal, then MCU will pull up and wait 20~40us for RHT03's response.

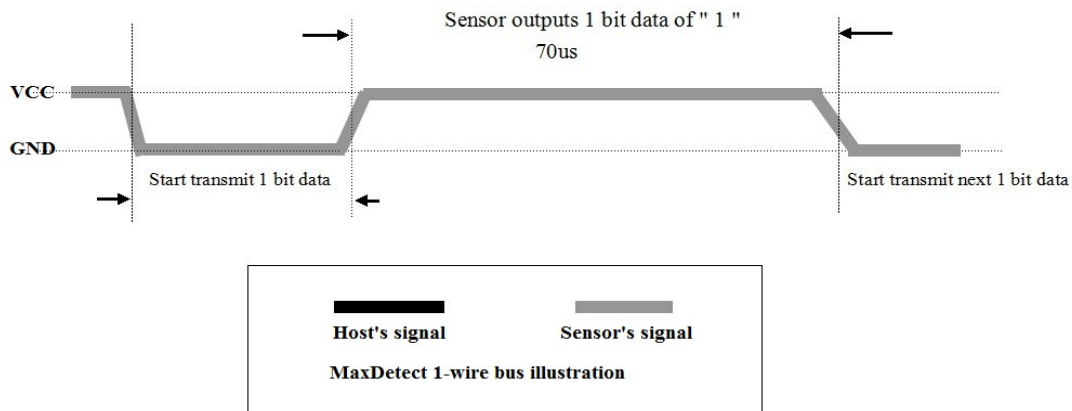
When RHT03 detect the start signal, RHT03 will pull low the bus 80us as response signal, then RHT03 pulls up 80us for preparation to send data. See below figure:



2). Step 2: RHT03 send data to MCU

When RHT03 is sending data to MCU, every bit's transmission begin with low-voltage-level that last 50us, the following high-voltage-level signal's length decide the bit is "1" or "0". See below figures:





Attention:

If signal from RHT03 is always high-voltage-level, it means RHT03 is not working properly, please check the electrical connection status.

7. Electrical Characteristics:

Items	Condition	Min	Typical	Max	Unit
Power supply	DC	3.3	5	6	V
Current supply	Measuring	1		1.5	mA
	Stand-by	40	Null	50	uA
Collecting period	Second		2		Second

8. Attentions of application:

(1) Operating and storage conditions

We don't recommend the applying RH-range beyond the range stated in this specification. The RHT03 sensor can recover after working in abnormal operating condition to calibrated status, but will accelerate sensors' aging.

(2) Attentions to chemical materials

Vapor from chemical materials may interfere RHT03's sensitive-elements and debase RHT03's sensitivity.

(3) Disposal when (1) & (2) happens

Step one: Keep the RHT03 sensor at condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;

Step two: After step one, keep the RHT03 sensor at condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Attention to temperature's affection

Relative humidity strongly depend on temperature, that is why we use temperature compensation technology to ensure accurate measurement of RH. But it's still be much better to keep the sensor at same temperature when sensing.

RHT03 should be mounted at the place as far as possible from parts that may cause change to temperature.

(5) Attentions to light

Long time exposure to strong light and ultraviolet may debase RHT03's performance.

(6) Attentions to connection wires

The connection wires' quality will effect communication's quality and distance, high quality shielding-wire is recommended.

(7) Other attentions

- * Welding temperature should be bellow 260Celsius.

- * Avoid using the sensor under dew condition.

- * Don't use this product in safety or emergency stop devices or any other occasion that failure of RHT03 may cause personal injury.

HC-05

-Bluetooth to Serial Port Module

Overview



HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications

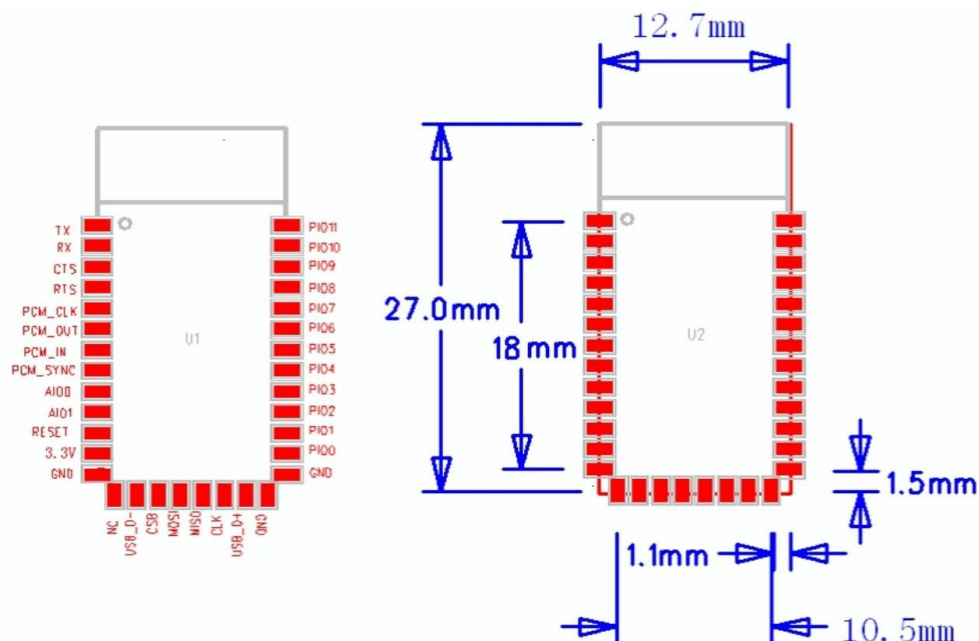
Hardware features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Hardware



PIN Name	PIN #	Pad type	Description	Note
GND	13 21 22	VSS	Ground pot	
3.3 VCC	12	3.3V	Integrated 3.3V (+) supply with On-chip linear regulator output within 3.15-3.3V	
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	
PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA(if fitted)	
PIO2	25	Bi-Directional	Programmable input/output line	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	
PIO7	30	Bi-Directional	Programmable input/output line	
PIO8	31	Bi-Directional	Programmable input/output line	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	

RESETB	11	CMOS input with weak internal pull-up	Reset if low.input debounced so must be low for >5MS to cause a reset	
UART_RTS	4	CMOS output, tri-stable with weak internal pull-up	UART request to send, active low	
UART_CTS	3	CMOS input with weak internal pull-down	UART clear to send, active low	
UART_RX	2	CMOS input with weak internal pull-down	UART Data input	
UART_TX	1	CMOS output, Tri-stable with weak internal pull-up	UART Data output	
SPI_MOSI	17	CMOS input with weak internal pull-down	Serial peripheral interface data input	

SPI_CSB	16	CMOS input with weak internal pull-up	Chip select for serial peripheral interface, active low	
SPI_CLK	19	CMOS input with weak internal pull-down	Serial peripheral interface clock	
SPI_MISO	18	CMOS input with weak internal pull-down	Serial peripheral interface data Output	
USB_-	15	Bi-Directional		

USB_+	20	Bi-Directional		
NC	14			
PCM_CLK	5	Bi-Directional	Synchronous PCM data clock	
PCM_OUT	6	CMOS output	Synchronous PCM data output	
PCM_IN	7	CMOS Input	Synchronous PCM data input	
PCM_SYNC	8	Bi-Directional	Synchronous PCM data strobe	

AT command Default:

How to set the mode to server (master):

1. Connect PIO11 to high level.
2. Power on, module into command state.
3. Using baud rate 38400, sent the "AT+ROLE=1\r\n" to module, with "OK\r\n" means setting successes.
4. Connect the PIO11 to low level, repower the module, the module work as server (master).

AT commands: (all end with \r\n)

1. Test command:

Command	Respond	Parameter
AT	OK	-

2. Reset

Command	Respond	Parameter
AT+RESET	OK	-

3. Get firmware version

Command	Respond	Parameter
AT+VERSION?	+VERSION:<Param> OK	Param : firmware version

Example:

AT+VERSION?\r\n

+VERSION:2.0-20100601

OK

4. Restore default

Command	Respond	Parameter
AT+ORGL	OK	-

Default state:

Slave mode, pin code :1234, device name: H-C-2010-06-01 ,Baud 38400bits/s.

5. Get module address

Command	Respond	Parameter
AT+ADDR?	+ADDR:<Param> OK	Param: address of Bluetooth module

Bluetooth address: NAP: UAP : LAP

Example:

AT+ADDR?\r\n

+ADDR:1234:56:abcdef

OK

6. Set/Check module name:

Command	Respond	Parameter
AT+NAME=<Param>	OK	Param: Bluetooth module name (Default :HC-05)
AT+NAME?	+NAME:<Param> OK (/FAIL)	

Example:

AT+NAME=HC-05\r\n set the module name to "HC-05"

OK

AT+NAME=ITeadStudio\r\n

OK

AT+NAME?\r\n

+NAME: ITeadStudio

OK

7. Get the Bluetooth device name:

Command	Respond	Parameter
AT+RNAME?<Param1>	1. +NAME:<Param2> OK 2. FAIL	Param1,Param 2 : the address of Bluetooth device

Example: (Device address 00:02:72:0d:22:24, name: ITead)

AT+RNAME? 0002, 72, 0d2224\r\n

+RNAME:ITead

OK

8. Set/Check module mode:

Command	Respond	Parameter
AT+ROLE=<Param>	OK	Param: 0- Slave
AT+ROLE?	+ROLE:<Param>	

	OK	1-Master 2-Slave-Loop
--	----	--------------------------

9. Set/Check device class

Command	Respond	Parameter
AT+CLASS=<Param>	OK	Param: Device Class
AT+ CLASS?	1. +CLASS:<Param> OK 2. FAIL	

10. Set/Check GIAC (General Inquire Access Code)

Command	Respond	Parameter
AT+IAC=<Param>	1.OK 2. FAIL	Param: GIAC (Default : 9e8b33)
AT+IAC	+IAC:<Param> OK	

Example:

AT+IAC=9e8b3f\r\n

OK

AT+IAC?\r\n

+IAC: 9e8b3f

OK

11. Set/Check -- Query access patterns

Command	Respond	Parameter
AT+INQM=<Param>,<Param2>,<Param3>	1.OK 2. FAIL	Param: 0——inquiry_mode_standard 1——inquiry_mode_rssi Param2: Maximum number of Bluetooth devices to respond to Param3: Timeout (1-48 : 1.28s to 61.44s)
AT+ INQM?	+INQM : <Param>,<Param2>,<Param3> OK	

Example:

AT+INQM=1,9,48\r\n

OK

AT+INQM\r\n

+INQM:1, 9, 48

OK

12. Set/Check PIN code:

Command	Respond	Parameter
AT+PSWD=<Param>	OK	Param: PIN code (Default 1234)
AT+ PSWD?	+ PSWD : <Param> OK	

13. Set/Check serial parameter:

Command	Respond	Parameter
AT+UART=<Param>,<Param2>,<Param3>	OK	Param1: Baud Param2: Stop bit Param3: Parity
AT+ UART?	+UART=<Param>,<Param2>,<Param3> OK	

Example:

AT+UART=115200, 1,2,\r\n

OK

AT+UART?

+UART:115200,1,2

OK

14. Set/Check connect mode:

Command	Respond	Parameter
AT+CMODE=<Param>	OK	Param: 0 - connect fixed address 1 - connect any address 2 - slave-Loop
AT+ CMODE?	+ CMODE:<Param> OK	

15. Set/Check fixed address:

Command	Respond	Parameter
AT+BIND=<Param>	OK	Param: Fixed address (Default 00:00:00:00:00:00)
AT+ BIND?	+ BIND:<Param> OK	

Example:

AT+BIND=1234, 56, abcdef\r\n

OK

AT+BIND?\r\n

+BIND:1234:56:abcdef

OK

16. Set/Check LED I/O

Command	Respond	Parameter
AT+POLAR=<Param1,<Param2>	OK	Param1: 0- PIO8 low drive LED 1- PIO8 high drive LED
AT+ POLAR?	+ POLAR=<Param1>,<Param2> OK	

		Param2: 0- PIO9 low drive LED 1- PIO9 high drive LED
--	--	--

17. Set PIO output

Command	Respond	Parameter
AT+PIO=<Param1>,<Param2>	OK	Param1: PIO number Param2: PIO level 0- low 1- high

Example:

1. PIO10 output high level

AT+PIO=10, 1\r\n

OK

18. Set/Check – scan parameter

Command	Respond	Parameter
AT+IPSCAN=<Param1>,<Param2>,<Param3>,<Param4>	OK	Param1: Query time interval
AT+IPSCAN?	+IPSCAN:<Param1>,<Param2>,<Param3>,<Param4> OK	Param2: Query duration Param3: Paging interval Param4: Call duration

Example:

AT+IPSCAN =1234,500,1200,250\r\n

OK

AT+IPSCAN?

+IPSCAN:1234,500,1200,250

19. Set/Check – SHIFF parameter

Command	Respond	Parameter
AT+SNIFF=<Param1>,<Param2>,<Param3>,<Param4>	OK	Param1: Max time Param2: Min time
AT+ SNIFF?	+SNIFF:<Param1>,<Param2>,<Param3>,<Param4> OK	Param3: Retry time Param4: Time out

20. Set/Check security mode

Command	Respond	Parameter
AT+SENM=<Param1>,<Param2>	1. OK 2. FAIL	Param1:
AT+ SENM?	+ SENM:<Param1>,<Param2>	0—sec_mode0+off 1—sec_mode1+non_se

	OK	cure 2—sec_mode2_service 3—sec_mode3_link 4—sec_mode_unknow n Param2: 0—hci_enc_mode_off 1—hci_enc_mode_pt_t o_pt 2—hci_enc_mode_pt_t o_pt_and_bcast
--	----	--

21. Delete Authenticated Device

Command	Respond	Parameter
AT+PMSAD=<Param>	OK	Param: Authenticated Device Address

Example:

AT+PMSAD =1234,56,abcdef\r\n

OK

22. Delete All Authenticated Device

Command	Respond	Parameter
AT+ RMAAD	OK	-

23. Search Authenticated Device

Command	Respond	Parameter
AT+FSAD=<Param>	1. OK 2. FAIL	Param: Device address

24. Get Authenticated Device Count

Command	Respond	Parameter
AT+ADCN?	+ADCN: <Param> OK	Param: Device Count

25. Most Recently Used Authenticated Device

Command	Respond	Parameter
AT+MRAD?	+ MRAD: <Param> OK	Param: Recently Authenticated Device Address

26. Get the module working state

Command	Respond	Parameter
---------	---------	-----------

AT+ STATE?	+ STATE: <Param> OK	Param: "INITIALIZED" "READY" "PAIRABLE" "PAIRED" "INQUIRING" "CONNECTING" "CONNECTED" "DISCONNECTED" "NUKNOW"
------------	------------------------	--

27. Initialize the SPP profile lib

Command	Respond	Parameter
AT+INIT	1. OK 2. FAIL	-

28. Inquiry Bluetooth Device

Command	Respond	Parameter
AT+INQ	+INQ: <Param1> , <Param2> , <Param3> OK	Param1: Address Param2: Device Class Param3 : RSSI Signal strength

Example:

```

AT+INIT\r\n
OK
AT+IAC=9e8b33\r\n
OK
AT+CLASS=0\r\n
AT+INQM=1,9,48\r\n
At+INQ\r\n
+INQ:2:72:D2224,3E0104,FFBC
+INQ:1234:56:0,1F1F,FFC1
+INQ:1234:56:0,1F1F,FFC0
+INQ:1234:56:0,1F1F,FFC1
+INQ:2:72:D2224,3F0104,FFAD
+INQ:1234:56:0,1F1F,FFBE
+INQ:1234:56:0,1F1F,FFC2
+INQ:1234:56:0,1F1F,FFBE
+INQ:2:72:D2224,3F0104,FFBC
OK
  
```

28. Cancel Inquiring Bluetooth Device

Command	Respond	Parameter
AT+ INQC	OK	-

29. Equipment Matching

Command	Respond	Parameter
AT+PAIR=<Param1>,<Param2>	1. OK 2. FAIL	Param1: Device Address Param2: Time out

30. Connect Device

Command	Respond	Parameter
AT+LINK=<Param>	1. OK 2. FAIL	Param: Device Address

Example:

AT+FSAD=1234,56,abcdef\r\n

OK

AT+LINK=1234,56,abcdef\r\n

OK

31. Disconnect

Command	Respond	Parameter
AT+DISC	1. +DISC:SUCCESS OK 2. +DISC:LINK_LOSS OK 3. +DISC:NO_SLC OK 4. +DISC:TIMEOUT OK 5. +DISC:ERROR OK	Param: Device Address

32. Energy-saving mode

Command	Respond	Parameter
AT+ENSNIFF=<Param>	OK	Param: Device Address

33. Exerts Energy-saving mode

Command	Respond	Parameter
AT+ EXSNIFF =<Param>	OK	Param: Device Address

Revision History

Rev.	Description	Release date
v1.0	Initial version	7/18/2010

RN-171-XV 802.11 b/g Wireless LAN Module

Features

- Drop-in Wi-Fi solution for existing systems that currently use 802.15.4 modules
- Based on Roving Networks' robust RN-171 Wi-Fi module
- Based on a pseudo-standard footprint
- On-board TCP/IP stack provides Internet access to every node
- No custom profiles needed
- Ultra-low power: 4-uA sleep, 40-mA Rx, 180-mA Tx at 10 dBm
- Configurable transmit power: 0 to +12 dBm
- Hardware interface: TTL UART
- Data rate: 464 Kbps using hardware flow control
- Through-hole board simplifies system integration
- 8 general purpose digital I/O pins
- 3 analog sensor interfaces
- Real-time clock for wakeup and time stamping
- Complete TCP/IP networking stack
- Wi-Fi Alliance certified for WEP, WPA, and WPA2-PSK
- WPS mode for easy configuration
- FCC/CE/ICS certified and RoHS compliant

Applications

- Industrial metering
- HVAC control
- Room temperature sensors
- Pump configuration and control
- Telemetry
- PV/solar controllers
- Robotics



Description

The RN-171-XV is a 802.11 b/g solution especially designed for designers who want to migrate their existing 802.15.4 architecture to a more standard TCP/IP-based platform without redesigning their existing hardware.

Based on a pseudo standard footprint often found in embedded applications, the Roving Networks' RN-171-XV module provides Wi-Fi connectivity using 802.11 b/g standards in legacy and existing designs that may have been based upon the 802.15.4 standard.

The RN-171-XV module is based on Roving Networks' robust RN-171 Wi-Fi module and incorporates an 802.11 b/g radio, a 32-bit SPARC processor, a TCP/IP stack, a real-time clock, a crypto accelerator, power management unit, and an analog sensor interface.

The module offers additional functionality through its on-board programmable GPIO pins (10) and ADCs (8). The ADCs provide 14-bit resolution while the GPIOs can be configured to provide standard functionality or status signaling to a host microcontroller, reducing the need for serial polling between the Wi-Fi module and host microcontroller.

The module is pre-loaded with firmware to simplify integration and minimize applications development time. In the simplest configuration, the hardware only requires four connections (PWR, TX, RX, and GND) to create a wireless data connection.

OVERVIEW

- Host data rates up to 460 Kbps over the UART
- Intelligent built-in power management with programmable wakeup events (timers and I/O)
- Real-time clock for time stamping, auto-sleep, and auto-wakeup modes
- Configuration over Wi-Fi or UART using simple ASCII commands
- Supports over the air firmware upgrade via FTP
- Secure Wi-Fi authentication via WEP, WPA-PSK (TKIP), and WPA2-PSK
- Built-in networking applications: DHCP, DNS, ARP, ICMP UDP, Telnet, FTP client, and HTML client
- Configurable transmit power of 0 to 12 dBm
- WPS mode for easy and secure wireless setup
- Built-in HTML web client to send GPIO, UART, and sensor data to remote web servers

The module's size is:

- *Size*—0.96" x 1.15"
- *Weight*—5.5 g

Tables 1 through 5 provide detailed specifications for the module.

Table 1. Environmental Conditions

Parameter	RN-171-XV
Temperature Range (Operating)	-40 °C ~ +85 °C
Temperature Range (Storage)	-40 °C ~ +85 °C
Relative Humidity (Operating)	≤90%
Relative Humidity (Storage)	≤90%

Table 2. Electrical Characteristics

Supply Voltage	Min	Typ.	Max.	Units
Input Power	3.0	3.3	3.7	V
Power Consumption				
Sleep		4		uA
Standby (doze)		15		mA
Connected (idle, RX)		40		mA
Connected (TX)		180 at 10 dBm		mA

Table 3. Analog Sensor Inputs

Parameter	Value
Sensor 0, 1, 2, 3 wakeup detect threshold	500 mV
A/D sensor 0 - 4 measurement range	0 - 400 mV
Precision	14 bits = 12 uV
Accuracy	5% un-calibrated, .01% calibrated
Minimum conversion time	35 uS (5 kHz over Wi-Fi)
Sensor power (pin 33) output resistance 3.3 V	10 ohms, maximum current = 50 mA

Table 4. Radio Characteristics

Parameter	Specifications
Frequency	2,402 ~ 2,480 MHz
Modulation	802.11b compatibility: DSSS (CCK-11, CCK-5.5, DQPSK-2, and DBPSK-1) 802.11g: OFDM (default)
Channel intervals	5 MHz
Channels	1 - 14
Transmission rate (over the air)	1 – 11 Mbps for 802.11b 6 – 54 Mbps for 802.11g
Receive sensitivity	-83 dBm typical
Output level (Class 1)	0 to +12 dBm (software configurable)

Table 5. Transmit Power

Output Power	802.11b (2 Mbps) Current in mA Note (1)	802.11g (24 Mbps) Current in mA Note (1)
0	120	135
2	130	150
4	170	190
6	175	200
8	180	210
10	185	225
12	190	240

Note:

1. Measured at 3.3-V DC VCC. The power consumption is the average power, active during actual power consumption.

Table 6 describes the LED indicators.

Table 6. RN-171-XV Evaluation Board LED Indicators

Condition	Red LED	Yellow LED	Green LED
On solid	-	-	Connected over TCP
Fast blink	Not associated	Rx/Tx data transfer	No IP address
Slow blink	Associated, no Internet	-	IP address OK
Off	Associated, Internet OK	-	-

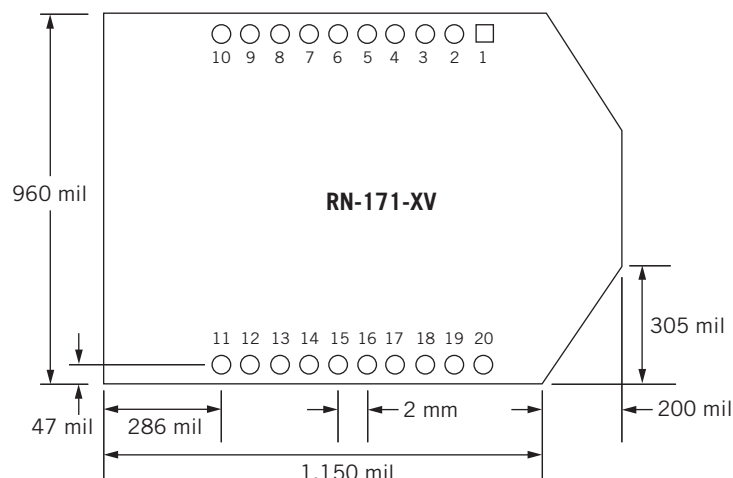
TYPICAL APPLICATION SCHEMATIC

Figure 1 shows a typical application schematic. RN-171-XV connector pins 13, 15, 18, and 19 can be configured as GPIO pins or sensor inputs, depending on the installed resistors. When configured as GPIO pins, you should ONLY install column 1 resistors. When configured as sensor inputs, ONLY stuff column 2 and column 3 resistors. The default configuration is:

- *Sensor inputs*—Pins 19 and 20
- *GPIO*—Pins 13, 15, and 18

[illegible]

Figure 2 shows the pin pads and dimensions. Table 7 describes the pins.

Figure 2. Pin Pads & Dimensions


1 mil = 0.0254 mm

Table 7. Pin Description

Pad Number	Signal Name	Description	Optional Function	Direction
1	VDD_3V3	3.3-V regulated power input to the module.		Power
2	UART_TX	UART TX, 8-mA drive, 3.3-V tolerant.		OUT →
3	UART_RX	UART RX, 3.3-V tolerant.		IN ←
4	GPIO8	GPIO, 24-mA drive, 3.3-V tolerant. The RN-171-XV drives GPIO8 HIGH on powerup, which overrides software configured powerup values, such as set sys value 0x0000 on GPIO8.		I/O
5	RESET	<i>Optional</i> module reset signal (active low), 100-k pull up, apply pulse of at least 160 us, 3.3-V tolerant.		Input
6	GPIO5	GPIO, 24-mA drive, 3.3-V tolerant.	Data TX/RX	OUT →
7	GPIO7	GPIO, 24-mA drive, 3.3-V tolerant.		I/O
8	GPIO9	Enable ad-hoc mode, restore factory defaults, 8 mA drive, 3.3-V tolerant.		I/O
9	GPIO1	GPIO, 8-mA drive, 3.3-V tolerant.		I/O
10	GND	Ground.		Ground
11	GPIO14	GPIO, 8-mA drive, 3.3-V tolerant.		I/O
12	UART_RTS	UART RTS flow control, 8-mA drive, 3.3-V tolerant.		OUT →
13	GPIO4/sensor 6	GPIO, 24-mA drive, 3.3-V tolerant/ADC input, (3.3-V tolerant) . Defaults to GPIO4. <i>Note (1)</i>		I/O
14	Not Used			No Connect
15	GPIO6/ sensor 7	GPIO, 24-mA drive, 3.3-V tolerant/ADC input, (3.3-V tolerant) . Defaults to GPIO6. <i>Note (1)</i>		Power
16	UART_CTS	UART CTS flow control, 3.3-V tolerant.		IN ←
17	Sensor 5	Sensor interface, analog input to module, (3.3-V tolerant) .		Input
18	GPIO3/ sensor 4	GPIO, 8-mA drive, 3.3-V tolerant/ADC input (3.3-V tolerant) . Defaults to GPIO3.		I/O
19	GPIO2/ sensor 3	GPIO, 8-mA drive, 3.3-V tolerant/ADC input (3.3-V tolerant) . Defaults to sensor 3.		I/O
20	Sensor 2	Sensor interface, analog input to module, 3.3-V tolerant .		Input

Note:

- For information on alternate I/O functions, refer to the WiFly advanced user manual.

DESIGN CONCERNS

The following sections provide information on designing with the RN-171-XV module.

Powering the Module

Apply ONLY 3.3 V \pm 10% regulated power to pin 1 (VDD) and pin 10 (ground). The module does not have an on-board voltage regulator and MUST be powered from a regulated 3.3-V power supply.

Antenna

The RN-171-XV ships with a wire antenna. Custom antenna configurations such as chip, U.FL., and reverse polarity SMA connectors are available with extended lead times.

ORDERING INFORMATION

Table 7 provides ordering information. Table 8 provides information on related products.

Table 7. Ordering Information

Part Number	Description
RN-171-XV-W	Standard configuration, industrial temperature (-40 °C to +85 °C) 802.15.4 replacement solution with ¼ inch wire antenna.
RN-171-XV-U	Custom configuration, industrial temperature (-40 °C to +85 °C) 802.15.4 replacement solution with U.FL. connector.
RN-171-XV-S	Custom configuration, industrial temperature (-40 °C to +85 °C) 802.15.4 replacement solution with a reverse polarity SMA connector.

Table 8. Related Products

Part Number	Description
RN-XV-RD2	Two-relay evaluation board for the RN-171-XV.
RN-XV-RD1	One-relay evaluation board for the RN-171-XV.

Go to <http://www.rovingnetworks.com> for current pricing and a list of distributors carrying Roving Networks products.

Roving Networks, Inc.
102 Cooper Court
Los Gatos, CA 95032
+1 (408) 395-5300
www.rovingnetworks.com

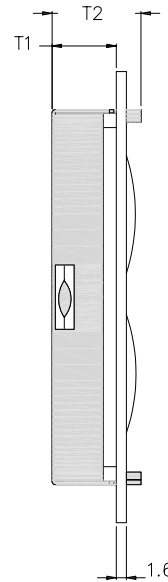
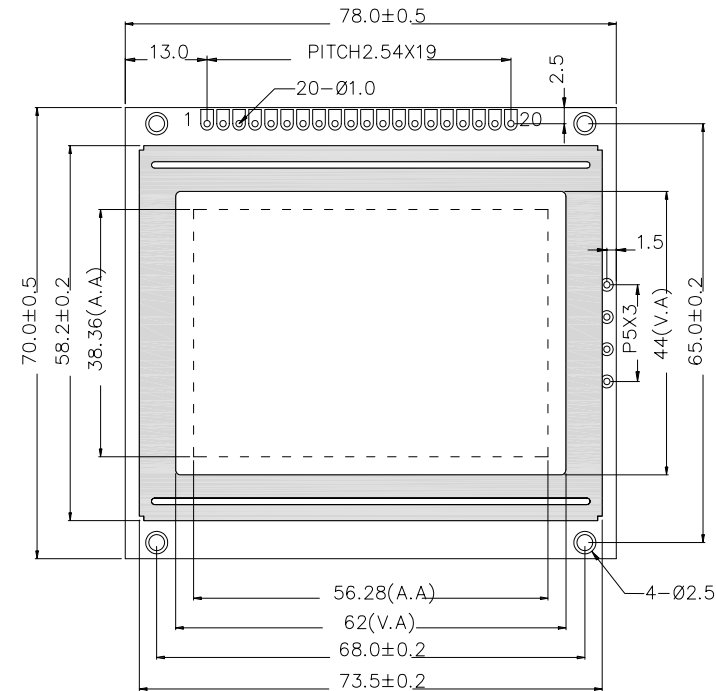
Copyright © 2012 Roving Networks. All rights reserved. Roving Networks is a registered trademark of Roving Networks. Apple Inc., iPhone, iPad, iTunes, Made for iPhone are registered trademarks of Apple Computer.

Roving Networks reserves the right to make corrections, modifications, and other changes to its products, documentation and services at any time. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

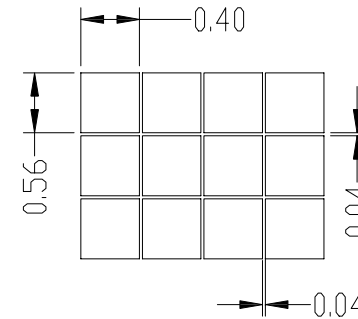
Roving Networks assumes no liability for applications assistance or customer's product design. Customers are responsible for their products and applications that use Roving Networks components. To minimize customer product risks, customers should provide adequate design and operating safeguards.

Roving Networks products are not authorized for use in safety-critical applications (such as life support) where a failure of the Roving Networks product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use.

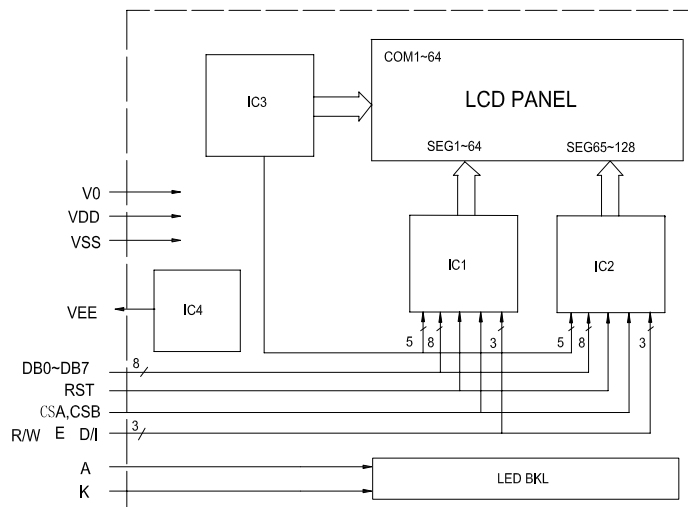
Pin	Symbol
1	CSA
2	CSB
3	VSS
4	VDD
5	V0
6	RST
7	RW
8	E
9~16	DB0~DB7
17	RST
18	VEE
19	A
20	K



DOTS DETAIL



Block diagram



Feature

- 128dots X 64dots matrix.
- Built-in controller (S6B0108)
- Interface with 8-bit ;5V power supply
- 5V;1/64Duty cycle, 1/9Bias.
- Display Mode & Backlight Variations.
- ROHS Compliant

Mechanical thickness

LCD module	T1	T2	Unit
Without or EL	5	10.0	mm
Edge backlight	---	---	mm
Bottom backlight	10.2	15.0	mm

Mechanical data

Item	Standard	Unit
Module size	78.0X70.0	mm
View area	62.0X44.0	mm
Dot size	0.40X0.56	mm
Area size	56.28X38.36	mm

Maximum Absolute Power Ratings

Item	Symbol	Standard			Unit
		Min	Typ	Max	
Input voltage	V _{DD}	-0.3	---	5.5	V
Power supply	V _{in}	-0.3	---	VDD	

Electronical characteristics

Item	Symbol	Condition	Standard			Unit
			Min	Typ	Max	
Operating voltage	V _{DD}	V _{DD} =5.0V	4.5	---	5.5	V
Operating current	I _{DD}	V _{DD} =5.0V	---	7.2	8	mA
Recommended LCD driving voltage for normal temp module	V _{LCD}	T _A =+25	---	8.5	---	V
		T _A =0	---	8.7	---	
		T _A =-20	---	9.1	---	
Backlight voltage	V _{LED}	Y-G	---	4.2	---	V
Backlight current	I _{LED}	V _{LED} =4.2V	---	240	---	mA
Operating temperature	T _O	---	-20	---	70	
Storage temperature	T _S	---	-30	---	80	



LPC1769/68/67/66/65/64/63

32-bit ARM Cortex-M3 microcontroller; up to 512 kB flash and 64 kB SRAM with Ethernet, USB 2.0 Host/Device/OTG, CAN

Rev. 9 — 10 August 2012

Product data sheet



1. General description

The LPC1769/68/67/66/65/64/63 are ARM Cortex-M3 based microcontrollers for embedded applications featuring a high level of integration and low power consumption. The ARM Cortex-M3 is a next generation core that offers system enhancements such as enhanced debug features and a higher level of support block integration.

The LPC1768/67/66/65/64/63 operate at CPU frequencies of up to 100 MHz. The LPC1769 operates at CPU frequencies of up to 120 MHz. The ARM Cortex-M3 CPU incorporates a 3-stage pipeline and uses a Harvard architecture with separate local instruction and data buses as well as a third bus for peripherals. The ARM Cortex-M3 CPU also includes an internal prefetch unit that supports speculative branching.

The peripheral complement of the LPC1769/68/67/66/65/64/63 includes up to 512 kB of flash memory, up to 64 kB of data memory, Ethernet MAC, USB Device/Host/OTG interface, 8-channel general purpose DMA controller, 4 UARTs, 2 CAN channels, 2 SSP controllers, SPI interface, 3 I²C-bus interfaces, 2-input plus 2-output I²S-bus interface, 8-channel 12-bit ADC, 10-bit DAC, motor control PWM, Quadrature Encoder interface, four general purpose timers, 6-output general purpose PWM, ultra-low power Real-Time Clock (RTC) with separate battery supply, and up to 70 general purpose I/O pins.

The LPC1769/68/67/66/65/64/63 are pin-compatible to the 100-pin LPC236x ARM7-based microcontroller series.

2. Features and benefits

- ARM Cortex-M3 processor, running at frequencies of up to 100 MHz (LPC1768/67/66/65/64/63) or of up to 120 MHz (LPC1769). A Memory Protection Unit (MPU) supporting eight regions is included.
- ARM Cortex-M3 built-in Nested Vectored Interrupt Controller (NVIC).
- Up to 512 kB on-chip flash programming memory. Enhanced flash memory accelerator enables high-speed 120 MHz operation with zero wait states.
- In-System Programming (ISP) and In-Application Programming (IAP) via on-chip bootloader software.
- On-chip SRAM includes:
 - ◆ 32/16 kB of SRAM on the CPU with local code/data bus for high-performance CPU access.



- ◆ Two/one 16 kB SRAM blocks with separate access paths for higher throughput. These SRAM blocks may be used for Ethernet, USB, and DMA memory, as well as for general purpose CPU instruction and data storage.
- Eight channel General Purpose DMA controller (GPDMA) on the AHB multilayer matrix that can be used with SSP, I²S-bus, UART, Analog-to-Digital and Digital-to-Analog converter peripherals, timer match signals, and for memory-to-memory transfers.
- Multilayer AHB matrix interconnect provides a separate bus for each AHB master. AHB masters include the CPU, General Purpose DMA controller, Ethernet MAC, and the USB interface. This interconnect provides communication with no arbitration delays.
- Split APB bus allows high throughput with few stalls between the CPU and DMA.
- Serial interfaces:
 - ◆ Ethernet MAC with RMII interface and dedicated DMA controller. (Not available on all parts, see [Table 2](#).)
 - ◆ USB 2.0 full-speed device/Host/OTG controller with dedicated DMA controller and on-chip PHY for device, Host, and OTG functions. (Not available on all parts, see [Table 2](#).)
 - ◆ Four UARTs with fractional baud rate generation, internal FIFO, and DMA support. One UART has modem control I/O and RS-485/EIA-485 support, and one UART has IrDA support.
 - ◆ CAN 2.0B controller with two channels. (Not available on all parts, see [Table 2](#).)
 - ◆ SPI controller with synchronous, serial, full duplex communication and programmable data length.
 - ◆ Two SSP controllers with FIFO and multi-protocol capabilities. The SSP interfaces can be used with the GPDMA controller.
 - ◆ Three enhanced I²C bus interfaces, one with an open-drain output supporting full I²C specification and Fast mode plus with data rates of 1 Mbit/s, two with standard port pins. Enhancements include multiple address recognition and monitor mode.
 - ◆ I²S (Inter-IC Sound) interface for digital audio input or output, with fractional rate control. The I²S-bus interface can be used with the GPDMA. The I²S-bus interface supports 3-wire and 4-wire data transmit and receive as well as master clock input/output. (Not available on all parts, see [Table 2](#).)
- Other peripherals:
 - ◆ 70 (100 pin package) General Purpose I/O (GPIO) pins with configurable pull-up/down resistors. All GPIOs support a new, configurable open-drain operating mode. The GPIO block is accessed through the AHB multilayer bus for fast access and located in memory such that it supports Cortex-M3 bit banding and use by the General Purpose DMA Controller.
 - ◆ 12-bit Analog-to-Digital Converter (ADC) with input multiplexing among eight pins, conversion rates up to 200 kHz, and multiple result registers. The 12-bit ADC can be used with the GPDMA controller.
 - ◆ 10-bit Digital-to-Analog Converter (DAC) with dedicated conversion timer and DMA support. (Not available on all parts, see [Table 2](#).)
 - ◆ Four general purpose timers/counters, with a total of eight capture inputs and ten compare outputs. Each timer block has an external count input. Specific timer events can be selected to generate DMA requests.
 - ◆ One motor control PWM with support for three-phase motor control.

- ◆ Quadrature encoder interface that can monitor one external quadrature encoder.
- ◆ One standard PWM/timer block with external count input.
- ◆ RTC with a separate power domain and dedicated RTC oscillator. The RTC block includes 20 bytes of battery-powered backup registers.
- ◆ WatchDog Timer (WDT). The WDT can be clocked from the internal RC oscillator, the RTC oscillator, or the APB clock.
- ◆ ARM Cortex-M3 system tick timer, including an external clock input option.
- ◆ Repetitive interrupt timer provides programmable and repeating timed interrupts.
- ◆ Each peripheral has its own clock divider for further power savings.
- Standard JTAG test/debug interface for compatibility with existing tools. Serial Wire Debug and Serial Wire Trace Port options.
- Emulation trace module enables non-intrusive, high-speed real-time tracing of instruction execution.
- Integrated PMU (Power Management Unit) automatically adjusts internal regulators to minimize power consumption during Sleep, Deep sleep, Power-down, and Deep power-down modes.
- Four reduced power modes: Sleep, Deep-sleep, Power-down, and Deep power-down.
- Single 3.3 V power supply (2.4 V to 3.6 V).
- Four external interrupt inputs configurable as edge/level sensitive. All pins on Port 0 and Port 2 can be used as edge sensitive interrupt sources.
- Non-maskable Interrupt (NMI) input.
- Clock output function that can reflect the main oscillator clock, IRC clock, RTC clock, CPU clock, and the USB clock.
- The Wake-up Interrupt Controller (WIC) allows the CPU to automatically wake up from any priority interrupt that can occur while the clocks are stopped in deep sleep, Power-down, and Deep power-down modes.
- Processor wake-up from Power-down mode via any interrupt able to operate during Power-down mode (includes external interrupts, RTC interrupt, USB activity, Ethernet wake-up interrupt, CAN bus activity, Port 0/2 pin interrupt, and NMI).
- Brownout detect with separate threshold for interrupt and forced reset.
- Power-On Reset (POR).
- Crystal oscillator with an operating range of 1 MHz to 25 MHz.
- 4 MHz internal RC oscillator trimmed to 1 % accuracy that can optionally be used as a system clock.
- PLL allows CPU operation up to the maximum CPU rate without the need for a high-frequency crystal. May be run from the main oscillator, the internal RC oscillator, or the RTC oscillator.
- USB PLL for added flexibility.
- Code Read Protection (CRP) with different security levels.
- Unique device serial number for identification purposes.
- Available as LQFP100 (14 mm × 14 mm × 1.4 mm) and TFBGA100¹ (9 mm × 9 mm × 0.7 mm) package.

1. LPC1768/65 only.

3. Applications

- eMetering
- Lighting
- Industrial networking
- Alarm systems
- White goods
- Motor control

4. Ordering information

Table 1. Ordering information

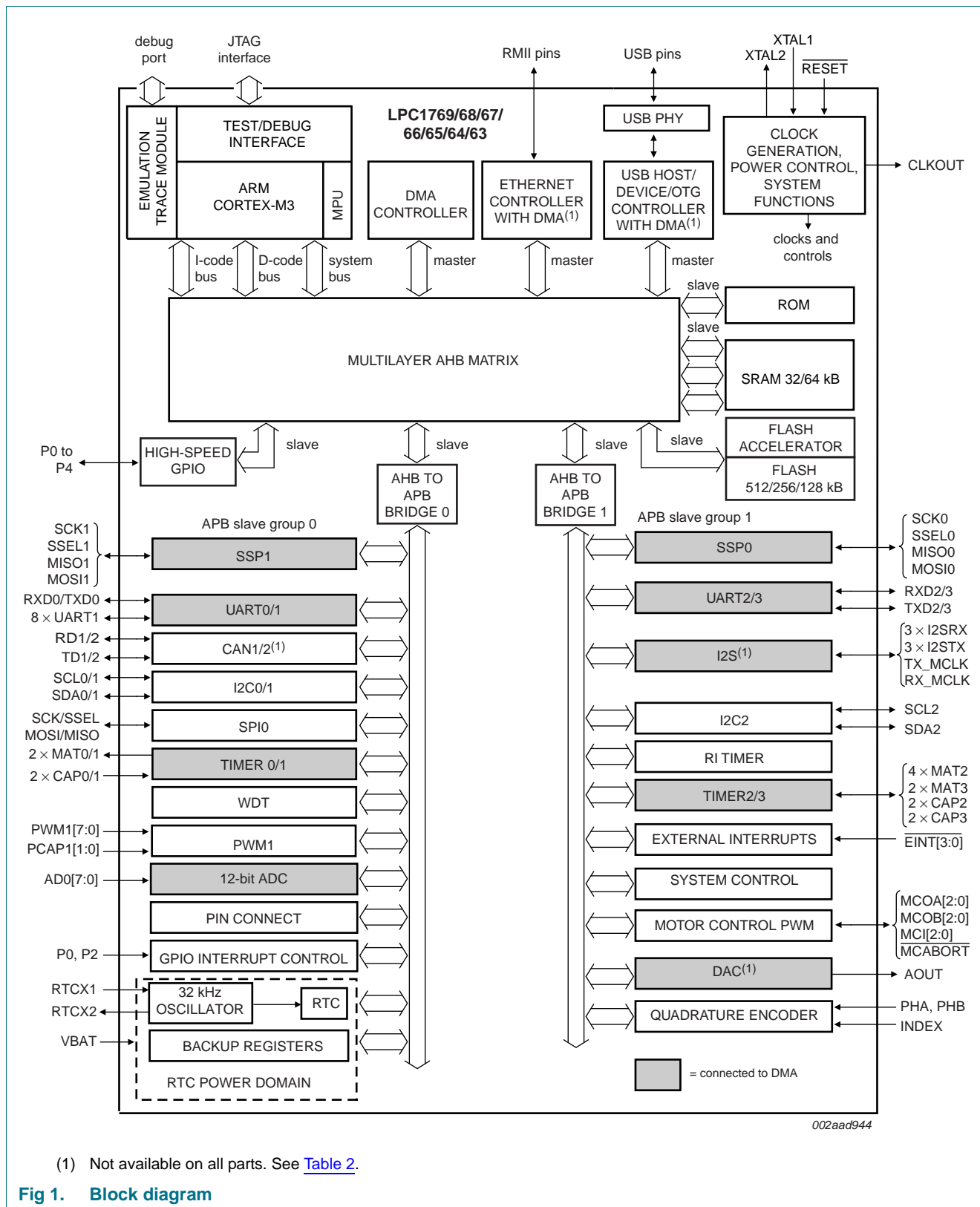
Type number	Package		
	Name	Description	Version
LPC1769FBD100	LQFP100	plastic low profile quad flat package; 100 leads; body 14 × 14 × 1.4 mm	SOT407-1
LPC1768FBD100	LQFP100	plastic low profile quad flat package; 100 leads; body 14 × 14 × 1.4 mm	SOT407-1
LPC1768FET100	TFBGA100	plastic thin fine-pitch ball grid array package; 100 balls; body 9 × 9 × 0.7 mm	SOT926-1
LPC1767FBD100	LQFP100	plastic low profile quad flat package; 100 leads; body 14 × 14 × 1.4 mm	SOT407-1
LPC1766FBD100	LQFP100	plastic low profile quad flat package; 100 leads; body 14 × 14 × 1.4 mm	SOT407-1
LPC1765FBD100	LQFP100	plastic low profile quad flat package; 100 leads; body 14 × 14 × 1.4 mm	SOT407-1
LPC1765FET100	TFBGA100	plastic thin fine-pitch ball grid array package; 100 balls; body 9 × 9 × 0.7 mm	SOT926-1
LPC1764FBD100	LQFP100	plastic low profile quad flat package; 100 leads; body 14 × 14 × 1.4 mm	SOT407-1
LPC1763FBD100	LQFP100	plastic low profile quad flat package; 100 leads; body 14 × 14 × 1.4 mm	SOT407-1

4.1 Ordering options

Table 2. Ordering options

Type number	Flash	SRAM in kB				Ethernet	USB	CAN	I ² S	DAC	Maximum CPU operating frequency
		CPU	AHB SRAM0	AHB SRAM1	Total						
LPC1769FBD100	512 kB	32	16	16	64	yes	Device/Host/OTG	2	yes	yes	120 MHz
LPC1768FBD100	512 kB	32	16	16	64	yes	Device/Host/OTG	2	yes	yes	100 MHz
LPC1768FET100	512 kB	32	16	16	64	yes	Device/Host/OTG	2	yes	yes	100 MHz
LPC1767FBD100	512 kB	32	16	16	64	yes	no	no	yes	yes	100 MHz
LPC1766FBD100	256 kB	32	16	16	64	yes	Device/Host/OTG	2	yes	yes	100 MHz
LPC1765FBD100	256 kB	32	16	16	64	no	Device/Host/OTG	2	yes	yes	100 MHz
LPC1765FET100	256 kB	32	16	16	64	no	Device/Host/OTG	2	yes	yes	100 MHz
LPC1764FBD100	128 kB	16	16	-	32	yes	Device only	2	no	no	100 MHz
LPC1763FBD100	256 kB	32	16	16	64	no	no	no	yes	yes	100 MHz

5. Block diagram



6. Pinning information

6.1 Pinning

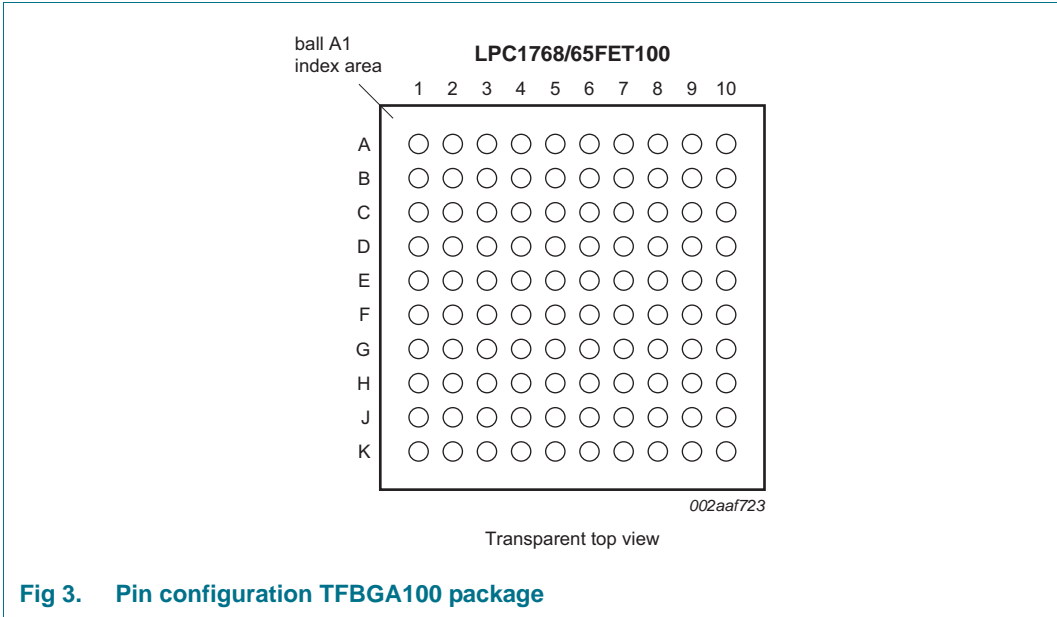
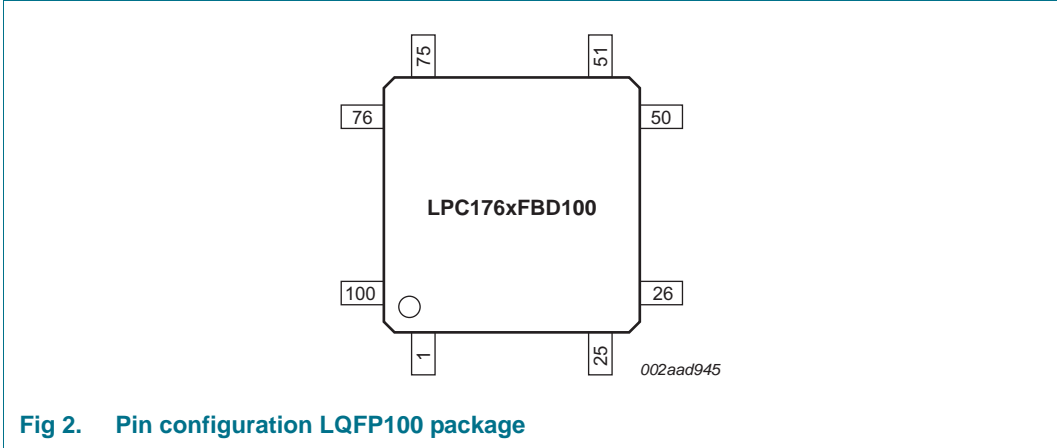


Table 3. Pin allocation table

Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol
Row A							
1	TDO/SWO	2	P0[3]/RXD0/AD0[6]	3	V _{DD} (3V3)	4	P1[4]/ENET_TX_EN
5	P1[10]/ENET_RXD1	6	P1[16]/ENET_MDC	7	V _{DD} (REG)(3V3)	8	P0[4]/I2SRX_CLK/ RD2/CAP2[0]
9	P0[7]/I2STX_CLK/ SCK1/MAT2[1]	10	P0[9]/I2STX_SDA/ MOSI1/MAT2[3]	11	-	12	-
Row B							
1	TMS/SWDIO	2	RTCK	3	V _{SS}	4	P1[1]/ENET_TXD1

Table 3. Pin allocation table ...continued

Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol
5	P1[9]/ENET_RXD0	6	P1[17]/ ENET_MDIO	7	V _{SS}	8	P0[6]/I2SRX_SDA/ SSEL1/MAT2[0]
9	P2[0]/PWM1[1]/TXD1	10	P2[1]/PWM1[2]/RXD1	11	-	12	-
Row C							
1	TCK/SWDCLK	2	$\overline{\text{TRST}}$	3	TDI	4	P0[2]/TXD0/AD0[7]
5	P1[8]/ENET_CRD	6	P1[15]/ ENET_REF_CLK	7	P4[28]/RX_MCLK/ MAT2[0]/TXD3	8	P0[8]/I2STX_WS/ MISO1/MAT2[2]
9	V _{SS}	10	V _{DD(3V3)}	11	-	12	-
Row D							
1	P0[24]/AD0[1]/ I2SRX_WS/CAP3[1]	2	P0[25]/AD0[2]/ I2SRX_SDA/TXD3	3	P0[26]/AD0[3]/ AOUT/RXD3	4	n.c.
5	P1[0]/ENET_TXD0	6	P1[14]/ENET_RX_ER	7	P0[5]/I2SRX_WS/ TD2/CAP2[1]	8	P2[2]/PWM1[3]/ CTS1/TRACEDATA[3]
9	P2[4]/PWM1[5]/ DSR1/TRACEDATA[1]	10	P2[5]/PWM1[6]/ DTR1/TRACEDATA[0]	11	-	12	-
Row E							
1	V _{SSA}	2	V _{DDA}	3	VREFP	4	n.c.
5	P0[23]/AD0[0]/ I2SRX_CLK/CAP3[0]	6	P4[29]/TX_MCLK/ MAT2[1]/RXD3	7	P2[3]/PWM1[4]/ DCD1/TRACEDATA[2]	8	P2[6]/PCAP1[0]/ RI1/TRACECLK
9	P2[7]/RD2/RTS1	10	P2[8]/TD2/TXD2	11	-	12	-
Row F							
1	VREFN	2	RTCX1	3	$\overline{\text{RESET}}$	4	P1[31]/SCK1/ AD0[5]
5	P1[21]/ $\overline{\text{MCABORT}}$ / PWM1[3]/SSEL0	6	P0[18]/DCD1/ MOSI0/MOSI	7	P2[9]/USB_CONNECT/ RXD2	8	P0[16]/RXD1/ SSEL0/SSEL
9	P0[17]/CTS1/ MISO0/MISO	10	P0[15]/TXD1/ SCK0/SCK	11	-	12	-
Row G							
1	RTCX2	2	VBAT	3	XTAL2	4	P0[30]/USB_D-
5	P1[25]/MCOA1/ MAT1[1]	6	P1[29]/MCOB2/ PCAP1[1]/MAT0[1]	7	V _{SS}	8	P0[21]/RI1/RD1
9	P0[20]/DTR1/SCL1	10	P0[19]/DSR1/SDA1	11	-	12	-
Row H							
1	P1[30]/V _{BUS} / AD0[4]	2	XTAL1	3	P3[25]/MAT0[0]/ PWM1[2]	4	P1[18]/USB_UP_LED/ PWM1[1]/CAP1[0]
5	P1[24]/MCI2/ PWM1[5]/MOSI0	6	V _{DD(REG)(3V3)}	7	P0[10]/TXD2/ SDA2/MAT3[0]	8	P2[11]/EINT1/ I2STX_CLK
9	V _{DD(3V3)}	10	P0[22]/RTS1/TD1	11	-	12	-

Table 3. Pin allocation table ...continued

Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol
Row J							
1	P0[28]/SCL0/ USB_SCL	2	P0[27]/SDA0/ USB_SDA	3	P0[29]/USB_D+	4	P1[19]/MCOA0/ USB_PPWR/ CAP1[1]
5	P1[22]/MCOB0/ USB_PWRD/ MAT1[0]	6	V _{SS}	7	P1[28]/MCOA2/ PCAP1[0]/ MAT0[0]	8	P0[1]/TD1/RXD3/SCL1
9	P2[13]/EINT3/ I2STX_SDA	10	P2[10]/EINT0/NMI	11	-	12	-
Row K							
1	P3[26]/STCLK/ MAT0[1]/PWM1[3]	2	V _{DD} (3V3)	3	V _{SS}	4	P1[20]/MCI0/ PWM1[2]/SCK0
5	P1[23]/MCI1/ PWM1[4]/MISO0	6	P1[26]/MCOB1/ PWM1[6]/CAP0[0]	7	P1[27]/CLKOUT /USB_OVRCR/ CAP0[1]	8	P0[0]/RD1/TXD3/SDA1
9	P0[11]/RXD2/ SCL2/MAT3[1]	10	P2[12]/EINT2/ I2STX_WS	11	-	12	-

6.2 Pin description

Table 4. Pin description

Symbol	Pin	Ball	Type	Description
P0[0] to P0[31]			I/O	Port 0: Port 0 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 0 pins depends upon the pin function selected via the pin connect block. Pins 12, 13, 14, and 31 of this port are not available.
P0[0]/RD1/TXD3/ SDA1	46 ^[1]	K8 ^[1]	I/O	P0[0] — General purpose digital input/output pin.
			I	RD1 — CAN1 receiver input. (LPC1769/68/66/65/64 only).
			O	TXD3 — Transmitter output for UART3.
			I/O	SDA1 — I ² C1 data input/output. (This is not an I ² C-bus compliant open-drain pin).
P0[1]/TD1/RXD3/ SCL1	47 ^[1]	J8 ^[1]	I/O	P0[1] — General purpose digital input/output pin.
			O	TD1 — CAN1 transmitter output. (LPC1769/68/66/65/64 only).
			I	RXD3 — Receiver input for UART3.
			I/O	SCL1 — I ² C1 clock input/output. (This is not an I ² C-bus compliant open-drain pin).
P0[2]/TXD0/AD0[7]	98 ^[2]	C4 ^[2]	I/O	P0[2] — General purpose digital input/output pin.
			O	TXD0 — Transmitter output for UART0.
			I	AD0[7] — A/D converter 0, input 7.
P0[3]/RXD0/AD0[6]	99 ^[2]	A2 ^[2]	I/O	P0[3] — General purpose digital input/output pin.
			I	RXD0 — Receiver input for UART0.
			I	AD0[6] — A/D converter 0, input 6.

Table 4. Pin description ...continued

Symbol	Pin	Ball	Type	Description
P0[4]/ I2SRX_CLK/ RD2/CAP2[0]	81 ^[1]	A8 ^[1]	I/O	P0[4] — General purpose digital input/output pin.
			I/O	I2SRX_CLK — Receive Clock. It is driven by the master and received by the slave. Corresponds to the signal SCK in the <i>I²S-bus specification</i> . (LPC1769/68/67/66/65/63 only).
			I	RD2 — CAN2 receiver input. (LPC1769/68/66/65/64 only).
			I	CAP2[0] — Capture input for Timer 2, channel 0.
P0[5]/ I2SRX_WS/ TD2/CAP2[1]	80 ^[1]	D7 ^[1]	I/O	P0[5] — General purpose digital input/output pin.
			I/O	I2SRX_WS — Receive Word Select. It is driven by the master and received by the slave. Corresponds to the signal WS in the <i>I²S-bus specification</i> . (LPC1769/68/67/66/65/63 only).
			O	TD2 — CAN2 transmitter output. (LPC1769/68/66/65/64 only).
			I	CAP2[1] — Capture input for Timer 2, channel 1.
P0[6]/ I2SRX_SDA/ SSEL1/MAT2[0]	79 ^[1]	B8 ^[1]	I/O	P0[6] — General purpose digital input/output pin.
			I/O	I2SRX_SDA — Receive data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the <i>I²S-bus specification</i> . (LPC1769/68/67/66/65/63 only).
			I/O	SSEL1 — Slave Select for SSP1.
			O	MAT2[0] — Match output for Timer 2, channel 0.
P0[7]/ I2STX_CLK/ SCK1/MAT2[1]	78 ^[1]	A9 ^[1]	I/O	P0[7] — General purpose digital input/output pin.
			I/O	I2STX_CLK — Transmit Clock. It is driven by the master and received by the slave. Corresponds to the signal SCK in the <i>I²S-bus specification</i> . (LPC1769/68/67/66/65/63 only).
			I/O	SCK1 — Serial Clock for SSP1.
			O	MAT2[1] — Match output for Timer 2, channel 1.
P0[8]/ I2STX_WS/ MISO1/MAT2[2]	77 ^[1]	C8 ^[1]	I/O	P0[8] — General purpose digital input/output pin.
			I/O	I2STX_WS — Transmit Word Select. It is driven by the master and received by the slave. Corresponds to the signal WS in the <i>I²S-bus specification</i> . (LPC1769/68/67/66/65/63 only).
			I/O	MISO1 — Master In Slave Out for SSP1.
			O	MAT2[2] — Match output for Timer 2, channel 2.
P0[9]/ I2STX_SDA/ MOSI1/MAT2[3]	76 ^[1]	A10 ^[1]	I/O	P0[9] — General purpose digital input/output pin.
			I/O	I2STX_SDA — Transmit data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the <i>I²S-bus specification</i> . (LPC1769/68/67/66/65/63 only).
			I/O	MOSI1 — Master Out Slave In for SSP1.
			O	MAT2[3] — Match output for Timer 2, channel 3.
P0[10]/TXD2/ SDA2/MAT3[0]	48 ^[1]	H7 ^[1]	I/O	P0[10] — General purpose digital input/output pin.
			O	TXD2 — Transmitter output for UART2.
			I/O	SDA2 — I ² C2 data input/output (this is not an open-drain pin).
			O	MAT3[0] — Match output for Timer 3, channel 0.
P0[11]/RXD2/ SCL2/MAT3[1]	49 ^[1]	K9 ^[1]	I/O	P0[11] — General purpose digital input/output pin.
			I	RXD2 — Receiver input for UART2.
			I/O	SCL2 — I ² C2 clock input/output (this is not an open-drain pin).
			O	MAT3[1] — Match output for Timer 3, channel 1.

Table 4. Pin description ...continued

Symbol	Pin	Ball	Type	Description
P0[15]/TXD1/ SCK0/SCK	62 ^[1]	F10 ^[1]	I/O	P0[15] — General purpose digital input/output pin.
			O	TXD1 — Transmitter output for UART1.
			I/O	SCK0 — Serial clock for SSP0.
			I/O	SCK — Serial clock for SPI.
P0[16]/RXD1/ SSEL0/SSEL	63 ^[1]	F8 ^[1]	I/O	P0[16] — General purpose digital input/output pin.
			I	RXD1 — Receiver input for UART1.
			I/O	SSEL0 — Slave Select for SSP0.
			I/O	SSEL — Slave Select for SPI.
P0[17]/CTS1/ MISO0/MISO	61 ^[1]	F9 ^[1]	I/O	P0[17] — General purpose digital input/output pin.
			I	CTS1 — Clear to Send input for UART1.
			I/O	MISO0 — Master In Slave Out for SSP0.
			I/O	MISO — Master In Slave Out for SPI.
P0[18]/DCD1/ MOSI0/MOSI	60 ^[1]	F6 ^[1]	I/O	P0[18] — General purpose digital input/output pin.
			I	DCD1 — Data Carrier Detect input for UART1.
			I/O	MOSI0 — Master Out Slave In for SSP0.
			I/O	MOSI — Master Out Slave In for SPI.
P0[19]/DSR1/ SDA1	59 ^[1]	G10 ^[1]	I/O	P0[19] — General purpose digital input/output pin.
			I	DSR1 — Data Set Ready input for UART1.
			I/O	SDA1 — I ² C1 data input/output (this is not an I ² C-bus compliant open-drain pin).
P0[20]/DTR1/SCL1	58 ^[1]	G9 ^[1]	I/O	P0[20] — General purpose digital input/output pin.
			O	DTR1 — Data Terminal Ready output for UART1. Can also be configured to be an RS-485/EIA-485 output enable signal.
			I/O	SCL1 — I ² C1 clock input/output (this is not an I ² C-bus compliant open-drain pin).
P0[21]/RI1/RD1	57 ^[1]	G8 ^[1]	I/O	P0[21] — General purpose digital input/output pin.
			I	RI1 — Ring Indicator input for UART1.
			I	RD1 — CAN1 receiver input. (LPC1769/68/66/65/64 only).
P0[22]/RTS1/TD1	56 ^[1]	H10 ^[1]	I/O	P0[22] — General purpose digital input/output pin.
			O	RTS1 — Request to Send output for UART1. Can also be configured to be an RS-485/EIA-485 output enable signal.
			O	TD1 — CAN1 transmitter output. (LPC1769/68/66/65/64 only).
P0[23]/AD0[0]/ I2SRX_CLK/ CAP3[0]	9 ^[2]	E5 ^[2]	I/O	P0[23] — General purpose digital input/output pin.
			I	AD0[0] — A/D converter 0, input 0.
			I/O	I2SRX_CLK — Receive Clock. It is driven by the master and received by the slave. Corresponds to the signal SCK in the I ² S-bus specification. (LPC1769/68/67/66/65/63 only).
			I	CAP3[0] — Capture input for Timer 3, channel 0.

Table 4. Pin description ...continued

Symbol	Pin	Ball	Type	Description
P0[24]/AD0[1]/ I2SRX_WS/ CAP3[1]	8 ^[2]	D1 ^[2]	I/O	P0[24] — General purpose digital input/output pin.
			I	AD0[1] — A/D converter 0, input 1.
			I/O	I2SRX_WS — Receive Word Select. It is driven by the master and received by the slave. Corresponds to the signal WS in the <i>I²S-bus specification</i> . (LPC1769/68/67/66/65/63 only).
			I	CAP3[1] — Capture input for Timer 3, channel 1.
P0[25]/AD0[2]/ I2SRX_SDA/ TXD3	7 ^[2]	D2 ^[2]	I/O	P0[25] — General purpose digital input/output pin.
			I	AD0[2] — A/D converter 0, input 2.
			I/O	I2SRX_SDA — Receive data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the <i>I²S-bus specification</i> . (LPC1769/68/67/66/65/63 only).
			O	TXD3 — Transmitter output for UART3.
P0[26]/AD0[3]/ AOUT/RXD3	6 ^[3]	D3 ^[3]	I/O	P0[26] — General purpose digital input/output pin.
			I	AD0[3] — A/D converter 0, input 3.
			O	AOUT — DAC output (LPC1769/68/67/66/65/63 only).
			I	RXD3 — Receiver input for UART3.
P0[27]/SDA0/ USB_SDA	25 ^[4]	J2 ^[4]	I/O	P0[27] — General purpose digital input/output pin. Output is open-drain.
			I/O	SDA0 — I ² C0 data input/output. Open-drain output (for I ² C-bus compliance).
			I/O	USB_SDA — USB port I ² C serial data (OTG transceiver, LPC1769/68/66/65 only).
P0[28]/SCL0/ USB_SCL	24 ^[4]	J1 ^[4]	I/O	P0[28] — General purpose digital input/output pin. Output is open-drain.
			I/O	SCL0 — I ² C0 clock input/output. Open-drain output (for I ² C-bus compliance).
			I/O	USB_SCL — USB port I ² C serial clock (OTG transceiver, LPC1769/68/66/65 only).
P0[29]/USB_D+	29 ^[5]	J3 ^[5]	I/O	P0[29] — General purpose digital input/output pin.
			I/O	USB_D+ — USB bidirectional D+ line. (LPC1769/68/66/65/64 only).
P0[30]/USB_D–	30 ^[5]	G4 ^[5]	I/O	P0[30] — General purpose digital input/output pin.
			I/O	USB_D– — USB bidirectional D– line. (LPC1769/68/66/65/64 only).
P1[0] to P1[31]			I/O	Port 1: Port 1 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 1 pins depends upon the pin function selected via the pin connect block. Pins 2, 3, 5, 6, 7, 11, 12, and 13 of this port are not available.
P1[0]/ ENET_TXD0	95 ^[1]	D5 ^[1]	I/O	P1[0] — General purpose digital input/output pin.
			O	ENET_TXD0 — Ethernet transmit data 0. (LPC1769/68/67/66/64 only).
P1[1]/ ENET_TXD1	94 ^[1]	B4 ^[1]	I/O	P1[1] — General purpose digital input/output pin.
			O	ENET_TXD1 — Ethernet transmit data 1. (LPC1769/68/67/66/64 only).
P1[4]/ ENET_TX_EN	93 ^[1]	A4 ^[1]	I/O	P1[4] — General purpose digital input/output pin.
			O	ENET_TX_EN — Ethernet transmit data enable. (LPC1769/68/67/66/64 only).
P1[8]/ ENET_CRS	92 ^[1]	C5 ^[1]	I/O	P1[8] — General purpose digital input/output pin.
			I	ENET_CRS — Ethernet carrier sense. (LPC1769/68/67/66/64 only).

Table 4. Pin description ...continued

Symbol	Pin	Ball	Type	Description
P1[9]/ ENET_RXD0	91 ^[1]	B5 ^[1]	I/O	P1[9] — General purpose digital input/output pin.
			I	ENET_RXD0 — Ethernet receive data. (LPC1769/68/67/66/64 only).
P1[10]/ ENET_RXD1	90 ^[1]	A5 ^[1]	I/O	P1[10] — General purpose digital input/output pin.
			I	ENET_RXD1 — Ethernet receive data. (LPC1769/68/67/66/64 only).
P1[14]/ ENET_RX_ER	89 ^[1]	D6 ^[1]	I/O	P1[14] — General purpose digital input/output pin.
			I	ENET_RX_ER — Ethernet receive error. (LPC1769/68/67/66/64 only).
P1[15]/ ENET_REF_CLK	88 ^[1]	C6 ^[1]	I/O	P1[15] — General purpose digital input/output pin.
			I	ENET_REF_CLK — Ethernet reference clock. (LPC1769/68/67/66/64 only).
P1[16]/ ENET_MDC	87 ^[1]	A6 ^[1]	I/O	P1[16] — General purpose digital input/output pin.
			O	ENET_MDC — Ethernet MIIM clock (LPC1769/68/67/66/64 only).
P1[17]/ ENET_MDIO	86 ^[1]	B6 ^[1]	I/O	P1[17] — General purpose digital input/output pin.
			I/O	ENET_MDIO — Ethernet MIIM data input and output. (LPC1769/68/67/66/64 only).
P1[18]/ USB_UP_LED/ PWM1[1]/ CAP1[0]	32 ^[1]	H4 ^[1]	I/O	P1[18] — General purpose digital input/output pin.
			O	USB_UP_LED — USB GoodLink LED indicator. It is LOW when the device is configured (non-control endpoints enabled), or when the host is enabled and has detected a device on the bus. It is HIGH when the device is not configured, or when host is enabled and has not detected a device on the bus, or during global suspend. It transitions between LOW and HIGH (flashes) when the host is enabled and detects activity on the bus. (LPC1769/68/66/65/64 only).
			O	PWM1[1] — Pulse Width Modulator 1, channel 1 output.
			I	CAP1[0] — Capture input for Timer 1, channel 0.
P1[19]/MCOA0/ USB_PPWR/ CAP1[1]	33 ^[1]	J4 ^[1]	I/O	P1[19] — General purpose digital input/output pin.
			O	MCOA0 — Motor control PWM channel 0, output A.
			O	USB_PPWR — Port Power enable signal for USB port. (LPC1769/68/66/65 only).
			I	CAP1[1] — Capture input for Timer 1, channel 1.
P1[20]/MCI0/ PWM1[2]/SCK0	34 ^[1]	K4 ^[1]	I/O	P1[20] — General purpose digital input/output pin.
			I	MCI0 — Motor control PWM channel 0, input. Also Quadrature Encoder Interface PHA input.
			O	PWM1[2] — Pulse Width Modulator 1, channel 2 output.
			I/O	SCK0 — Serial clock for SSP0.
P1[21]/MCABORT/ PWM1[3]/ SSEL0	35 ^[1]	F5 ^[1]	I/O	P1[21] — General purpose digital input/output pin.
			O	MCABORT — Motor control PWM, LOW-active fast abort.
			O	PWM1[3] — Pulse Width Modulator 1, channel 3 output.
			I/O	SSEL0 — Slave Select for SSP0.
P1[22]/MCOB0/ USB_PWRD/ MAT1[0]	36 ^[1]	J5 ^[1]	I/O	P1[22] — General purpose digital input/output pin.
			O	MCOB0 — Motor control PWM channel 0, output B.
			I	USB_PWRD — Power Status for USB port (host power switch, LPC1769/68/66/65 only).
			O	MAT1[0] — Match output for Timer 1, channel 0.

Table 4. Pin description ...continued

Symbol	Pin	Ball	Type	Description
P1[23]/MCI1/ PWM1[4]/MISO0	37 ^[1]	K5 ^[1]	I/O	P1[23] — General purpose digital input/output pin.
			I	MCI1 — Motor control PWM channel 1, input. Also Quadrature Encoder Interface PHB input.
			O	PWM1[4] — Pulse Width Modulator 1, channel 4 output.
			I/O	MISO0 — Master In Slave Out for SSP0.
P1[24]/MCI2/ PWM1[5]/MOSI0	38 ^[1]	H5 ^[1]	I/O	P1[24] — General purpose digital input/output pin.
			I	MCI2 — Motor control PWM channel 2, input. Also Quadrature Encoder Interface INDEX input.
			O	PWM1[5] — Pulse Width Modulator 1, channel 5 output.
			I/O	MOSI0 — Master Out Slave in for SSP0.
P1[25]/MCOA1/ MAT1[1]	39 ^[1]	G5 ^[1]	I/O	P1[25] — General purpose digital input/output pin.
			O	MCOA1 — Motor control PWM channel 1, output A.
			O	MAT1[1] — Match output for Timer 1, channel 1.
P1[26]/MCOB1/ PWM1[6]/CAP0[0]	40 ^[1]	K6 ^[1]	I/O	P1[26] — General purpose digital input/output pin.
			O	MCOB1 — Motor control PWM channel 1, output B.
			O	PWM1[6] — Pulse Width Modulator 1, channel 6 output.
			I	CAP0[0] — Capture input for Timer 0, channel 0.
P1[27]/CLKOUT /USB_OVRCCR/ CAP0[1]	43 ^[1]	K7 ^[1]	I/O	P1[27] — General purpose digital input/output pin.
			O	CLKOUT — Clock output pin.
			I	USB_OVRCCR — USB port Over-Current status. (LPC1769/68/66/65 only).
			I	CAP0[1] — Capture input for Timer 0, channel 1.
P1[28]/MCOA2/ PCAP1[0]/ MAT0[0]	44 ^[1]	J7 ^[1]	I/O	P1[28] — General purpose digital input/output pin.
			O	MCOA2 — Motor control PWM channel 2, output A.
			I	PCAP1[0] — Capture input for PWM1, channel 0.
			O	MAT0[0] — Match output for Timer 0, channel 0.
P1[29]/MCOB2/ PCAP1[1]/ MAT0[1]	45 ^[1]	G6 ^[1]	I/O	P1[29] — General purpose digital input/output pin.
			O	MCOB2 — Motor control PWM channel 2, output B.
			I	PCAP1[1] — Capture input for PWM1, channel 1.
			O	MAT0[1] — Match output for Timer 0, channel 1.
P1[30]/V _{BUS} / AD0[4]	21 ^[2]	H1 ^[2]	I/O	P1[30] — General purpose digital input/output pin.
			I	V_{BUS} — Monitors the presence of USB bus power. (LPC1769/68/66/65/64 only). Note: This signal must be HIGH for USB reset to occur.
			I	AD0[4] — A/D converter 0, input 4.
P1[31]/SCK1/ AD0[5]	20 ^[2]	F4 ^[2]	I/O	P1[31] — General purpose digital input/output pin.
			I/O	SCK1 — Serial Clock for SSP1.
			I	AD0[5] — A/D converter 0, input 5.
P2[0] to P2[31]			I/O	Port 2: Port 2 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 2 pins depends upon the pin function selected via the pin connect block. Pins 14 through 31 of this port are not available.

Table 4. Pin description ...continued

Symbol	Pin	Ball	Type	Description
P2[0]/PWM1[1]/TXD1	75 ^[1]	B9 ^[1]	I/O	P2[0] — General purpose digital input/output pin.
			O	PWM1[1] — Pulse Width Modulator 1, channel 1 output.
			O	TXD1 — Transmitter output for UART1.
P2[1]/PWM1[2]/RXD1	74 ^[1]	B10 ^[1]	I/O	P2[1] — General purpose digital input/output pin.
			O	PWM1[2] — Pulse Width Modulator 1, channel 2 output.
			I	RXD1 — Receiver input for UART1.
P2[2]/PWM1[3]/CTS1/TRACEDATA[3]	73 ^[1]	D8 ^[1]	I/O	P2[2] — General purpose digital input/output pin.
			O	PWM1[3] — Pulse Width Modulator 1, channel 3 output.
			I	CTS1 — Clear to Send input for UART1.
			O	TRACEDATA[3] — Trace data, bit 3.
P2[3]/PWM1[4]/DCD1/TRACEDATA[2]	70 ^[1]	E7 ^[1]	I/O	P2[3] — General purpose digital input/output pin.
			O	PWM1[4] — Pulse Width Modulator 1, channel 4 output.
			I	DCD1 — Data Carrier Detect input for UART1.
			O	TRACEDATA[2] — Trace data, bit 2.
P2[4]/PWM1[5]/DSR1/TRACEDATA[1]	69 ^[1]	D9 ^[1]	I/O	P2[4] — General purpose digital input/output pin.
			O	PWM1[5] — Pulse Width Modulator 1, channel 5 output.
			I	DSR1 — Data Set Ready input for UART1.
			O	TRACEDATA[1] — Trace data, bit 1.
P2[5]/PWM1[6]/DTR1/TRACEDATA[0]	68 ^[1]	D10 ^[1]	I/O	P2[5] — General purpose digital input/output pin.
			O	PWM1[6] — Pulse Width Modulator 1, channel 6 output.
			O	DTR1 — Data Terminal Ready output for UART1. Can also be configured to be an RS-485/EIA-485 output enable signal.
			O	TRACEDATA[0] — Trace data, bit 0.
P2[6]/PCAP1[0]/RI1/TRACECLK	67 ^[1]	E8 ^[1]	I/O	P2[6] — General purpose digital input/output pin.
			I	PCAP1[0] — Capture input for PWM1, channel 0.
			I	RI1 — Ring Indicator input for UART1.
			O	TRACECLK — Trace Clock.
P2[7]/RD2/RTS1	66 ^[1]	E9 ^[1]	I/O	P2[7] — General purpose digital input/output pin.
			I	RD2 — CAN2 receiver input. (LPC1769/68/66/65/64 only).
			O	RTS1 — Request to Send output for UART1. Can also be configured to be an RS-485/EIA-485 output enable signal.
P2[8]/TD2/TXD2	65 ^[1]	E10 ^[1]	I/O	P2[8] — General purpose digital input/output pin.
			O	TD2 — CAN2 transmitter output. (LPC1769/68/66/65/64 only).
			O	TXD2 — Transmitter output for UART2.
P2[9]/USB_CONNECT/RXD2	64 ^[1]	F7 ^[1]	I/O	P2[9] — General purpose digital input/output pin.
			O	USB_CONNECT — Signal used to switch an external 1.5 kΩ resistor under software control. Used with the SoftConnect USB feature. (LPC1769/68/66/65/64 only).
			I	RXD2 — Receiver input for UART2.

Table 4. Pin description ...continued

Symbol	Pin	Ball	Type	Description
P2[10]/EINT0/NMI	53 ^[6]	J10 ^[6]	I/O	P2[10] — General purpose digital input/output pin. A LOW level on this pin during reset starts the ISP command handler.
			I	EINT0 — External interrupt 0 input.
			I	NMI — Non-maskable interrupt input.
P2[11]/EINT1/I2STX_CLK	52 ^[6]	H8 ^[6]	I/O	P2[11] — General purpose digital input/output pin.
			I	EINT1 — External interrupt 1 input.
			I/O	I2STX_CLK — Transmit Clock. It is driven by the master and received by the slave. Corresponds to the signal SCK in the <i>I²S-bus specification</i> . (LPC1769/68/67/66/65/63 only).
P2[12]/EINT2/I2STX_WS	51 ^[6]	K10 ^[6]	I/O	P2[12] — General purpose digital input/output pin.
			I	EINT2 — External interrupt 2 input.
			I/O	I2STX_WS — Transmit Word Select. It is driven by the master and received by the slave. Corresponds to the signal WS in the <i>I²S-bus specification</i> . (LPC1769/68/67/66/65/63 only).
P2[13]/EINT3/I2STX_SDA	50 ^[6]	J9 ^[6]	I/O	P2[13] — General purpose digital input/output pin.
			I	EINT3 — External interrupt 3 input.
			I/O	I2STX_SDA — Transmit data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the <i>I²S-bus specification</i> . (LPC1769/68/67/66/65/63 only).
P3[0] to P3[31]			I/O	Port 3: Port 3 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 3 pins depends upon the pin function selected via the pin connect block. Pins 0 through 24, and 27 through 31 of this port are not available.
P3[25]/MAT0[0]/PWM1[2]	27 ^[1]	H3 ^[1]	I/O	P3[25] — General purpose digital input/output pin.
			O	MAT0[0] — Match output for Timer 0, channel 0.
			O	PWM1[2] — Pulse Width Modulator 1, output 2.
P3[26]/STCLK/MAT0[1]/PWM1[3]	26 ^[1]	K1 ^[1]	I/O	P3[26] — General purpose digital input/output pin.
			I	STCLK — System tick timer clock input. The maximum STCLK frequency is 1/4 of the ARM processor clock frequency CCLK.
			O	MAT0[1] — Match output for Timer 0, channel 1.
			O	PWM1[3] — Pulse Width Modulator 1, output 3.
P4[0] to P4[31]			I/O	Port 4: Port 4 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 4 pins depends upon the pin function selected via the pin connect block. Pins 0 through 27, 30, and 31 of this port are not available.
P4[28]/RX_MCLK/MAT2[0]/TXD3	82 ^[1]	C7 ^[1]	I/O	P4[28] — General purpose digital input/output pin.
			I	RX_MCLK — I ² S receive master clock. (LPC1769/68/67/66/65 only).
			O	MAT2[0] — Match output for Timer 2, channel 0.
			O	TXD3 — Transmitter output for UART3.
P4[29]/TX_MCLK/MAT2[1]/RXD3	85 ^[1]	E6 ^[1]	I/O	P4[29] — General purpose digital input/output pin.
			I	TX_MCLK — I ² S transmit master clock. (LPC1769/68/67/66/65 only).
			O	MAT2[1] — Match output for Timer 2, channel 1.
			I	RXD3 — Receiver input for UART3.
TDO/SWO	1 ^{[1][7]}	A1 ^[1]	O	TDO — Test Data out for JTAG interface.
			O	SWO — Serial wire trace output.

Table 4. Pin description ...continued

Symbol	Pin	Ball	Type	Description
TDI	2 ^{[1][8]}	C3 ^{[1][8]}	I	TDI — Test Data in for JTAG interface.
TMS/SWDIO	3 ^{[1][8]}	B1 ^{[1][8]}	I	TMS — Test Mode Select for JTAG interface.
			I/O	SWDIO — Serial wire debug data input/output.
TRST	4 ^{[1][8]}	C2 ^{[1][8]}	I	TRST — Test Reset for JTAG interface.
TCK/SWDCLK	5 ^{[1][7]}	C1 ^{[1][7]}	I	TCK — Test Clock for JTAG interface.
			I	SWDCLK — Serial wire clock.
RTCK	100 ^{[1][7]}	B2 ^{[1][7]}	O	RTCK — JTAG interface control signal.
RSTOUT	14	-	O	RSTOUT — This is a 3.3 V pin. LOW on this pin indicates the microcontroller being in Reset state.
RESET	17 ^[9]	F3 ^[9]	I	External reset input: A LOW-going pulse as short as 50 ns on this pin resets the device, causing I/O ports and peripherals to take on their default states, and processor execution to begin at address 0. TTL with hysteresis, 5 V tolerant.
XTAL1	22 ^{[10][11]}	H2 ^{[10][11]}	I	Input to the oscillator circuit and internal clock generator circuits.
XTAL2	23 ^{[10][11]}	G3 ^{[10][11]}	O	Output from the oscillator amplifier.
RTCX1	16 ^{[10][12]}	F2 ^{[10][11]}	I	Input to the RTC oscillator circuit.
RTCX2	18 ^[10]	G1 ^[10]	O	Output from the RTC oscillator circuit.
V _{SS}	31, 41, 55, 72, 83, 97 ^[10]	B3, B7, C9, G7, J6, K3 ^[10]	I	ground: 0 V reference.
V _{SSA}	11 ^[10]	E1 ^[10]	I	analog ground: 0 V reference. This should nominally be the same voltage as V _{SS} , but should be isolated to minimize noise and error.
V _{DD(3V3)}	28, 54, 71, 96 ^[10]	K2, H9, C10, A3 ^[10]	I	3.3 V supply voltage: This is the power supply voltage for the I/O ports.
V _{DD(REG)(3V3)}	42, 84 ^[10]	H6, A7 ^[10]	I	3.3 V voltage regulator supply voltage: This is the supply voltage for the on-chip voltage regulator only.
V _{DDA}	10 ^[10]	E2 ^[10]	I	analog 3.3 V pad supply voltage: This should be nominally the same voltage as V _{DD(3V3)} but should be isolated to minimize noise and error. This voltage is used to power the ADC and DAC. This pin should be tied to 3.3 V if the ADC and DAC are not used.
VREFP	12 ^[10]	E3 ^[10]	I	ADC positive reference voltage: This should be nominally the same voltage as V _{DDA} but should be isolated to minimize noise and error. Level on this pin is used as a reference for ADC and DAC. This pin should be tied to 3.3 V if the ADC and DAC are not used.
VREFN	15	F1	I	ADC negative reference voltage: This should be nominally the same voltage as V _{SS} but should be isolated to minimize noise and error. Level on this pin is used as a reference for ADC and DAC.
VBAT	19 ^{[10][12]}	G2 ^{[10][12]}	I	RTC pin power supply: 3.3 V on this pin supplies the power to the RTC peripheral.
n.c.	13	D4, E4	-	not connected.

[1] 5 V tolerant pad providing digital I/O functions with TTL levels and hysteresis. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.

[2] 5 V tolerant pad providing digital I/O functions (with TTL levels and hysteresis) and analog input. When configured as a ADC input, digital section of the pad is disabled and the pin is not 5 V tolerant. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.

[3] 5 V tolerant pad providing digital I/O with TTL levels and hysteresis and analog output function. When configured as the DAC output, digital section of the pad is disabled. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.

- [4] Open-drain 5 V tolerant digital I/O pad, compatible with I²C-bus 400 kHz specification. This pad requires an external pull-up to provide output functionality. When power is switched off, this pin connected to the I²C-bus is floating and does not disturb the I²C lines. Open-drain configuration applies to all functions on this pin.
- [5] Pad provides digital I/O and USB functions. It is designed in accordance with the *USB specification, revision 2.0* (Full-speed and Low-speed mode only). This pad is not 5 V tolerant.
- [6] 5 V tolerant pad with 10 ns glitch filter providing digital I/O functions with TTL levels and hysteresis. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.
- [7] 5 V tolerant pad with TTL levels and hysteresis. Internal pull-up and pull-down resistors disabled.
- [8] 5 V tolerant pad with TTL levels and hysteresis and internal pull-up resistor.
- [9] 5 V tolerant pad with 20 ns glitch filter providing digital I/O function with TTL levels and hysteresis.
- [10] Pad provides special analog functionality. A 32 kHz crystal oscillator must be used with the RTC.
- [11] When the system oscillator is not used, connect XTAL1 and XTAL2 as follows: XTAL1 can be left floating or can be grounded (grounding is preferred to reduce susceptibility to noise). XTAL2 should be left floating.
- [12] When the RTC is not used, connect VBAT to V_{DD(REG)(3V3)} and leave RTCX1 floating.

7. Functional description

7.1 Architectural overview

Remark: In the following, the notation LPC17xx refers to all parts: LPC1769/68/67/66/65/64/63.

The ARM Cortex-M3 includes three AHB-Lite buses: the system bus, the I-code bus, and the D-code bus (see [Figure 1](#)). The I-code and D-code core buses are faster than the system bus and are used similarly to TCM interfaces: one bus dedicated for instruction fetch (I-code) and one bus for data access (D-code). The use of two core buses allows for simultaneous operations if concurrent operations target different devices.

The LPC17xx use a multi-layer AHB matrix to connect the ARM Cortex-M3 buses and other bus masters to peripherals in a flexible manner that optimizes performance by allowing peripherals that are on different slaves ports of the matrix to be accessed simultaneously by different bus masters.

7.2 ARM Cortex-M3 processor

The ARM Cortex-M3 is a general purpose, 32-bit microprocessor, which offers high performance and very low power consumption. The ARM Cortex-M3 offers many new features, including a Thumb-2 instruction set, low interrupt latency, hardware divide, interruptible/continuable multiple load and store instructions, automatic state save and restore for interrupts, tightly integrated interrupt controller with wake-up interrupt controller, and multiple core buses capable of simultaneous accesses.

Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory.

The ARM Cortex-M3 processor is described in detail in the *Cortex-M3 Technical Reference Manual* that can be found on official ARM website.

7.3 On-chip flash program memory

The LPC17xx contain up to 512 kB of on-chip flash memory. A new two-port flash accelerator maximizes performance for use with the two fast AHB-Lite buses.

7.4 On-chip SRAM

The LPC17xx contain a total of 64 kB on-chip static RAM memory. This includes the main 32 kB SRAM, accessible by the CPU and DMA controller on a higher-speed bus, and two additional 16 kB each SRAM blocks situated on a separate slave port on the AHB multilayer matrix.

This architecture allows CPU and DMA accesses to be spread over three separate RAMs that can be accessed simultaneously.

7.5 Memory Protection Unit (MPU)

The LPC17xx have a Memory Protection Unit (MPU) which can be used to improve the reliability of an embedded system by protecting critical data within the user application.

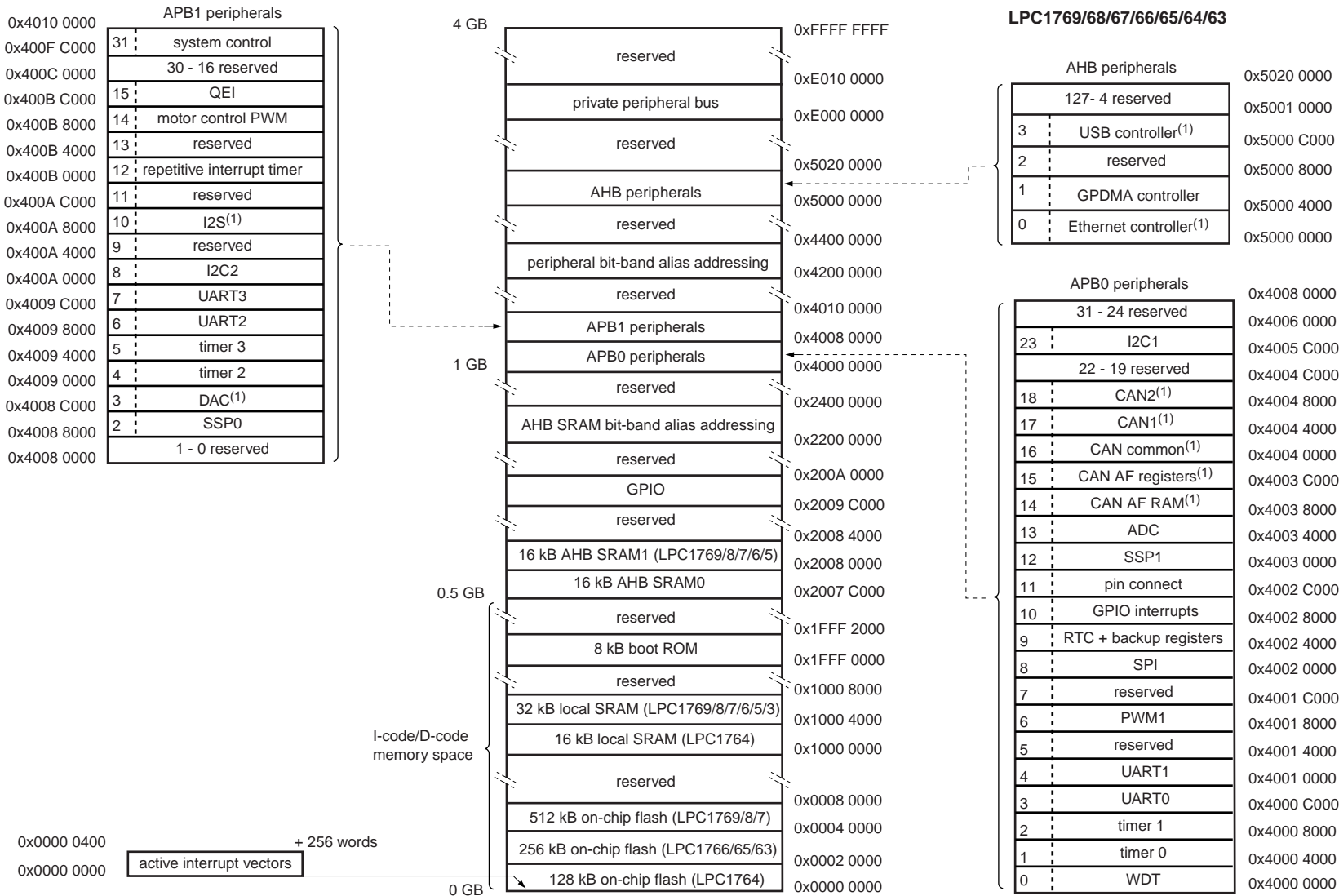
The MPU allows separating processing tasks by disallowing access to each other's data, disabling access to memory regions, allowing memory regions to be defined as read-only and detecting unexpected memory accesses that could potentially break the system.

The MPU separates the memory into distinct regions and implements protection by preventing disallowed accesses. The MPU supports up to 8 regions each of which can be divided into 8 subregions. Accesses to memory locations that are not defined in the MPU regions, or not permitted by the region setting, will cause the Memory Management Fault exception to take place.

7.6 Memory map

The LPC17xx incorporates several distinct memory regions, shown in the following figures. [Figure 4](#) shows the overall map of the entire address space from the user program viewpoint following reset. The interrupt vector area supports address remapping.

The AHB peripheral area is 2 MB in size and is divided to allow for up to 128 peripherals. The APB peripheral area is 1 MB in size and is divided to allow for up to 64 peripherals. Each peripheral of either type is allocated 16 kB of space. This allows simplifying the address decoding for each peripheral.



002aad946

(1) Not available on all parts. See [Table 2](#).

Fig 4. LPC17xx memory map

7.7 Nested Vectored Interrupt Controller (NVIC)

The NVIC is an integral part of the Cortex-M3. The tight coupling to the CPU allows for low interrupt latency and efficient processing of late arriving interrupts.

7.7.1 Features

- Controls system exceptions and peripheral interrupts
- In the LPC17xx, the NVIC supports 33 vectored interrupts
- 32 programmable interrupt priority levels, with hardware priority level masking
- Relocatable vector table
- Non-Maskable Interrupt (NMI)
- Software interrupt generation

7.7.2 Interrupt sources

Each peripheral device has one interrupt line connected to the NVIC but may have several interrupt flags. Individual interrupt flags may also represent more than one interrupt source.

Any pin on Port 0 and Port 2 (total of 42 pins) regardless of the selected function, can be programmed to generate an interrupt on a rising edge, a falling edge, or both.

7.8 Pin connect block

The pin connect block allows selected pins of the microcontroller to have more than one function. Configuration registers control the multiplexers to allow connection between the pin and the on-chip peripherals.

Peripherals should be connected to the appropriate pins prior to being activated and prior to any related interrupt(s) being enabled. Activity of any enabled peripheral function that is not mapped to a related pin should be considered undefined.

Most pins can also be configured as open-drain outputs or to have a pull-up, pull-down, or no resistor enabled.

7.9 General purpose DMA controller

The GPDMA is an AMBA AHB compliant peripheral allowing selected peripherals to have DMA support.

The GPDMA enables peripheral-to-memory, memory-to-peripheral, peripheral-to-peripheral, and memory-to-memory transactions. The source and destination areas can each be either a memory region or a peripheral, and can be accessed through the AHB master. The GPDMA controller allows data transfers between the USB and Ethernet controllers and the various on-chip SRAM areas. The supported APB peripherals are SSP0/1, all UARTs, the I²S-bus interface, the ADC, and the DAC. Two match signals for each timer can be used to trigger DMA transfers.

Remark: The Ethernet controller is available on parts LPC1769/68/67/66/64. The USB controller is available on parts LPC1769/68/66/65/64. The I²S-bus interface is available on parts LPC1769/68/67/66/65. The DAC is available on parts LPC1769/68/67/66/65/63.

7.9.1 Features

- Eight DMA channels. Each channel can support an unidirectional transfer.
- 16 DMA request lines.
- Single DMA and burst DMA request signals. Each peripheral connected to the DMA Controller can assert either a burst DMA request or a single DMA request. The DMA burst size is set by programming the DMA Controller.
- Memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral transfers are supported.
- Scatter or gather DMA is supported through the use of linked lists. This means that the source and destination areas do not have to occupy contiguous areas of memory.
- Hardware DMA channel priority.
- AHB slave DMA programming interface. The DMA Controller is programmed by writing to the DMA control registers over the AHB slave interface.
- One AHB bus master for transferring data. The interface transfers data when a DMA request goes active.
- 32-bit AHB master bus width.
- Incrementing or non-incrementing addressing for source and destination.
- Programmable DMA burst size. The DMA burst size can be programmed to more efficiently transfer data.
- Internal four-word FIFO per channel.
- Supports 8, 16, and 32-bit wide transactions.
- Big-endian and little-endian support. The DMA Controller defaults to little-endian mode on reset.
- An interrupt to the processor can be generated on a DMA completion or when a DMA error has occurred.
- Raw interrupt status. The DMA error and DMA count raw interrupt status can be read prior to masking.

7.10 Fast general purpose parallel I/O

Device pins that are not connected to a specific peripheral function are controlled by the GPIO registers. Pins may be dynamically configured as inputs or outputs. Separate registers allow setting or clearing any number of outputs simultaneously. The value of the output register may be read back as well as the current state of the port pins.

LPC17xx use accelerated GPIO functions:

- GPIO registers are accessed through the AHB multilayer bus so that the fastest possible I/O timing can be achieved.
- Mask registers allow treating sets of port bits as a group, leaving other bits unchanged.
- All GPIO registers are byte and half-word addressable.
- Entire port value can be written in one instruction.
- Support for Cortex-M3 bit banding.
- Support for use with the GPDMA controller.

Additionally, any pin on Port 0 and Port 2 (total of 42 pins) providing a digital function can be programmed to generate an interrupt on a rising edge, a falling edge, or both. The edge detection is asynchronous, so it may operate when clocks are not present such as during Power-down mode. Each enabled interrupt can be used to wake up the chip from Power-down mode.

7.10.1 Features

- Bit level set and clear registers allow a single instruction to set or clear any number of bits in one port.
- Direction control of individual bits.
- All I/O default to inputs after reset.
- Pull-up/pull-down resistor configuration and open-drain configuration can be programmed through the pin connect block for each GPIO pin.

7.11 Ethernet

Remark: The Ethernet controller is available on parts LPC1769/68/67/66/64. The Ethernet block supports bus clock rates of up to 100 MHz (LPC1768/67/66/64) or 120 MHz (LPC1769). See [Table 2](#).

The Ethernet block contains a full featured 10 Mbit/s or 100 Mbit/s Ethernet MAC designed to provide optimized performance through the use of DMA hardware acceleration. Features include a generous suite of control registers, half or full duplex operation, flow control, control frames, hardware acceleration for transmit retry, receive packet filtering and wake-up on LAN activity. Automatic frame transmission and reception with scatter-gather DMA off-loads many operations from the CPU.

The Ethernet block and the CPU share the ARM Cortex-M3 D-code and system bus through the AHB-multilayer matrix to access the various on-chip SRAM blocks for Ethernet data, control, and status information.

The Ethernet block interfaces between an off-chip Ethernet PHY using the Reduced MII (RMII) protocol and the on-chip Media Independent Interface Management (MIIM) serial bus.

7.11.1 Features

- Ethernet standards support:
 - Supports 10 Mbit/s or 100 Mbit/s PHY devices including 10 Base-T, 100 Base-TX, 100 Base-FX, and 100 Base-T4.
 - Fully compliant with *IEEE standard 802.3*.
 - Fully compliant with 802.3x full duplex flow control and half duplex back pressure.
 - Flexible transmit and receive frame options.
 - Virtual Local Area Network (VLAN) frame support.
- Memory management:
 - Independent transmit and receive buffers memory mapped to shared SRAM.
 - DMA managers with scatter/gather DMA and arrays of frame descriptors.
 - Memory traffic optimized by buffering and pre-fetching.

- Enhanced Ethernet features:
 - Receive filtering.
 - Multicast and broadcast frame support for both transmit and receive.
 - Optional automatic Frame Check Sequence (FCS) insertion with Cyclic Redundancy Check (CRC) for transmit.
 - Selectable automatic transmit frame padding.
 - Over-length frame support for both transmit and receive allows any length frames.
 - Promiscuous receive mode.
 - Automatic collision back-off and frame retransmission.
 - Includes power management by clock switching.
 - Wake-on-LAN power management support allows system wake-up: using the receive filters or a magic frame detection filter.
- Physical interface:
 - Attachment of external PHY chip through standard RMII interface.
 - PHY register access is available via the MIIM interface.

7.12 USB interface

Remark: The USB controller is available as device/Host/OTG controller on parts LPC1769/68/66/65 and as device-only controller on part LPC1764.

The Universal Serial Bus (USB) is a 4-wire bus that supports communication between a host and one or more (up to 127) peripherals. The host controller allocates the USB bandwidth to attached devices through a token-based protocol. The bus supports hot plugging and dynamic configuration of the devices. All transactions are initiated by the host controller.

The USB interface includes a device, Host, and OTG controller with on-chip PHY for device and Host functions. The OTG switching protocol is supported through the use of an external controller. Details on typical USB interfacing solutions can be found in [Section 14.1](#).

7.12.1 USB device controller

The device controller enables 12 Mbit/s data exchange with a USB Host controller. It consists of a register interface, serial interface engine, endpoint buffer memory, and a DMA controller. The serial interface engine decodes the USB data stream and writes data to the appropriate endpoint buffer. The status of a completed USB transfer or error condition is indicated via status registers. An interrupt is also generated if enabled. When enabled, the DMA controller transfers data between the endpoint buffer and the on-chip SRAM.

7.12.1.1 Features

- Fully compliant with *USB 2.0 specification (full speed)*.
- Supports 32 physical (16 logical) endpoints with a 4 kB endpoint buffer RAM.
- Supports Control, Bulk, Interrupt and Isochronous endpoints.
- Scalable realization of endpoints at run time.

- Endpoint Maximum packet size selection (up to USB maximum specification) by software at run time.
- Supports SoftConnect and GoodLink features.
- While USB is in the Suspend mode, the part can enter one of the reduced power modes and wake up on USB activity.
- Supports DMA transfers with all on-chip SRAM blocks on all non-control endpoints.
- Allows dynamic switching between CPU-controlled slave and DMA modes.
- Double buffer implementation for Bulk and Isochronous endpoints.

7.12.2 USB host controller

The host controller enables full- and low-speed data exchange with USB devices attached to the bus. It consists of a register interface, a serial interface engine, and a DMA controller. The register interface complies with the OHCI specification.

7.12.2.1 Features

- OHCI compliant.
- One downstream port.
- Supports port power switching.

7.12.3 USB OTG controller

USB OTG is a supplement to the USB 2.0 specification that augments the capability of existing mobile devices and USB peripherals by adding host functionality for connection to USB peripherals.

The OTG Controller integrates the host controller, device controller, and a master-only I²C-bus interface to implement OTG dual-role device functionality. The dedicated I²C-bus interface controls an external OTG transceiver.

7.12.3.1 Features

- Fully compliant with *On-The-Go supplement to the USB 2.0 Specification, Revision 1.0a*.
- Hardware support for Host Negotiation Protocol (HNP).
- Includes a programmable timer required for HNP and Session Request Protocol (SRP).
- Supports any OTG transceiver compliant with the *OTG Transceiver Specification (CEA-2011), Rev. 1.0*.

7.13 CAN controller and acceptance filters

Remark: The CAN controllers are available on parts LPC1769/68/66/65/64. See [Table 2](#).

The Controller Area Network (CAN) is a serial communications protocol which efficiently supports distributed real-time control with a very high level of security. Its domain of application ranges from high-speed networks to low cost multiplex wiring.

The CAN block is intended to support multiple CAN buses simultaneously, allowing the device to be used as a gateway, switch, or router among a number of CAN buses in industrial or automotive applications.

7.13.1 Features

- Two CAN controllers and buses.
- Data rates to 1 Mbit/s on each bus.
- 32-bit register and RAM access.
- Compatible with *CAN specification 2.0B, ISO 11898-1*.
- Global Acceptance Filter recognizes standard (11-bit) and extended-frame (29-bit) receive identifiers for all CAN buses.
- Acceptance Filter can provide FullCAN-style automatic reception for selected Standard Identifiers.
- FullCAN messages can generate interrupts.

7.14 12-bit ADC

The LPC17xx contain a single 12-bit successive approximation ADC with eight channels and DMA support.

7.14.1 Features

- 12-bit successive approximation ADC.
- Input multiplexing among 8 pins.
- Power-down mode.
- Measurement range VREFN to VREFP.
- 12-bit conversion rate: 200 kHz.
- Individual channels can be selected for conversion.
- Burst conversion mode for single or multiple inputs.
- Optional conversion on transition of input pin or Timer Match signal.
- Individual result registers for each ADC channel to reduce interrupt overhead.
- DMA support.

7.15 10-bit DAC

The DAC allows to generate a variable analog output. The maximum output value of the DAC is VREFP.

Remark: The DAC is available on parts LPC1769/68/67/66/65/63. See [Table 2](#).

7.15.1 Features

- 10-bit DAC
- Resistor string architecture
- Buffered output
- Power-down mode
- Selectable output drive
- Dedicated conversion timer
- DMA support

7.16 UARTs

The LPC17xx each contain four UARTs. In addition to standard transmit and receive data lines, UART1 also provides a full modem control handshake interface and support for RS-485/9-bit mode allowing both software address detection and automatic address detection using 9-bit mode.

The UARTs include a fractional baud rate generator. Standard baud rates such as 115200 Bd can be achieved with any crystal frequency above 2 MHz.

7.16.1 Features

- Maximum UART data bit rate of 6.25 Mbit/s.
- 16 B Receive and Transmit FIFOs.
- Register locations conform to 16C550 industry standard.
- Receiver FIFO trigger points at 1 B, 4 B, 8 B, and 14 B.
- Built-in fractional baud rate generator covering wide range of baud rates without a need for external crystals of particular values.
- Auto baud capabilities and FIFO control mechanism that enables software flow control implementation.
- UART1 equipped with standard modem interface signals. This module also provides full support for hardware flow control (auto-CTS/RTS).
- Support for RS-485/9-bit/EIA-485 mode (UART1).
- UART3 includes an IrDA mode to support infrared communication.
- All UARTs have DMA support.

7.17 SPI serial I/O controller

The LPC17xx contain one SPI controller. SPI is a full duplex serial interface designed to handle multiple masters and slaves connected to a given bus. Only a single master and a single slave can communicate on the interface during a given data transfer. During a data transfer the master always sends 8 bits to 16 bits of data to the slave, and the slave always sends 8 bits to 16 bits of data to the master.

7.17.1 Features

- Maximum SPI data bit rate of 12.5 Mbit/s
- Compliant with SPI specification
- Synchronous, serial, full duplex communication
- Combined SPI master and slave
- Maximum data bit rate of one eighth of the input clock rate
- 8 bits to 16 bits per transfer

7.18 SSP serial I/O controller

The LPC17xx contain two SSP controllers. The SSP controller is capable of operation on a SPI, 4-wire SSI, or Microwire bus. It can interact with multiple masters and slaves on the bus. Only a single master and a single slave can communicate on the bus during a given

data transfer. The SSP supports full duplex transfers, with frames of 4 bits to 16 bits of data flowing from the master to the slave and from the slave to the master. In practice, often only one of these data flows carries meaningful data.

7.18.1 Features

- Maximum SSP speed of 50 Mbit/s (master) or 8 Mbit/s (slave)
- Compatible with Motorola SPI, 4-wire Texas Instruments SSI, and National Semiconductor Microwire buses
- Synchronous serial communication
- Master or slave operation
- 8-frame FIFOs for both transmit and receive
- 4-bit to 16-bit frame
- DMA transfers supported by GPDMA

7.19 I²C-bus serial I/O controllers

The LPC17xx each contain three I²C-bus controllers.

The I²C-bus is bidirectional for inter-IC control using only two wires: a Serial Clock line (SCL) and a Serial Data line (SDA). Each device is recognized by a unique address and can operate as either a receiver-only device (e.g., an LCD driver) or a transmitter with the capability to both receive and send information (such as memory). Transmitters and/or receivers can operate in either master or slave mode, depending on whether the chip has to initiate a data transfer or is only addressed. The I²C is a multi-master bus and can be controlled by more than one bus master connected to it.

7.19.1 Features

- I²C0 is a standard I²C compliant bus interface with open-drain pins. I²C0 also supports Fast mode plus with bit rates up to 1 Mbit/s.
- I²C1 and I²C2 use standard I/O pins with bit rates of up to 400 kbit/s (Fast I²C-bus).
- Easy to configure as master, slave, or master/slave.
- Programmable clocks allow versatile rate control.
- Bidirectional data transfer between masters and slaves.
- Multi-master bus (no central master).
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus.
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus.
- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer.
- The I²C-bus can be used for test and diagnostic purposes.
- All I²C-bus controllers support multiple address recognition and a bus monitor mode.

7.20 I²S-bus serial I/O controllers

Remark: The I²S-bus interface is available on parts LPC1769/68/67/66/65/63. See [Table 2](#).

The I²S-bus provides a standard communication interface for digital audio applications.

The *I²S-bus specification* defines a 3-wire serial bus using one data line, one clock line, and one word select signal. The basic I²S-bus connection has one master, which is always the master, and one slave. The I²S-bus interface provides a separate transmit and receive channel, each of which can operate as either a master or a slave.

7.20.1 Features

- The interface has separate input/output channels each of which can operate in master or slave mode.
- Capable of handling 8-bit, 16-bit, and 32-bit word sizes.
- Mono and stereo audio data supported.
- The sampling frequency can range from 16 kHz to 96 kHz (16, 22.05, 32, 44.1, 48, 96) kHz.
- Support for an audio master clock.
- Configurable word select period in master mode (separately for I²S-bus input and output).
- Two 8-word FIFO data buffers are provided, one for transmit and one for receive.
- Generates interrupt requests when buffer levels cross a programmable boundary.
- Two DMA requests, controlled by programmable buffer levels. These are connected to the GPDMA block.
- Controls include reset, stop and mute options separately for I²S-bus input and I²S-bus output.

7.21 General purpose 32-bit timers/external event counters

The LPC17xx include four 32-bit timer/counters. The timer/counter is designed to count cycles of the system derived clock or an externally-supplied clock. It can optionally generate interrupts, generate timed DMA requests, or perform other actions at specified timer values, based on four match registers. Each timer/counter also includes two capture inputs to trap the timer value when an input signal transitions, optionally generating an interrupt.

7.21.1 Features

- A 32-bit timer/counter with a programmable 32-bit prescaler.
- Counter or timer operation.
- Two 32-bit capture channels per timer, that can take a snapshot of the timer value when an input signal transitions. A capture event may also generate an interrupt.
- Four 32-bit match registers that allow:
 - Continuous operation with optional interrupt generation on match.
 - Stop timer on match with optional interrupt generation.
 - Reset timer on match with optional interrupt generation.

- Up to four external outputs corresponding to match registers, with the following capabilities:
 - Set LOW on match.
 - Set HIGH on match.
 - Toggle on match.
 - Do nothing on match.
- Up to two match registers can be used to generate timed DMA requests.

7.22 Pulse width modulator

The PWM is based on the standard Timer block and inherits all of its features, although only the PWM function is pinned out on the LPC17xx. The Timer is designed to count cycles of the system derived clock and optionally switch pins, generate interrupts or perform other actions when specified timer values occur, based on seven match registers. The PWM function is in addition to these features, and is based on match register events.

The ability to separately control rising and falling edge locations allows the PWM to be used for more applications. For instance, multi-phase motor control typically requires three non-overlapping PWM outputs with individual control of all three pulse widths and positions.

Two match registers can be used to provide a single edge controlled PWM output. One match register (PWMMR0) controls the PWM cycle rate, by resetting the count upon match. The other match register controls the PWM edge position. Additional single edge controlled PWM outputs require only one match register each, since the repetition rate is the same for all PWM outputs. Multiple single edge controlled PWM outputs will all have a rising edge at the beginning of each PWM cycle, when an PWMMR0 match occurs.

Three match registers can be used to provide a PWM output with both edges controlled. Again, the PWMMR0 match register controls the PWM cycle rate. The other match registers control the two PWM edge positions. Additional double edge controlled PWM outputs require only two match registers each, since the repetition rate is the same for all PWM outputs.

With double edge controlled PWM outputs, specific match registers control the rising and falling edge of the output. This allows both positive going PWM pulses (when the rising edge occurs prior to the falling edge), and negative going PWM pulses (when the falling edge occurs prior to the rising edge).

7.22.1 Features

- One PWM block with Counter or Timer operation (may use the peripheral clock or one of the capture inputs as the clock source).
- Seven match registers allow up to 6 single edge controlled or 3 double edge controlled PWM outputs, or a mix of both types. The match registers also allow:
 - Continuous operation with optional interrupt generation on match.
 - Stop timer on match with optional interrupt generation.
 - Reset timer on match with optional interrupt generation.

- Supports single edge controlled and/or double edge controlled PWM outputs. Single edge controlled PWM outputs all go high at the beginning of each cycle unless the output is a constant low. Double edge controlled PWM outputs can have either edge occur at any position within a cycle. This allows for both positive going and negative going pulses.
- Pulse period and width can be any number of timer counts. This allows complete flexibility in the trade-off between resolution and repetition rate. All PWM outputs will occur at the same repetition rate.
- Double edge controlled PWM outputs can be programmed to be either positive going or negative going pulses.
- Match register updates are synchronized with pulse outputs to prevent generation of erroneous pulses. Software must 'release' new match values before they can become effective.
- May be used as a standard 32-bit timer/counter with a programmable 32-bit prescaler if the PWM mode is not enabled.

7.23 Motor control PWM

The motor control PWM is a specialized PWM supporting 3-phase motors and other combinations. Feedback inputs are provided to automatically sense rotor position and use that information to ramp speed up or down. An abort input is also provided that causes the PWM to immediately release all motor drive outputs. At the same time, the motor control PWM is highly configurable for other generalized timing, counting, capture, and compare applications.

7.24 Quadrature Encoder Interface (QEI)

A quadrature encoder, also known as a 2-channel incremental encoder, converts angular displacement into two pulse signals. By monitoring both the number of pulses and the relative phase of the two signals, the user can track the position, direction of rotation, and velocity. In addition, a third channel, or index signal, can be used to reset the position counter. The quadrature encoder interface decodes the digital pulses from a quadrature encoder wheel to integrate position over time and determine direction of rotation. In addition, the QEI can capture the velocity of the encoder wheel.

7.24.1 Features

- Tracks encoder position.
- Increments/decrements depending on direction.
- Programmable for 2× or 4× position counting.
- Velocity capture using built-in timer.
- Velocity compare function with "less than" interrupt.
- Uses 32-bit registers for position and velocity.
- Three position compare registers with interrupts.
- Index counter for revolution counting.
- Index compare register with interrupts.
- Can combine index and position interrupts to produce an interrupt for whole and partial revolution displacement.

- Digital filter with programmable delays for encoder input signals.
- Can accept decoded signal inputs (clk and direction).
- Connected to APB.

7.25 Repetitive Interrupt (RI) timer

The repetitive interrupt timer provides a free-running 32-bit counter which is compared to a selectable value, generating an interrupt when a match occurs. Any bits of the timer/compare can be masked such that they do not contribute to the match detection. The repetitive interrupt timer can be used to create an interrupt that repeats at predetermined intervals.

7.25.1 Features

- 32-bit counter running from PCLK. Counter can be free-running or be reset by a generated interrupt.
- 32-bit compare value.
- 32-bit compare mask. An interrupt is generated when the counter value equals the compare value, after masking. This allows for combinations not possible with a simple compare.

7.26 ARM Cortex-M3 system tick timer

The ARM Cortex-M3 includes a system tick timer (SYSTICK) that is intended to generate a dedicated SYSTICK exception at a 10 ms interval. In the LPC17xx, this timer can be clocked from the internal AHB clock or from a device pin.

7.27 Watchdog timer

The purpose of the watchdog is to reset the microcontroller within a reasonable amount of time if it enters an erroneous state. When enabled, the watchdog will generate a system reset if the user program fails to 'feed' (or reload) the watchdog within a predetermined amount of time.

7.27.1 Features

- Internally resets chip if not periodically reloaded.
- Debug mode.
- Enabled by software but requires a hardware reset or a watchdog reset/interrupt to be disabled.
- Incorrect/Incomplete feed sequence causes reset/interrupt if enabled.
- Flag to indicate watchdog reset.
- Programmable 32-bit timer with internal prescaler.
- Selectable time period from $(T_{cy(WDCLK)} \times 256 \times 4)$ to $(T_{cy(WDCLK)} \times 2^{32} \times 4)$ in multiples of $T_{cy(WDCLK)} \times 4$.
- The Watchdog Clock (WDCLK) source can be selected from the Internal RC (IRC) oscillator, the RTC oscillator, or the APB peripheral clock. This gives a wide range of potential timing choices of Watchdog operation under different power reduction

conditions. It also provides the ability to run the WDT from an entirely internal source that is not dependent on an external crystal and its associated components and wiring for increased reliability.

- Includes lock/safe feature.

7.28 RTC and backup registers

The RTC is a set of counters for measuring time when system power is on, and optionally when it is off. The RTC on the LPC17xx is designed to have extremely low power consumption, i.e. less than 1 μ A. The RTC will typically run from the main chip power supply, conserving battery power while the rest of the device is powered up. When operating from a battery, the RTC will continue working down to 2.1 V. Battery power can be provided from a standard 3 V Lithium button cell.

An ultra-low power 32 kHz oscillator will provide a 1 Hz clock to the time counting portion of the RTC, moving most of the power consumption out of the time counting function.

The RTC includes a calibration mechanism to allow fine-tuning the count rate in a way that will provide less than 1 second per day error when operated at a constant voltage and temperature. A clock output function (see [Section 7.29.4](#)) makes measuring the oscillator rate easy and accurate.

The RTC contains a small set of backup registers (20 bytes) for holding data while the main part of the LPC17xx is powered off.

The RTC includes an alarm function that can wake up the LPC17xx from all reduced power modes with a time resolution of 1 s.

7.28.1 Features

- Measures the passage of time to maintain a calendar and clock.
- Ultra low power design to support battery powered systems.
- Provides Seconds, Minutes, Hours, Day of Month, Month, Year, Day of Week, and Day of Year.
- Dedicated power supply pin can be connected to a battery or to the main 3.3 V.
- Periodic interrupts can be generated from increments of any field of the time registers.
- Backup registers (20 bytes) powered by VBAT.
- RTC power supply is isolated from the rest of the chip.

7.29 Clocking and power control

7.29.1 Crystal oscillators

The LPC17xx include three independent oscillators. These are the main oscillator, the IRC oscillator, and the RTC oscillator. Each oscillator can be used for more than one purpose as required in a particular application. Any of the three clock sources can be chosen by software to drive the main PLL and ultimately the CPU.

Following reset, the LPC17xx will operate from the Internal RC oscillator until switched by software. This allows systems to operate without any external crystal and the bootloader code to operate at a known frequency.

See [Figure 5](#) for an overview of the LPC17xx clock generation.

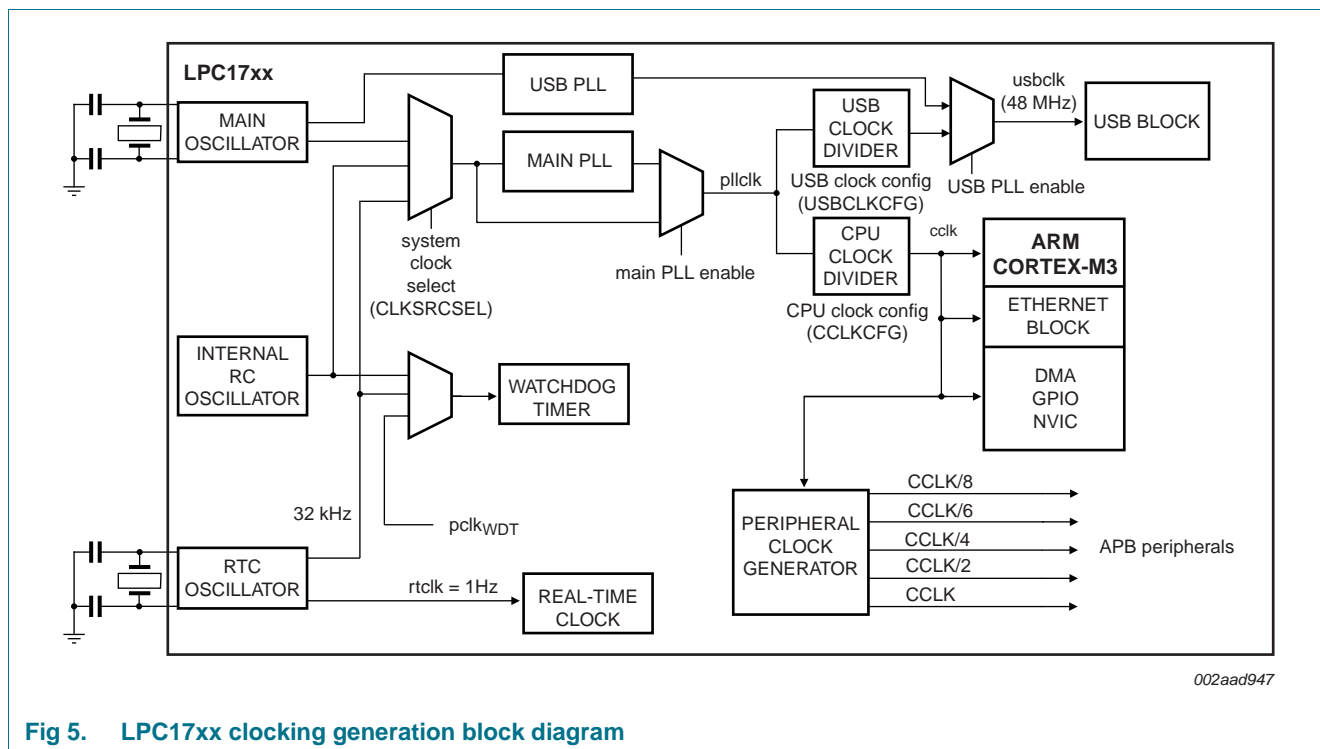


Fig 5. LPC17xx clocking generation block diagram

7.29.1.1 Internal RC oscillator

The IRC may be used as the clock source for the WDT, and/or as the clock that drives the PLL and subsequently the CPU. The nominal IRC frequency is 4 MHz. The IRC is trimmed to 1 % accuracy over the entire voltage and temperature range.

Upon power-up or any chip reset, the LPC17xx use the IRC as the clock source. Software may later switch to one of the other available clock sources.

7.29.1.2 Main oscillator

The main oscillator can be used as the clock source for the CPU, with or without using the PLL. The main oscillator also provides the clock source for the dedicated USB PLL.

The main oscillator operates at frequencies of 1 MHz to 25 MHz. This frequency can be boosted to a higher frequency, up to the maximum CPU operating frequency, by the main PLL. The clock selected as the PLL input is PLLCLKIN. The ARM processor clock frequency is referred to as CCLK elsewhere in this document. The frequencies of PLLCLKIN and CCLK are the same value unless the PLL is active and connected. The clock frequency for each peripheral can be selected individually and is referred to as PCLK. Refer to [Section 7.29.2](#) for additional information.

7.29.1.3 RTC oscillator

The RTC oscillator can be used as the clock source for the RTC block, the main PLL, and/or the CPU.

7.29.2 Main PLL (PLL0)

The PLL0 accepts an input clock frequency in the range of 32 kHz to 25 MHz. The input frequency is multiplied up to a high frequency, then divided down to provide the actual clock used by the CPU and/or the USB block.

The PLL0 input, in the range of 32 kHz to 25 MHz, may initially be divided down by a value 'N', which may be in the range of 1 to 256. This input division provides a wide range of output frequencies from the same input frequency.

Following the PLL0 input divider is the PLL0 multiplier. This can multiply the input divider output through the use of a Current Controlled Oscillator (CCO) by a value 'M', in the range of 1 through 32768. The resulting frequency must be in the range of 275 MHz to 550 MHz. The multiplier works by dividing the CCO output by the value of M, then using a phase-frequency detector to compare the divided CCO output to the multiplier input. The error value is used to adjust the CCO frequency.

The PLL0 is turned off and bypassed following a chip Reset and by entering Power-down mode. PLL0 is enabled by software only. The program must configure and activate the PLL0, wait for the PLL0 to lock, and then connect to the PLL0 as a clock source.

7.29.3 USB PLL (PLL1)

The LPC17xx contain a second, dedicated USB PLL1 to provide clocking for the USB interface.

The PLL1 receives its clock input from the main oscillator only and provides a fixed 48 MHz clock to the USB block only. The PLL1 is disabled and powered off on reset. If the PLL1 is left disabled, the USB clock will be supplied by the 48 MHz clock from the main PLL0.

The PLL1 accepts an input clock frequency in the range of 10 MHz to 25 MHz only. The input frequency is multiplied up the range of 48 MHz for the USB clock using a Current Controlled Oscillators (CCO). It is insured that the PLL1 output has a 50 % duty cycle.

7.29.4 RTC clock output

The LPC17xx feature a clock output function intended for synchronizing with external devices and for use during system development to allow checking the internal clocks CCLK, IRC clock, main crystal, RTC clock, and USB clock in the outside world. The RTC clock output allows tuning the RTC frequency without probing the pin, which would distort the results.

7.29.5 Wake-up timer

The LPC17xx begin operation at power-up and when awakened from Power-down mode by using the 4 MHz IRC oscillator as the clock source. This allows chip operation to resume quickly. If the main oscillator or the PLL is needed by the application, software will need to enable these features and wait for them to stabilize before they are used as a clock source.

When the main oscillator is initially activated, the wake-up timer allows software to ensure that the main oscillator is fully functional before the processor uses it as a clock source and starts to execute instructions. This is important at power on, all types of Reset, and

whenever any of the aforementioned functions are turned off for any reason. Since the oscillator and other functions are turned off during Power-down mode, any wake-up of the processor from Power-down mode makes use of the wake-up timer.

The Wake-up Timer monitors the crystal oscillator to check whether it is safe to begin code execution. When power is applied to the chip, or when some event caused the chip to exit Power-down mode, some time is required for the oscillator to produce a signal of sufficient amplitude to drive the clock logic. The amount of time depends on many factors, including the rate of $V_{DD(3V3)}$ ramp (in the case of power on), the type of crystal and its electrical characteristics (if a quartz crystal is used), as well as any other external circuitry (e.g., capacitors), and the characteristics of the oscillator itself under the existing ambient conditions.

7.29.6 Power control

The LPC17xx support a variety of power control features. There are four special modes of processor power reduction: Sleep mode, Deep-sleep mode, Power-down mode, and Deep power-down mode. The CPU clock rate may also be controlled as needed by changing clock sources, reconfiguring PLL values, and/or altering the CPU clock divider value. This allows a trade-off of power versus processing speed based on application requirements. In addition, Peripheral Power Control allows shutting down the clocks to individual on-chip peripherals, allowing fine tuning of power consumption by eliminating all dynamic power use in any peripherals that are not required for the application. Each of the peripherals has its own clock divider which provides even better power control.

Integrated PMU (Power Management Unit) automatically adjust internal regulators to minimize power consumption during Sleep, Deep sleep, Power-down, and Deep power-down modes.

The LPC17xx also implement a separate power domain to allow turning off power to the bulk of the device while maintaining operation of the RTC and a small set of registers for storing data during any of the power-down modes.

7.29.6.1 Sleep mode

When Sleep mode is entered, the clock to the core is stopped. Resumption from the Sleep mode does not need any special sequence but re-enabling the clock to the ARM core.

In Sleep mode, execution of instructions is suspended until either a Reset or interrupt occurs. Peripheral functions continue operation during Sleep mode and may generate interrupts to cause the processor to resume execution. Sleep mode eliminates dynamic power used by the processor itself, memory systems and related controllers, and internal buses.

7.29.6.2 Deep-sleep mode

In Deep-sleep mode, the oscillator is shut down and the chip receives no internal clocks. The processor state and registers, peripheral registers, and internal SRAM values are preserved throughout Deep-sleep mode and the logic levels of chip pins remain static. The output of the IRC is disabled but the IRC is not powered down for a fast wake-up later. The RTC oscillator is not stopped because the RTC interrupts may be used as the wake-up source. The PLL is automatically turned off and disconnected. The CCLK and USB clock dividers automatically get reset to zero.

The Deep-sleep mode can be terminated and normal operation resumed by either a Reset or certain specific interrupts that are able to function without clocks. Since all dynamic operation of the chip is suspended, Deep-sleep mode reduces chip power consumption to a very low value. Power to the flash memory is left on in Deep-sleep mode, allowing a very quick wake-up.

On wake-up from Deep-sleep mode, the code execution and peripherals activities will resume after 4 cycles expire if the IRC was used before entering Deep-sleep mode. If the main external oscillator was used, the code execution will resume when 4096 cycles expire. PLL and clock dividers need to be reconfigured accordingly.

7.29.6.3 Power-down mode

Power-down mode does everything that Deep-sleep mode does, but also turns off the power to the IRC oscillator and the flash memory. This saves more power but requires waiting for resumption of flash operation before execution of code or data access in the flash memory can be accomplished.

On the wake-up of Power-down mode, if the IRC was used before entering Power-down mode, it will take IRC 60 μ s to start-up. After this 4 IRC cycles will expire before the code execution can then be resumed if the code was running from SRAM. In the meantime, the flash wake-up timer then counts 4 MHz IRC clock cycles to make the 100 μ s flash start-up time. When it times out, access to the flash will be allowed. Users need to reconfigure the PLL and clock dividers accordingly.

7.29.6.4 Deep power-down mode

The Deep power-down mode can only be entered from the RTC block. In Deep power-down mode, power is shut off to the entire chip with the exception of the RTC module and the $\overline{\text{RESET}}$ pin.

The LPC17xx can wake up from Deep power-down mode via the $\overline{\text{RESET}}$ pin or an alarm match event of the RTC.

7.29.6.5 Wake-up interrupt controller

The Wake-up Interrupt Controller (WIC) allows the CPU to automatically wake up from any enabled priority interrupt that can occur while the clocks are stopped in Deep sleep, Power-down, and Deep power-down modes.

The WIC works in connection with the Nested Vectored Interrupt Controller (NVIC). When the CPU enters Deep sleep, Power-down, or Deep power-down mode, the NVIC sends a mask of the current interrupt situation to the WIC. This mask includes all of the interrupts that are both enabled and of sufficient priority to be serviced immediately. With this information, the WIC simply notices when one of the interrupts has occurred and then it wakes up the CPU.

The WIC eliminates the need to periodically wake up the CPU and poll the interrupts resulting in additional power savings.

7.29.7 Peripheral power control

A Power Control for Peripherals feature allows individual peripherals to be turned off if they are not needed in the application, resulting in additional power savings.

7.29.8 Power domains

The LPC17xx provide two independent power domains that allow the bulk of the device to have power removed while maintaining operation of the RTC and the backup Registers.

On the LPC17xx, I/O pads are powered by the 3.3 V ($V_{DD(3V3)}$) pins, while the $V_{DD(REG)(3V3)}$ pin powers the on-chip voltage regulator which in turn provides power to the CPU and most of the peripherals.

Depending on the LPC17xx application, a design can use two power options to manage power consumption.

The first option assumes that power consumption is not a concern and the design ties the $V_{DD(3V3)}$ and $V_{DD(REG)(3V3)}$ pins together. This approach requires only one 3.3 V power supply for both pads, the CPU, and peripherals. While this solution is simple, it does not support powering down the I/O pad ring “on the fly” while keeping the CPU and peripherals alive.

The second option uses two power supplies; a 3.3 V supply for the I/O pads ($V_{DD(3V3)}$) and a dedicated 3.3 V supply for the CPU ($V_{DD(REG)(3V3)}$). Having the on-chip voltage regulator powered independently from the I/O pad ring enables shutting down of the I/O pad power supply “on the fly”, while the CPU and peripherals stay active.

The VBAT pin supplies power only to the RTC domain. The RTC requires a minimum of power to operate, which can be supplied by an external battery. The device core power ($V_{DD(REG)(3V3)}$) is used to operate the RTC whenever $V_{DD(REG)(3V3)}$ is present. Therefore, there is no power drain from the RTC battery when $V_{DD(REG)(3V3)}$ is available.

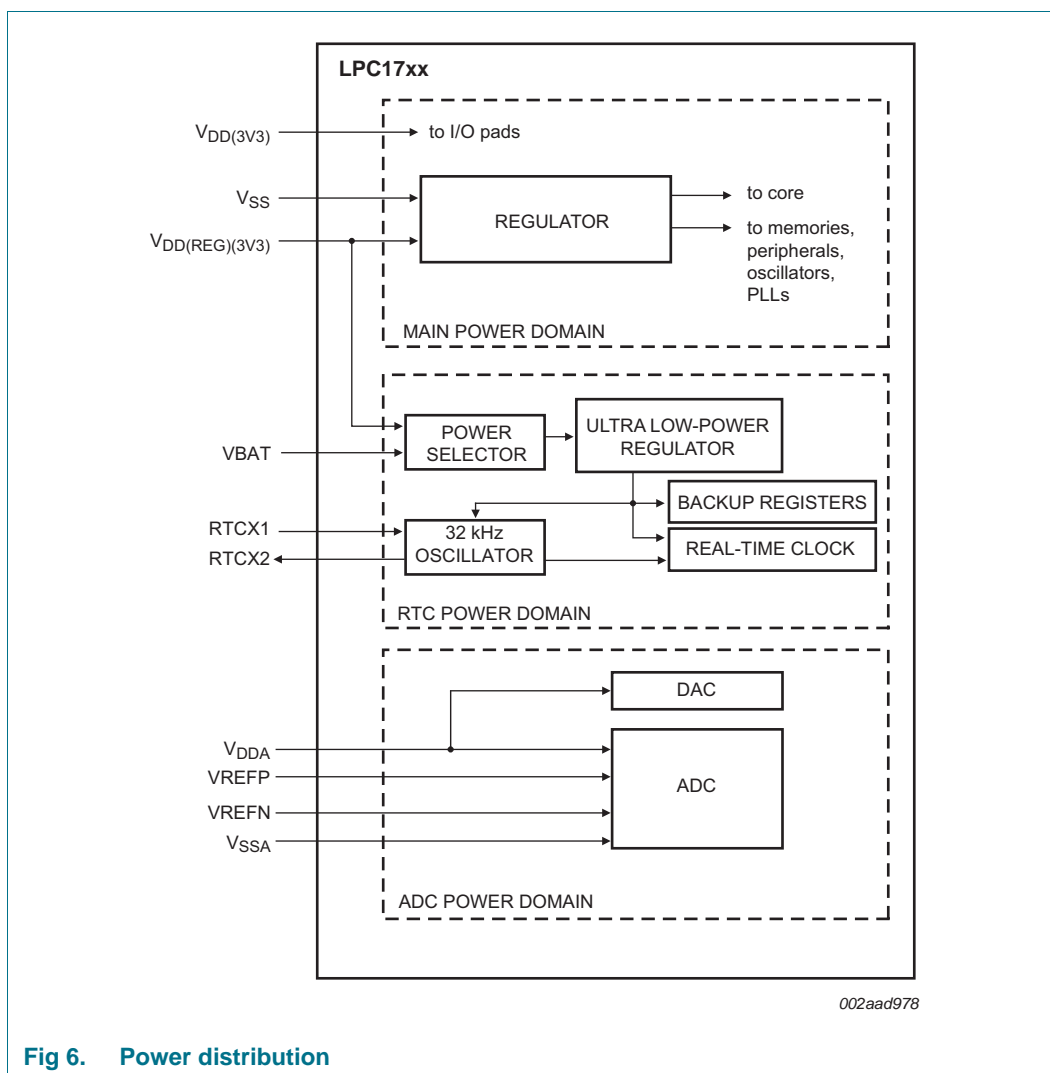


Fig 6. Power distribution

7.30 System control

7.30.1 Reset

Reset has four sources on the LPC17xx: the $\overline{\text{RESET}}$ pin, the Watchdog reset, power-on reset (POR), and the BrownOut Detection (BOD) circuit. The $\overline{\text{RESET}}$ pin is a Schmitt trigger input pin. Assertion of chip Reset by any source, once the operating voltage attains a usable level, causes the $\overline{\text{RSTOUT}}$ pin to go LOW and starts the wake-up timer (see description in [Section 7.29.5](#)). The wake-up timer ensures that reset remains asserted until the external Reset is de-asserted, the oscillator is running, a fixed number of clocks have passed, and the flash controller has completed its initialization. Once reset is de-asserted, or, in case of a BOD-triggered reset, once the voltage rises above the BOD threshold, the $\overline{\text{RSTOUT}}$ pin goes HIGH.

When the internal Reset is removed, the processor begins executing at address 0, which is initially the Reset vector mapped from the Boot Block. At that point, all of the processor and peripheral registers have been initialized to predetermined values.

7.30.2 Brownout detection

The LPC17xx include 2-stage monitoring of the voltage on the $V_{DD(REG)(3V3)}$ pins. If this voltage falls below 2.2 V, the BOD asserts an interrupt signal to the Vectored Interrupt Controller. This signal can be enabled for interrupt in the Interrupt Enable Register in the NVIC in order to cause a CPU interrupt; if not, software can monitor the signal by reading a dedicated status register.

The second stage of low-voltage detection asserts reset to inactivate the LPC17xx when the voltage on the $V_{DD(REG)(3V3)}$ pins falls below 1.85 V. This reset prevents alteration of the flash as operation of the various elements of the chip would otherwise become unreliable due to low voltage. The BOD circuit maintains this reset down below 1 V, at which point the power-on reset circuitry maintains the overall reset.

Both the 2.2 V and 1.85 V thresholds include some hysteresis. In normal operation, this hysteresis allows the 2.2 V detection to reliably interrupt, or a regularly executed event loop to sense the condition.

7.30.3 Code security (Code Read Protection - CRP)

This feature of the LPC17xx allows user to enable different levels of security in the system so that access to the on-chip flash and use of the JTAG and ISP can be restricted. When needed, CRP is invoked by programming a specific pattern into a dedicated flash location. IAP commands are not affected by the CRP.

There are three levels of the Code Read Protection.

CRP1 disables access to chip via the JTAG and allows partial flash update (excluding flash sector 0) using a limited set of the ISP commands. This mode is useful when CRP is required and flash field updates are needed but all sectors can not be erased.

CRP2 disables access to chip via the JTAG and only allows full flash erase and update using a reduced set of the ISP commands.

Running an application with level CRP3 selected fully disables any access to chip via the JTAG pins and the ISP. This mode effectively disables ISP override using P2[10] pin, too. It is up to the user's application to provide (if needed) flash update mechanism using IAP calls or call reinvoke ISP command to enable flash update via UART0.

CAUTION



If level three Code Read Protection (CRP3) is selected, no future factory testing can be performed on the device.

7.30.4 APB interface

The APB peripherals are split into two separate APB buses in order to distribute the bus bandwidth and thereby reducing stalls caused by contention between the CPU and the GPDMA controller.

7.30.5 AHB multilayer matrix

The LPC17xx use an AHB multilayer matrix. This matrix connects the instruction (I-code) and data (D-code) CPU buses of the ARM Cortex-M3 to the flash memory, the main (32 kB) static RAM, and the Boot ROM. The GPDMA can also access all of these memories. The peripheral DMA controllers, Ethernet, and USB can access all SRAM blocks. Additionally, the matrix connects the CPU system bus and all of the DMA controllers to the various peripheral functions.

7.30.6 External interrupt inputs

The LPC17xx include up to 46 edge sensitive interrupt inputs combined with up to four level sensitive external interrupt inputs as selectable pin functions. The external interrupt inputs can optionally be used to wake up the processor from Power-down mode.

7.30.7 Memory mapping control

The Cortex-M3 incorporates a mechanism that allows remapping the interrupt vector table to alternate locations in the memory map. This is controlled via the Vector Table Offset Register contained in the NVIC.

The vector table may be located anywhere within the bottom 1 GB of Cortex-M3 address space. The vector table must be located on a 128 word (512 byte) boundary because the NVIC on the LPC17xx is configured for 128 total interrupts.

7.31 Emulation and debugging

Debug and trace functions are integrated into the ARM Cortex-M3. Serial wire debug and trace functions are supported in addition to a standard JTAG debug and parallel trace functions. The ARM Cortex-M3 is configured to support up to eight breakpoints and four watch points.

8. Limiting values

Table 5. Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134).^[1]

Symbol	Parameter	Conditions	Min	Max	Unit
$V_{DD(3V3)}$	supply voltage (3.3 V)	external rail	2.4	3.6	V
$V_{DD(REG)(3V3)}$	regulator supply voltage (3.3 V)		2.4	3.6	V
V_{DDA}	analog 3.3 V pad supply voltage		-0.5	+4.6	V
$V_{I(VBAT)}$	input voltage on pin VBAT	for the RTC	-0.5	+4.6	V
$V_{I(VREFP)}$	input voltage on pin VREFP		-0.5	+4.6	V
V_{IA}	analog input voltage	on ADC related pins	-0.5	+5.1	V
V_I	input voltage	5 V tolerant I/O pins; only valid when the $V_{DD(3V3)}$ supply voltage is present	^[2] -0.5	+5.5	V
		other I/O pins	^{[2][3]} -0.5	$V_{DD(3V3)} + 0.5$	V
I_{DD}	supply current	per supply pin	-	100	mA
I_{SS}	ground current	per ground pin	-	100	mA
I_{latch}	I/O latch-up current	$-(0.5V_{DD(3V3)}) < V_I < (1.5V_{DD(3V3)})$; $T_J < 125\text{ }^{\circ}\text{C}$	-	100	mA
T_{stg}	storage temperature		^[4] -65	+150	$^{\circ}\text{C}$
$P_{tot(pack)}$	total power dissipation (per package)	based on package heat transfer, not device power consumption	-	1.5	W
V_{ESD}	electrostatic discharge voltage	human body model; all pins	^[5] -4000	+4000	V

[1] The following applies to the limiting values:

- This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maximum.
- Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V_{SS} unless otherwise noted.

[2] Including voltage on outputs in 3-state mode.

[3] Not to exceed 4.6 V.

[4] Dependent on package type.

[5] Human body model: equivalent to discharging a 100 pF capacitor through a 1.5 k Ω series resistor.

9. Thermal characteristics

The average chip junction temperature, T_j (°C), can be calculated using the following equation:

$$T_j = T_{amb} + (P_D \times R_{th(j-a)}) \quad (1)$$

- T_{amb} = ambient temperature (°C),
- $R_{th(j-a)}$ = the package junction-to-ambient thermal resistance (°C/W)
- P_D = sum of internal and I/O power dissipation

The internal power dissipation is the product of I_{DD} and V_{DD} . The I/O power dissipation of the I/O pins is often small and many times can be negligible. However it can be significant in some applications.

Table 6. Thermal characteristics

$V_{DD} = 2.4\text{ V to }3.6\text{ V}$; $T_{amb} = -40\text{ °C to }+85\text{ °C}$ unless otherwise specified;

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$T_{j(max)}$	maximum junction temperature		-	-	125	°C

10. Static characteristics

Table 7. Static characteristics

$T_{amb} = -40\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$, unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ ^[1]	Max	Unit
Supply pins						
$V_{DD(3V3)}$	supply voltage (3.3 V)	external rail	^[2] 2.4	3.3	3.6	V
$V_{DD(REG)(3V3)}$	regulator supply voltage (3.3 V)		2.4	3.3	3.6	V
V_{DDA}	analog 3.3 V pad supply voltage		2.7	3.3	3.6	V
$V_{i(VBAT)}$	input voltage on pin VBAT		^[3] 2.1	3.3	3.6	V
$V_{i(VREFP)}$	input voltage on pin VREFP		2.7	3.3	V_{DDA}	V
$I_{DD(REG)(3V3)}$	regulator supply current (3.3 V)	active mode; code <code>while(1){}</code> executed from flash; all peripherals disabled; $PCLK = CCLK/8$				
		CCLK = 12 MHz; PLL disabled	^{[4][5]} -	7	-	mA
		CCLK = 100 MHz; PLL enabled	^{[4][5]} -	42	-	mA
		CCLK = 100 MHz; PLL enabled (LPC1769)	^{[4][6]} -	50	-	mA
		CCLK = 120 MHz; PLL enabled (LPC1769)	^{[4][6]} -	67	-	mA
		sleep mode	^{[4][7]} -	2	-	mA
		deep sleep mode	^{[4][8]} -	240	-	μA
		power-down mode	^{[4][8]} -	31	-	μA
		deep power-down mode; RTC running	^[9] -	630	-	nA
		deep power-down mode; RTC running				
I_{BAT}	battery supply current	$V_{DD(REG)(3V3)}$ present	^[10] -	530	-	nA
		$V_{DD(REG)(3V3)}$ not present	^[11] -	1.1	-	μA
$I_{DD(IO)}$	I/O supply current	deep sleep mode	^[12] -	40	-	nA
		power-down mode	^[12] -	40	-	nA
		deep power-down mode	^[12] -	10	-	nA

Table 7. Static characteristics ...continued $T_{amb} = -40\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$, unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ ^[1]	Max	Unit
I _{DD} (ADC)	ADC supply current	active mode; [13] [14]	-	1.95	-	mA
		ADC powered				
		ADC in Power-down mode [13] [15]	-	<0.2	-	μA
		deep sleep mode [13]	-	38	-	nA
		power-down mode [13]	-	38	-	nA
		deep power-down mode [13]	-	24	-	nA
I _I (ADC)	ADC input current	on pin VREFP				
		deep sleep mode [16]	-	100	-	nA
		power-down mode [16]	-	100	-	nA
		deep power-down mode [16]	-	100	-	nA
Standard port pins, RESET , RTCK						
I _{IL}	LOW-level input current	V _I = 0 V; on-chip pull-up resistor disabled	-	0.5	10	nA
I _{IH}	HIGH-level input current	V _I = V _{DD(3V3)} ; on-chip pull-down resistor disabled	-	0.5	10	nA
I _{OZ}	OFF-state output current	V _O = 0 V; V _O = V _{DD(3V3)} ; on-chip pull-up/down resistors disabled	-	0.5	10	nA
V _I	input voltage	pin configured to provide a digital function [17] [18] [19]	0	-	5.0	V
V _O	output voltage	output active	0	-	V _{DD(3V3)}	V
V _{IH}	HIGH-level input voltage		0.7V _{DD(3V3)}	-	-	V
V _{IL}	LOW-level input voltage		-	-	0.3V _{DD(3V3)}	V
V _{hys}	hysteresis voltage		0.4	-	-	V
V _{OH}	HIGH-level output voltage	I _{OH} = −4 mA	V _{DD(3V3)} − 0.4	-	-	V
V _{OL}	LOW-level output voltage	I _{OL} = 4 mA	-	-	0.4	V
I _{OH}	HIGH-level output current	V _{OH} = V _{DD(3V3)} − 0.4 V	−4	-	-	mA
I _{OL}	LOW-level output current	V _{OL} = 0.4 V	4	-	-	mA
I _{OHS}	HIGH-level short-circuit output current	V _{OH} = 0 V [20]	-	-	−45	mA
I _{OLS}	LOW-level short-circuit output current	V _{OL} = V _{DD(3V3)} [20]	-	-	50	mA
I _{pd}	pull-down current	V _I = 5 V	10	50	150	μA
I _{pu}	pull-up current	V _I = 0 V	−15	−50	−85	μA
		V _{DD(3V3)} < V _I < 5 V	0	0	0	μA

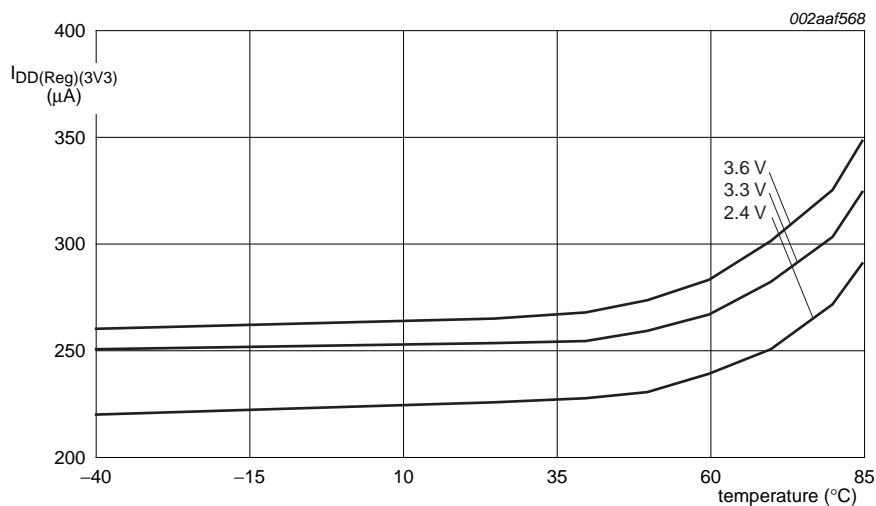
Table 7. Static characteristics ...continued $T_{amb} = -40\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$, unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ ^[1]	Max	Unit
I ² C-bus pins (P0[27] and P0[28])						
V _{IH}	HIGH-level input voltage		0.7V _{DD(3V3)}	-	-	V
V _{IL}	LOW-level input voltage		-	-	0.3V _{DD(3V3)}	V
V _{hys}	hysteresis voltage		-	0.05 × V _{DD(3V3)}	-	V
V _{OL}	LOW-level output voltage	I _{OLS} = 3 mA	-	-	0.4	V
I _{LI}	input leakage current	V _I = V _{DD(3V3)}	^[21] -	2	4	μA
		V _I = 5 V	-	10	22	μA
Oscillator pins						
V _{i(XTAL1)}	input voltage on pin XTAL1		−0.5	1.8	1.95	V
V _{o(XTAL2)}	output voltage on pin XTAL2		−0.5	1.8	1.95	V
V _{i(RTCX1)}	input voltage on pin RTCX1		−0.5	-	3.6	V
V _{o(RTCX2)}	output voltage on pin RTCX2		−0.5	-	3.6	V
USB pins (LPC1769/68/66/65/64 only)						
I _{OZ}	OFF-state output current	0 V < V _I < 3.3 V	^[2] -	-	±10	μA
V _{BUS}	bus supply voltage		^[2] -	-	5.25	V
V _{DI}	differential input sensitivity voltage	(D+) – (D–)	^[2] 0.2	-	-	V
V _{CM}	differential common mode voltage range	includes V _{DI} range	^[2] 0.8	-	2.5	V
V _{th(rs)se}	single-ended receiver switching threshold voltage		^[2] 0.8	-	2.0	V
V _{OL}	LOW-level output voltage for low-/full-speed	R _L of 1.5 kΩ to 3.6 V	^[2] -	-	0.18	V
V _{OH}	HIGH-level output voltage (driven) for low-/full-speed	R _L of 15 kΩ to GND	^[2] 2.8	-	3.5	V
C _{trans}	transceiver capacitance	pin to GND	^[2] -	-	20	pF
Z _{DRV}	driver output impedance for driver which is not high-speed capable	with 33 Ω series resistor; steady state drive	^{[22][2]} 36	-	44.1	Ω

[1] Typical ratings are not guaranteed. The values listed are at room temperature (25 $^{\circ}\text{C}$), nominal supply voltages.[2] For USB operation $3.0\text{ V} \leq V_{DD(3V3)} \leq 3.6\text{ V}$. Guaranteed by design.[3] The RTC typically fails when $V_{i(VBAT)}$ drops below 1.6 V.[4] $V_{DD(REG)(3V3)} = 3.3\text{ V}$; $T_{amb} = 25\text{ }^{\circ}\text{C}$ for all power consumption measurements.

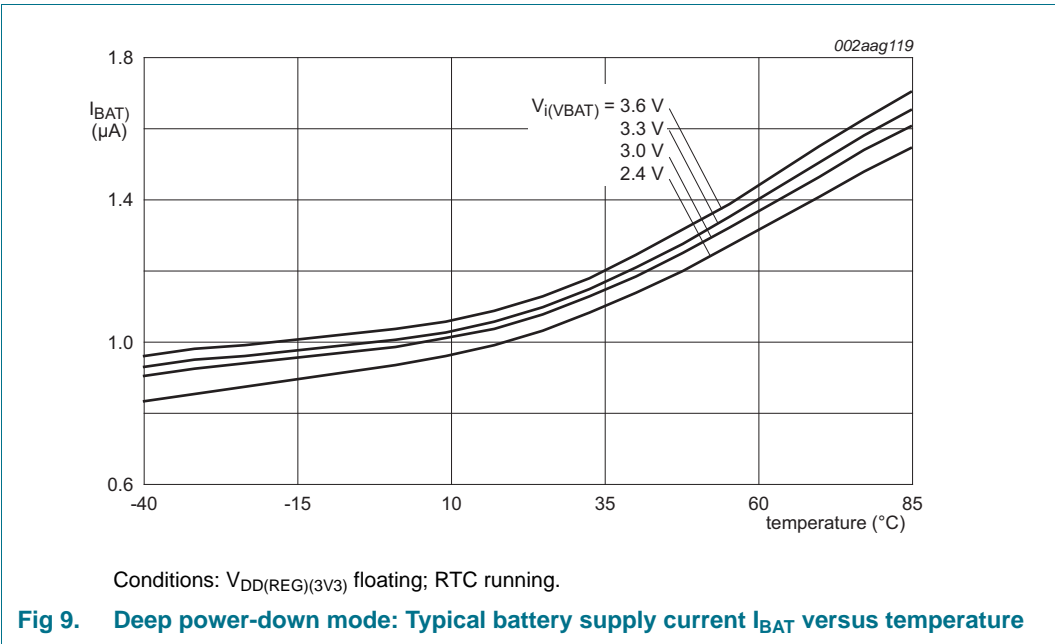
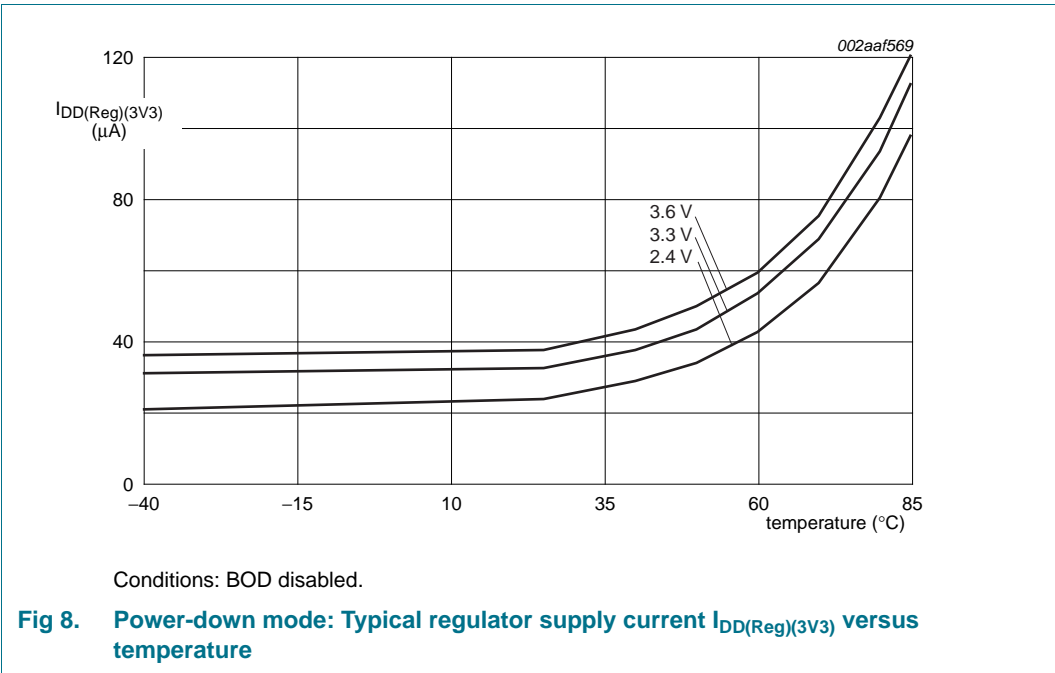
- [5] Applies to LPC1768/67/66/65/64/63.
- [6] Applies to LPC1769 only.
- [7] IRC running at 4 MHz; main oscillator and PLL disabled; $PCLK = CCLK/8$.
- [8] BOD disabled.
- [9] On pin $V_{DD(Reg)(3V3)}$, $I_{BAT} = 530$ nA. $V_{DD(Reg)(3V3)} = 3.0$ V; $V_{BAT} = 3.0$ V; $T_{amb} = 25$ °C.
- [10] On pin V_{BAT} ; $I_{DD(Reg)(3V3)} = 630$ nA; $V_{DD(Reg)(3V3)} = 3.0$ V; $V_{BAT} = 3.0$ V; $T_{amb} = 25$ °C.
- [11] On pin V_{BAT} ; $V_{BAT} = 3.0$ V; $T_{amb} = 25$ °C.
- [12] All internal pull-ups disabled. All pins configured as output and driven LOW. $V_{DD(3V3)} = 3.3$ V; $T_{amb} = 25$ °C.
- [13] On pin V_{DDA} ; $V_{DDA} = 3.3$ V; $T_{amb} = 25$ °C. The ADC is powered if the PDN bit in the AD0CR register is set to 1 and in Power-down mode of the PDN bit is set to 0.
- [14] The ADC is powered if the PDN bit in the AD0CR register is set to 1. See *LPC17xx user manual UM10360_1*.
- [15] The ADC is in Power-down mode if the PDN bit in the AD0CR register is set to 0. See *LPC17xx user manual UM10360_1*.
- [16] $V_{i(VREFP)} = 3.3$ V; $T_{amb} = 25$ °C.
- [17] Including voltage on outputs in 3-state mode.
- [18] $V_{DD(3V3)}$ supply voltages must be present.
- [19] 3-state outputs go into 3-state mode in Deep power-down mode.
- [20] Allowed as long as the current limit does not exceed the maximum current allowed by the device.
- [21] To V_{SS} .
- [22] Includes external resistors of $33\ \Omega \pm 1\%$ on D+ and D-.

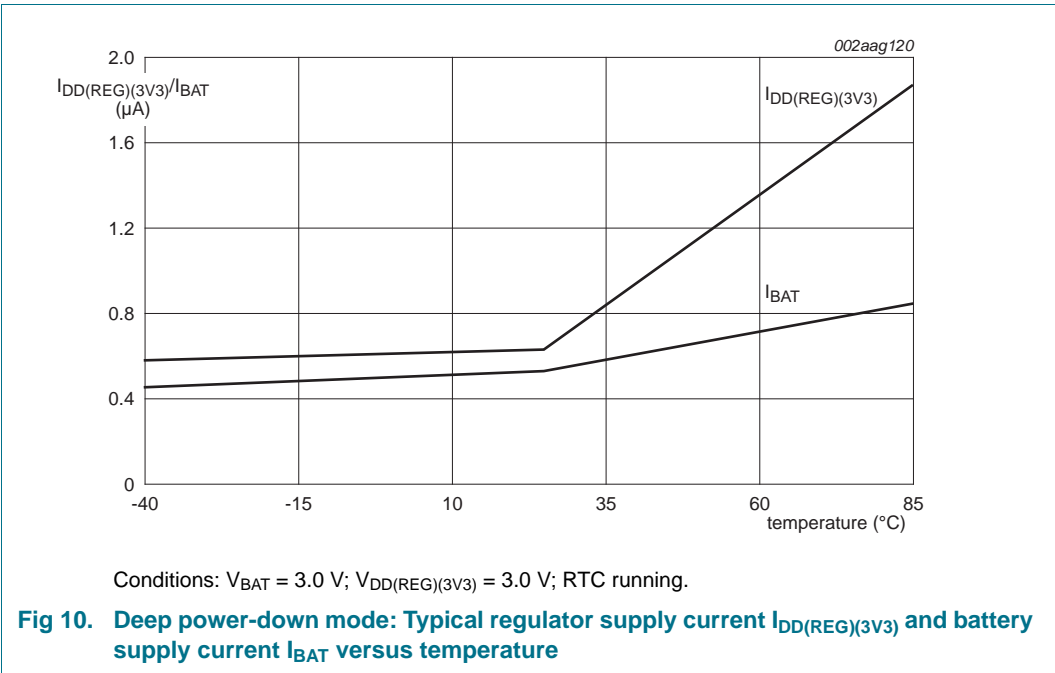
10.1 Power consumption



Conditions: BOD disabled.

Fig 7. Deep-sleep mode: typical regulator supply current $I_{DD(Reg)(3V3)}$ versus temperature





10.2 Peripheral power consumption

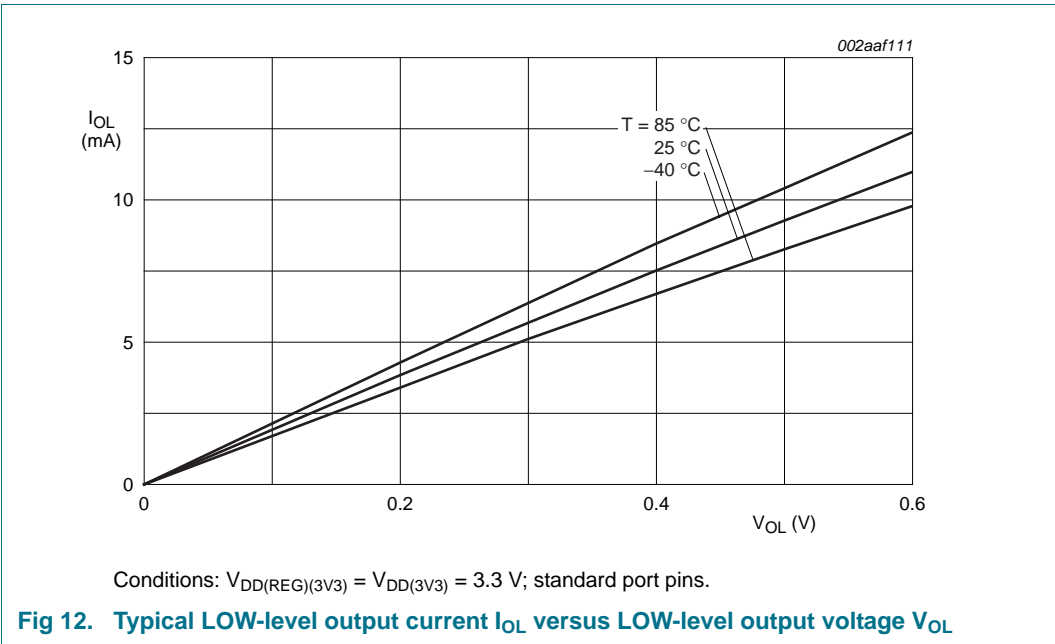
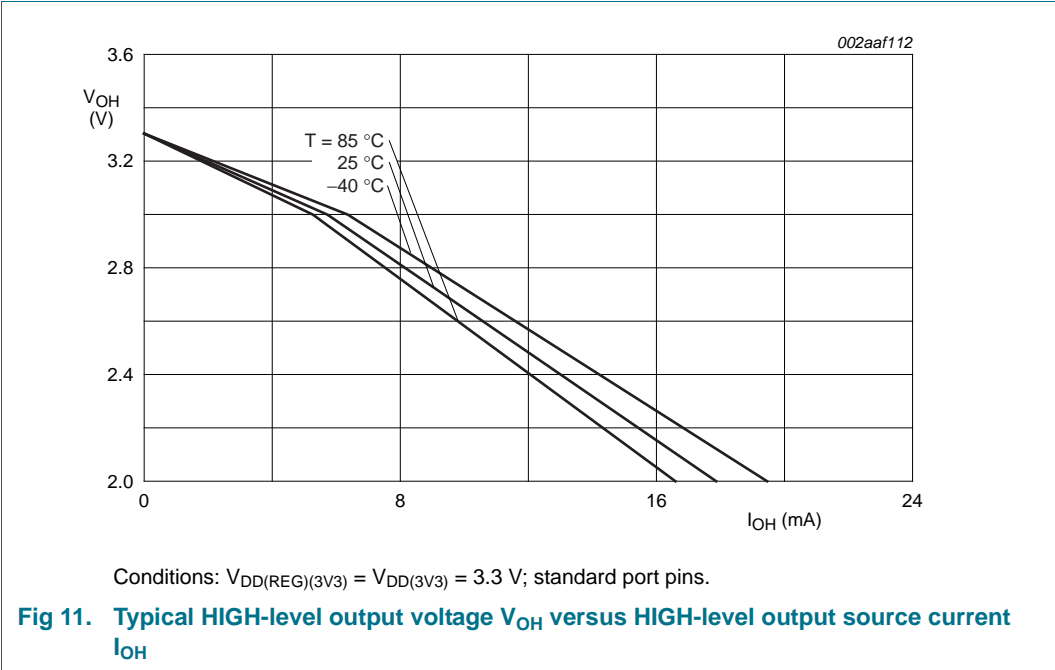
The supply current per peripheral is measured as the difference in supply current between the peripheral block enabled and the peripheral block disabled in the PCONP register. All other blocks are disabled and no code is executed. Measured on a typical sample at $T_{amb} = 25\text{ }^{\circ}\text{C}$. The peripheral clock PCLK = CCLK/4.

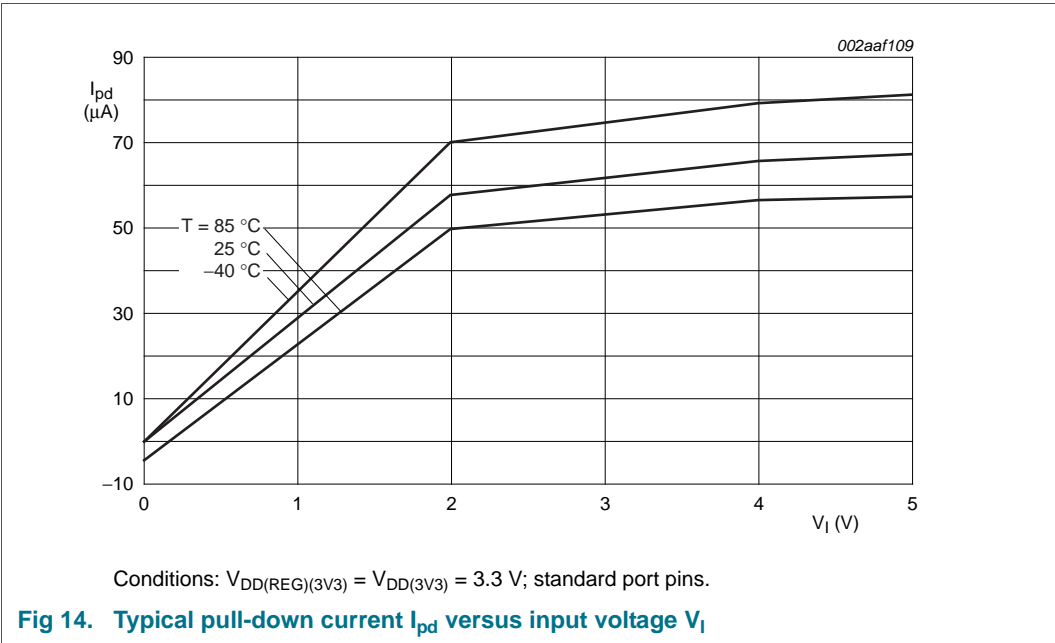
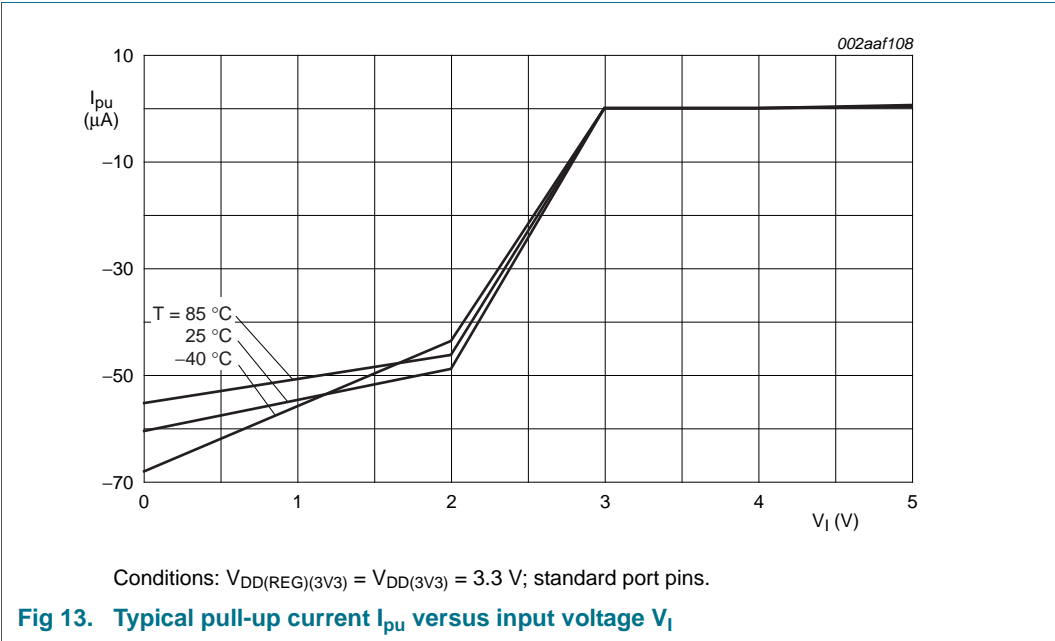
Table 8. Power consumption for individual analog and digital blocks

Peripheral	Conditions	Typical supply current in mA; CCLK =			Notes
		12 MHz	48 MHz	100 MHz	
Timer		0.03	0.11	0.23	Average current per timer
UART		0.07	0.26	0.53	Average current per UART
PWM		0.05	0.20	0.41	
Motor control PWM		0.05	0.21	0.42	
I2C		0.02	0.08	0.16	Average current per I2C
SPI		0.02	0.06	0.13	
SSP1		0.04	0.16	0.32	
ADC	PCLK = 12 MHz for CCLK = 12 MHz and 48 MHz; PCLK = 12.5 MHz for CCLK = 100 MHz	2.12	2.09	2.07	
CAN	PCLK = CCLK/6	0.13	0.49	1.00	Average current per CAN
CAN0, CAN1, acceptance filter	PCLK = CCLK/6	0.22	0.85	1.73	Both CAN blocks and acceptance filter ^[1]
DMA	PCLK = CCLK	1.33	5.10	10.36	
QEI		0.05	0.20	0.41	
GPIO		0.33	1.27	2.58	
I2S		0.09	0.34	0.70	
USB and PLL1		0.94	1.32	1.94	
Ethernet	Ethernet block enabled in the PCONP register; Ethernet not connected.	0.49	1.87	3.79	
Ethernet connected	Ethernet initialized, connected to network, and running web server example.	-	-	5.19	

[1] The combined current of several peripherals running at the same time can be less than the sum of each individual peripheral current measured separately.

10.3 Electrical pin characteristics





11. Dynamic characteristics

11.1 Flash memory

Table 9. Flash characteristics

$T_{amb} = -40\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$, unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
N_{endu}	endurance		[1] 10000	100000	-	cycles
t_{ret}	retention time	powered	10	-	-	years
		unpowered	20	-	-	years
t_{er}	erase time	sector or multiple consecutive sectors	95	100	105	ms
t_{prog}	programming time		[2] 0.95	1	1.05	ms

[1] Number of program/erase cycles.

[2] Programming times are given for writing 256 bytes from RAM to the flash. Data must be written to the flash in blocks of 256 bytes.

11.2 External clock

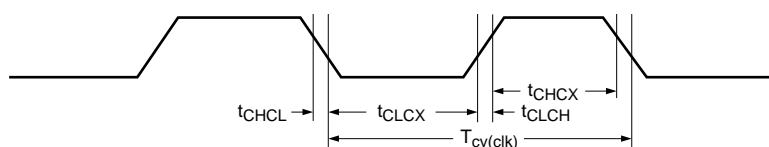
Table 10. Dynamic characteristic: external clock

$T_{amb} = -40\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$; $V_{DD(3V3)}$ over specified ranges. [1]

Symbol	Parameter	Conditions	Min	Typ [2]	Max	Unit
f_{osc}	oscillator frequency		1	-	25	MHz
$T_{cy(clk)}$	clock cycle time		40	-	1000	ns
t_{CHCX}	clock HIGH time		$T_{cy(clk)} \times 0.4$	-	-	ns
t_{CLCX}	clock LOW time		$T_{cy(clk)} \times 0.4$	-	-	ns
t_{CLCH}	clock rise time		-	-	5	ns
t_{CHCL}	clock fall time		-	-	5	ns

[1] Parameters are valid over operating temperature range unless otherwise specified.

[2] Typical ratings are not guaranteed. The values listed are at room temperature (25 °C), nominal supply voltages.



002aaa907

Fig 15. External clock timing (with an amplitude of at least $V_{i(RMS)} = 200\text{ mV}$)

11.3 Internal oscillators

Table 11. Dynamic characteristic: internal oscillators

$T_{amb} = -40\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$; $2.7\text{ V} \leq V_{DD(REG)(3V3)} \leq 3.6\text{ V}$ [1]

Symbol	Parameter	Conditions	Min	Typ[2]	Max	Unit
$f_{osc(RC)}$	internal RC oscillator frequency	-	3.96	4.02	4.04	MHz
$f_{i(RTC)}$	RTC input frequency	-	-	32.768	-	kHz

[1] Parameters are valid over operating temperature range unless otherwise specified.

[2] Typical ratings are not guaranteed. The values listed are at room temperature ($25\text{ }^{\circ}\text{C}$), nominal supply voltages.

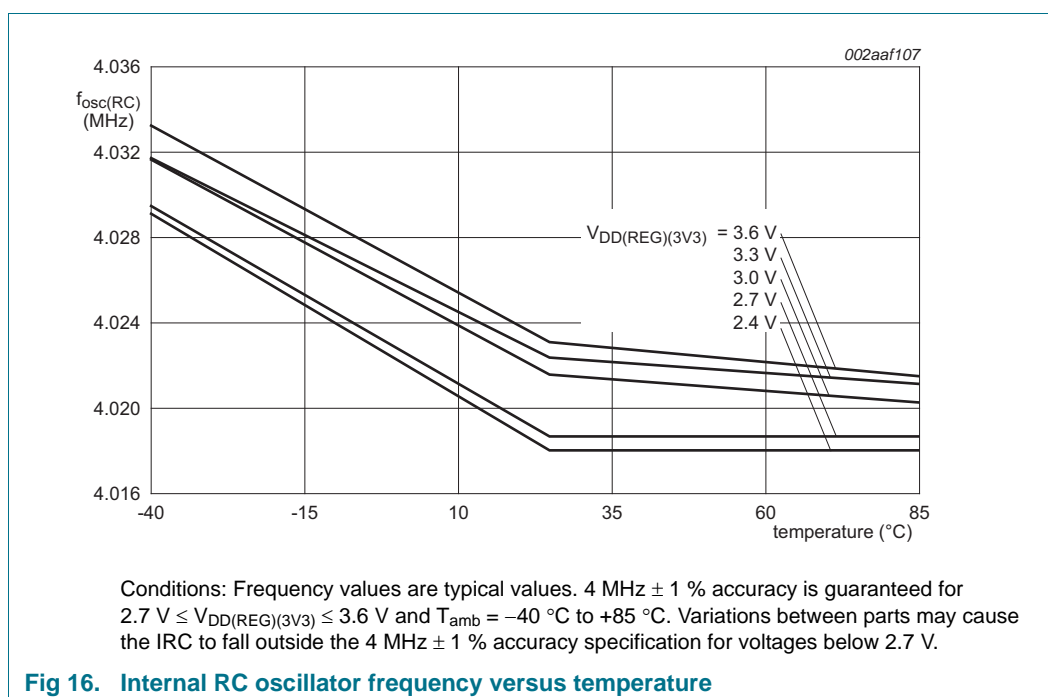


Fig 16. Internal RC oscillator frequency versus temperature

11.4 I/O pins

Table 12. Dynamic characteristic: I/O pins [1]

$T_{amb} = -40\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$; $V_{DD(3V3)}$ over specified ranges.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_r	rise time	pin configured as output	3.0	-	5.0	ns
t_f	fall time	pin configured as output	2.5	-	5.0	ns

[1] Applies to standard I/O pins.

11.5 I²C-bus

Table 13. Dynamic characteristic: I²C-bus pins^[1]

$T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$.^[2]

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCL}	SCL clock frequency	Standard-mode	0	100	kHz
		Fast-mode	0	400	kHz
		Fast-mode Plus	0	1	MHz
t_f	fall time	^{[4][5][6][7]} of both SDA and SCL signals	-	300	ns
		Standard-mode			
		Fast-mode	$20 + 0.1 \times C_b$	300	ns
t_{LOW}	LOW period of the SCL clock	Fast-mode Plus	-	120	ns
		Standard-mode	4.7	-	μs
		Fast-mode	1.3	-	μs
t_{HIGH}	HIGH period of the SCL clock	Fast-mode Plus	0.5	-	μs
		Standard-mode	4.0	-	μs
		Fast-mode	0.6	-	μs
$t_{HD;DAT}$	data hold time	^{[3][4][8]} Standard-mode	0	-	μs
		Fast-mode	0	-	μs
		Fast-mode Plus	0	-	μs
$t_{SU;DAT}$	data set-up time	^{[9][10]} Standard-mode	250	-	ns
		Fast-mode	100	-	ns
		Fast-mode Plus	50	-	ns

[1] See the I²C-bus specification *UM10204* for details.

[2] Parameters are valid over operating temperature range unless otherwise specified.

[3] $t_{HD;DAT}$ is the data hold time that is measured from the falling edge of SCL; applies to data in transmission and the acknowledge.

[4] A device must internally provide a hold time of at least 300 ns for the SDA signal (with respect to the $V_{IH(min)}$ of the SCL signal) to bridge the undefined region of the falling edge of SCL.

[5] C_b = total capacitance of one bus line in pF.

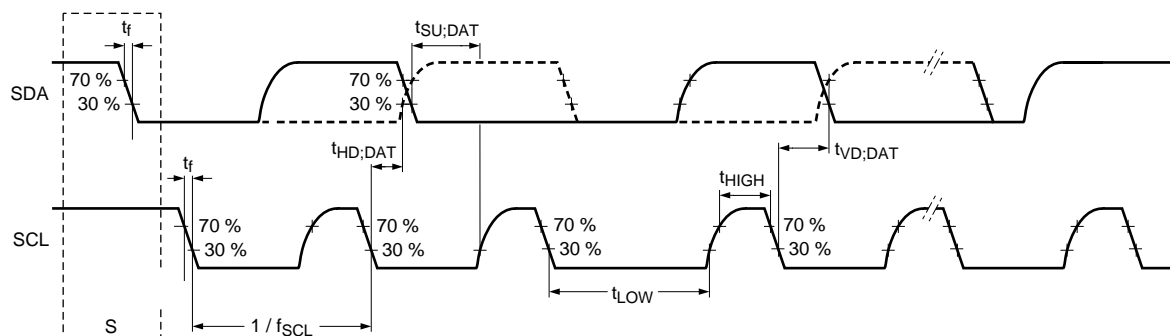
[6] The maximum t_f for the SDA and SCL bus lines is specified at 300 ns. The maximum fall time for the SDA output stage t_f is specified at 250 ns. This allows series protection resistors to be connected in between the SDA and the SCL pins and the SDA/SCL bus lines without exceeding the maximum specified t_f .

[7] In Fast-mode Plus, fall time is specified the same for both output stage and bus timing. If series resistors are used, designers should allow for this when considering bus timing.

[8] The maximum $t_{HD;DAT}$ could be 3.45 μs and 0.9 μs for Standard-mode and Fast-mode but must be less than the maximum of $t_{VD;DAT}$ or $t_{VD;ACK}$ by a transition time (see the I²C-bus specification *UM10204*). This maximum must only be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal. If the clock stretches the SCL, the data must be valid by the set-up time before it releases the clock.

[9] $t_{SU;DAT}$ is the data set-up time that is measured with respect to the rising edge of SCL; applies to data in transmission and the acknowledge.

[10] A Fast-mode I²C-bus device can be used in a Standard-mode I²C-bus system but the requirement $t_{SU;DAT} = 250\text{ ns}$ must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line $t_{r(max)} + t_{SU;DAT} = 1000 + 250 = 1250\text{ ns}$ (according to the Standard-mode I²C-bus specification) before the SCL line is released. Also the acknowledge timing must meet this set-up time.



002aaf425

Fig 17. I²C-bus pins clock timing

11.6 I²S-bus interface

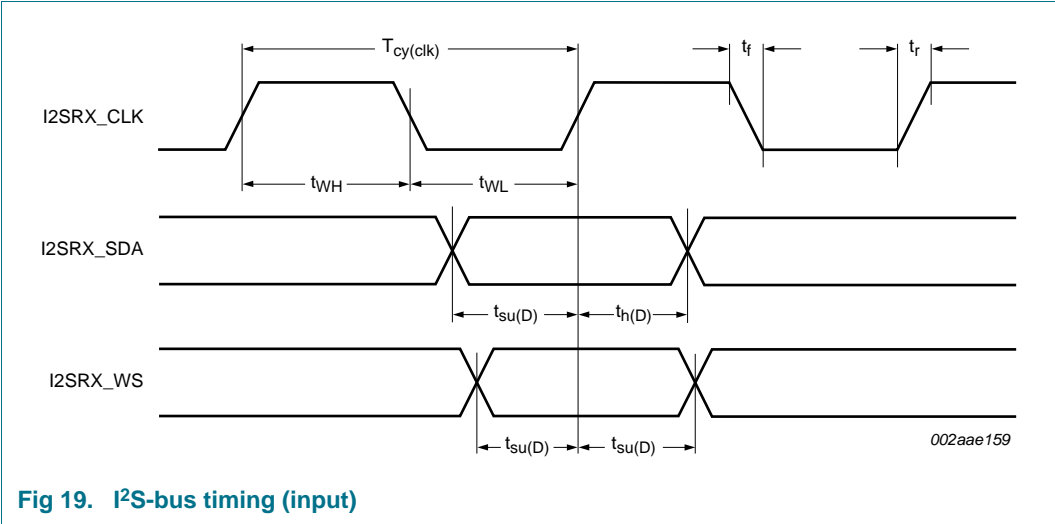
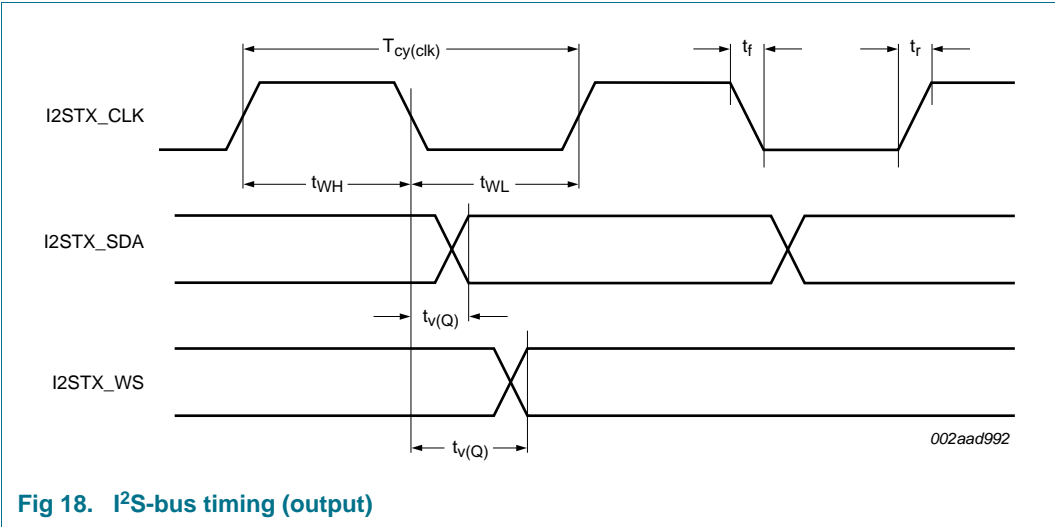
Remark: The I²S-bus interface is available on parts LPC1769/68/67/66/65/63. See [Table 2](#).

Table 14. Dynamic characteristics: I²S-bus interface pins

$T_{amb} = -40\text{ }^{\circ}\text{C to } +85\text{ }^{\circ}\text{C}$.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
common to input and output						
t _r	rise time		[1]	-	35	ns
t _f	fall time		[1]	-	35	ns
t _{WH}	pulse width HIGH	on pins I2STX_CLK and I2SRX_CLK	[1]	0.495 × T _{cy(clk)}	-	-
t _{WL}	pulse width LOW	on pins I2STX_CLK and I2SRX_CLK	[1]	-	0.505 × T _{cy(clk)}	ns
output						
t _{v(Q)}	data output valid time	on pin I2STX_SDA	[1]	-	30	ns
		on pin I2STX_WS	[1]	-	30	ns
input						
t _{su(D)}	data input set-up time	on pin I2SRX_SDA	[1]	3.5	-	ns
t _{h(D)}	data input hold time	on pin I2SRX_SDA	[1]	4.0	-	ns

[1] CCLK = 20 MHz; peripheral clock to the I²S-bus interface PCLK = $CCLK/4$; I²S clock cycle time $T_{cy(clk)} = 1600\text{ ns}$, corresponds to the SCK signal in the I²S-bus specification.

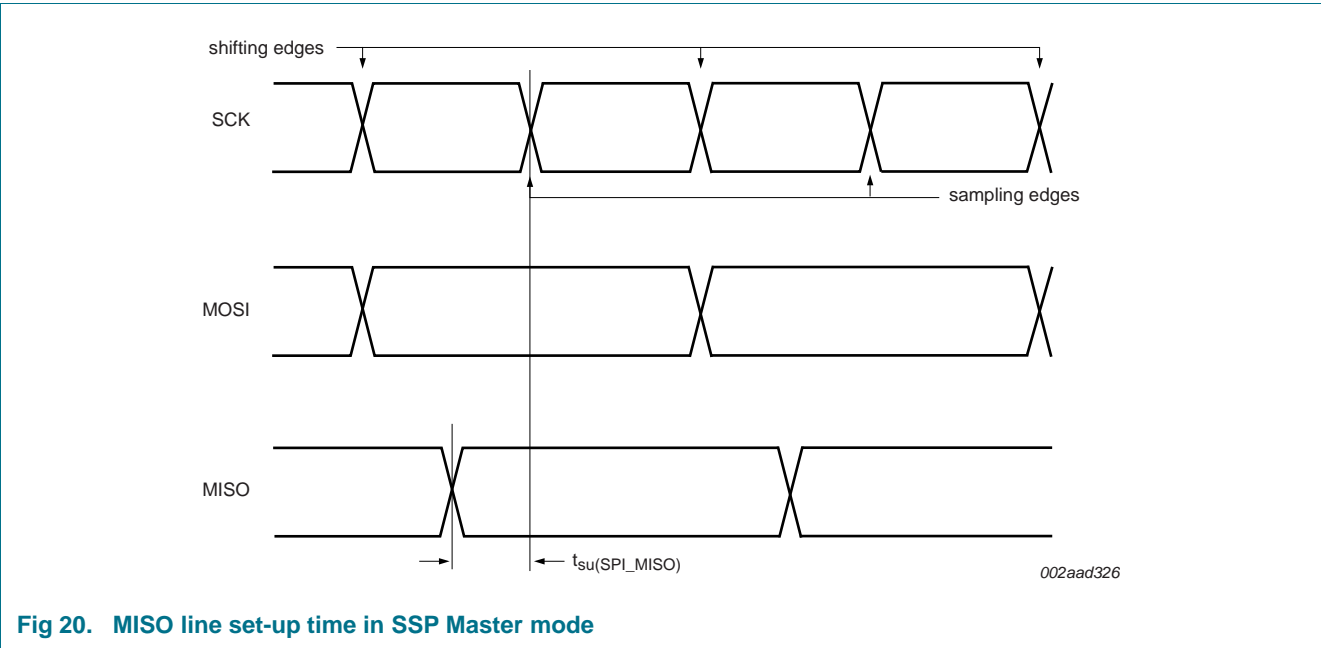


11.7 SSP interface

Table 15. Dynamic characteristic: SSP interface
Tamb = 25 °C; VDD(3V3) over specified ranges.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
SSP interface						
t _{su} (SPI_MISO)	SPI_MISO set-up time	measured in SPI Master mode; see Figure 20	[1] 30		-	ns

[1] The peripheral clock for SSP is PCLK = CCLK = 20 MHz.



11.8 USB interface

Remark: The USB controller is available as a device/Host/OTG controller on parts LPC1769/68/66/65 and as device-only controller on part LPC1764.

Table 16. Dynamic characteristics: USB pins (full-speed)

$C_L = 50\text{ pF}$; $R_{pu} = 1.5\text{ k}\Omega$ on D+ to $V_{DD(3V3)}$; $3.0\text{ V} \leq V_{DD(3V3)} \leq 3.6\text{ V}$.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_r	rise time	10 % to 90 %	8.5	-	13.8	ns
t_f	fall time	10 % to 90 %	7.7	-	13.7	ns
t_{FRFM}	differential rise and fall time matching	t_r / t_f	-	-	109	%
V_{CRS}	output signal crossover voltage		1.3	-	2.0	V
t_{FEOP}	source SE0 interval of EOP	see Figure 21	160	-	175	ns
t_{FDEOP}	source jitter for differential transition to SE0 transition	see Figure 21	-2	-	+5	ns
t_{JR1}	receiver jitter to next transition		-18.5	-	+18.5	ns
t_{JR2}	receiver jitter for paired transitions	10 % to 90 %	-9	-	+9	ns
t_{EOPR1}	EOP width at receiver	must reject as EOP; see Figure 21	[1] 40	-	-	ns
t_{EOPR2}	EOP width at receiver	must accept as EOP; see Figure 21	[1] 82	-	-	ns

[1] Characterized but not implemented as production test. Guaranteed by design.

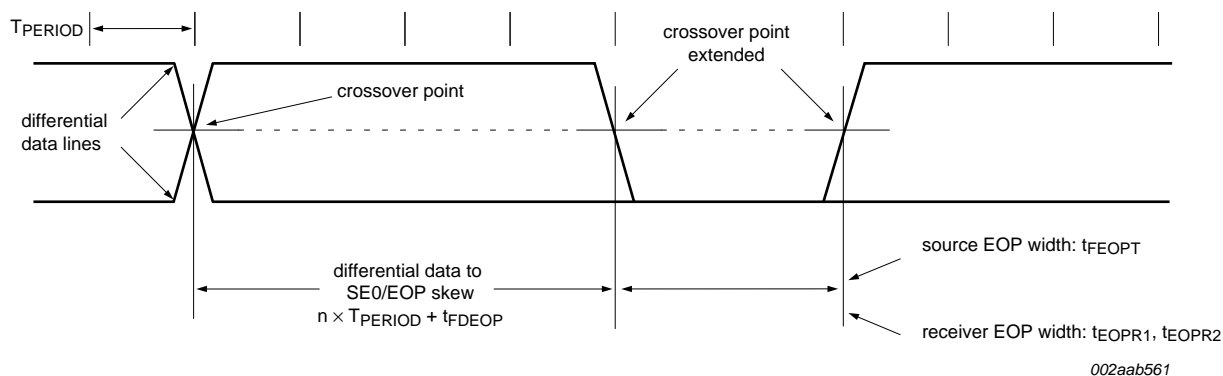


Fig 21. Differential data-to-EOP transition skew and EOP width

11.9 SPI

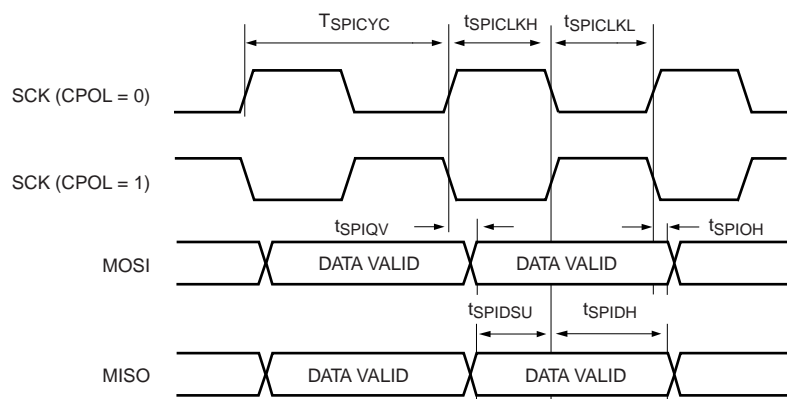
Table 17. Dynamic characteristics of SPI pins

$T_{amb} = -40\text{ }^{\circ}\text{C}$ to $+85\text{ }^{\circ}\text{C}$.

Symbol	Parameter	Min	Typ	Max	Unit
$T_{cy(PCLK)}$	PCLK cycle time	10	-	-	ns
T_{SPICYC}	SPI cycle time	[1] 79.6	-	-	ns
$t_{SPICLK H}$	SPICLK HIGH time	$0.485 \times T_{SPICYC}$	-	-	ns
$t_{SPICLK L}$	SPICLK LOW time	-	-	$0.515 \times T_{SPICYC}$	ns
SPI master					
t_{SPIDSU}	SPI data set-up time	[2] 0	-	-	ns
t_{SPIDH}	SPI data hold time	[2] $2 \times T_{cy(PCLK)} - 5$	-	-	ns
t_{SPIQV}	SPI data output valid time	[2] $2 \times T_{cy(PCLK)} + 30$	-	-	ns
t_{SPIOH}	SPI output data hold time	[2] $2 \times T_{cy(PCLK)} + 5$	-	-	ns
SPI slave					
t_{SPIDSU}	SPI data set-up time	[2] 0	-	-	ns
t_{SPIDH}	SPI data hold time	[2] $2 \times T_{cy(PCLK)} + 5$	-	-	ns
t_{SPIQV}	SPI data output valid time	[2] $2 \times T_{cy(PCLK)} + 35$	-	-	ns
t_{SPIOH}	SPI output data hold time	[2] $2 \times T_{cy(PCLK)} + 15$	-	-	ns

[1] $T_{SPICYC} = (T_{cy(PCLK)} \times n) \pm 0.5\%$, n is the SPI clock divider value ($n \geq 8$); PCLK is derived from the processor clock CCLK.

[2] Timing parameters are measured with respect to the 50 % edge of the clock SCK and the 10 % (90 %) edge of the data signal (MOSI or MISO).



002aad986

Fig 22. SPI master timing (CPHA = 1)

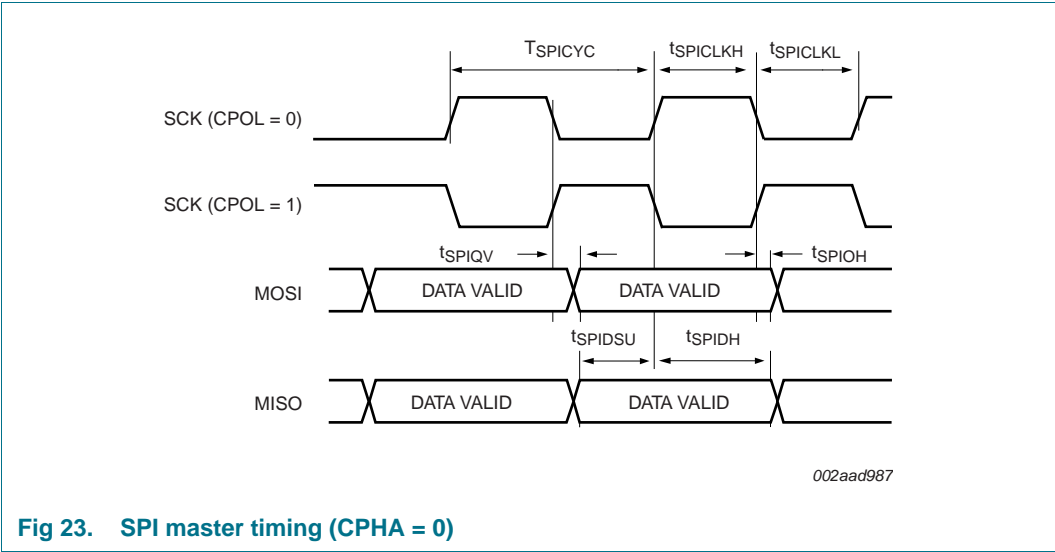


Fig 23. SPI master timing (CPHA = 0)

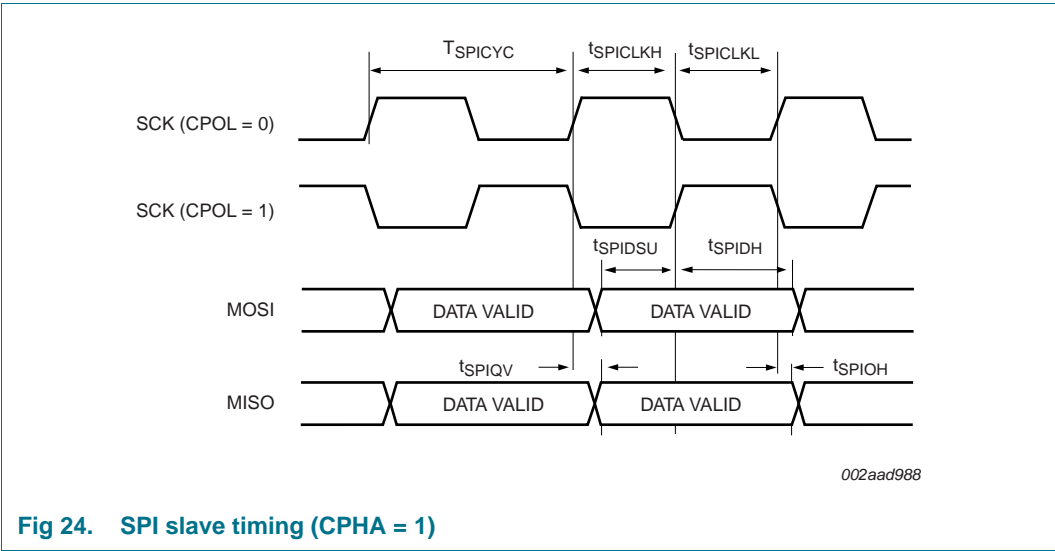


Fig 24. SPI slave timing (CPHA = 1)

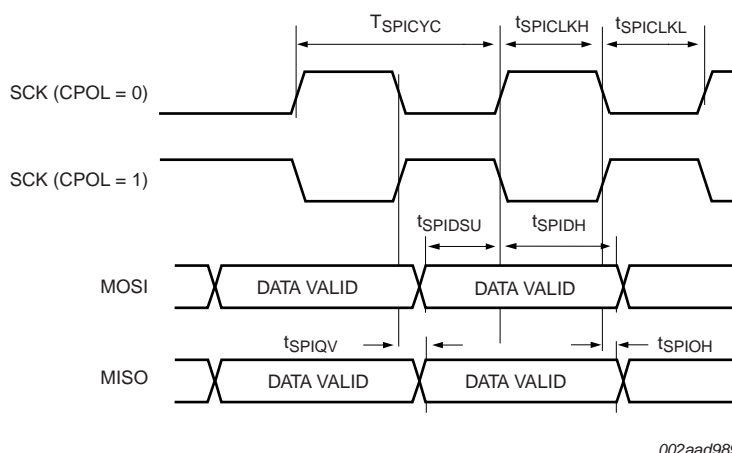


Fig 25. SPI slave timing (CPHA = 0)

12. ADC electrical characteristics

Table 18. ADC characteristics (full resolution)

$V_{DDA} = 2.7\text{ V to }3.6\text{ V}$; $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ unless otherwise specified; ADC frequency 13 MHz; 12-bit resolution.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IA}	analog input voltage		0	-	V_{DDA}	V
C_{ia}	analog input capacitance		-	-	15	pF
E_D	differential linearity error	[1][2]	-	-	± 1	LSB
$E_{L(adj)}$	integral non-linearity	[3]	-	-	± 3	LSB
E_O	offset error	[4][5]	-	-	± 2	LSB
E_G	gain error	[6]	-	-	0.5	%
E_T	absolute error	[7]	-	-	4	LSB
R_{vsi}	voltage source interface resistance	[8]	-	-	7.5	k Ω
$f_{clk(ADC)}$	ADC clock frequency		-	-	13	MHz
$f_{c(ADC)}$	ADC conversion frequency	[9]	-	-	200	kHz

- [1] The ADC is monotonic, there are no missing codes.
- [2] The differential linearity error (E_D) is the difference between the actual step width and the ideal step width. See [Figure 26](#).
- [3] The integral non-linearity ($E_{L(adj)}$) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset errors. See [Figure 26](#).
- [4] The offset error (E_O) is the absolute difference between the straight line which fits the actual curve and the straight line which fits the ideal curve. See [Figure 26](#).
- [5] ADCOFFS value (bits 7:4) = 2 in the ADTRM register. See *LPC17xx user manual UM10360*.
- [6] The gain error (E_G) is the relative difference in percent between the straight line fitting the actual transfer curve after removing offset error, and the straight line which fits the ideal transfer curve. See [Figure 26](#).
- [7] The absolute error (E_T) is the maximum difference between the center of the steps of the actual transfer curve of the non-calibrated ADC and the ideal transfer curve. See [Figure 26](#).
- [8] See [Figure 27](#).
- [9] The conversion frequency corresponds to the number of samples per second.

Table 19. ADC characteristics (lower resolution)*T_{amb} = -40 °C to +85 °C unless otherwise specified; 12-bit ADC used as 10-bit resolution ADC.*

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
E _D	differential linearity error		[1][2] -	±1	-	LSB
E _{L(adj)}	integral non-linearity		[3] -	±1.5	-	LSB
E _O	offset error		[4] -	±2	-	LSB
E _G	gain error		[5] -	±2	-	LSB
f _{clk(ADC)}	ADC clock frequency	3.0 V ≤ V _{DDA} ≤ 3.6 V	-	-	33	MHz
		2.7 V ≤ V _{DDA} < 3.0 V	-	-	25	MHz
f _{c(ADC)}	ADC conversion frequency	3 V ≤ V _{DDA} ≤ 3.6 V	[6] -	-	500	kHz
		2.7 V ≤ V _{DDA} < 3.0 V	[6] -	-	400	kHz

[1] The ADC is monotonic, there are no missing codes.

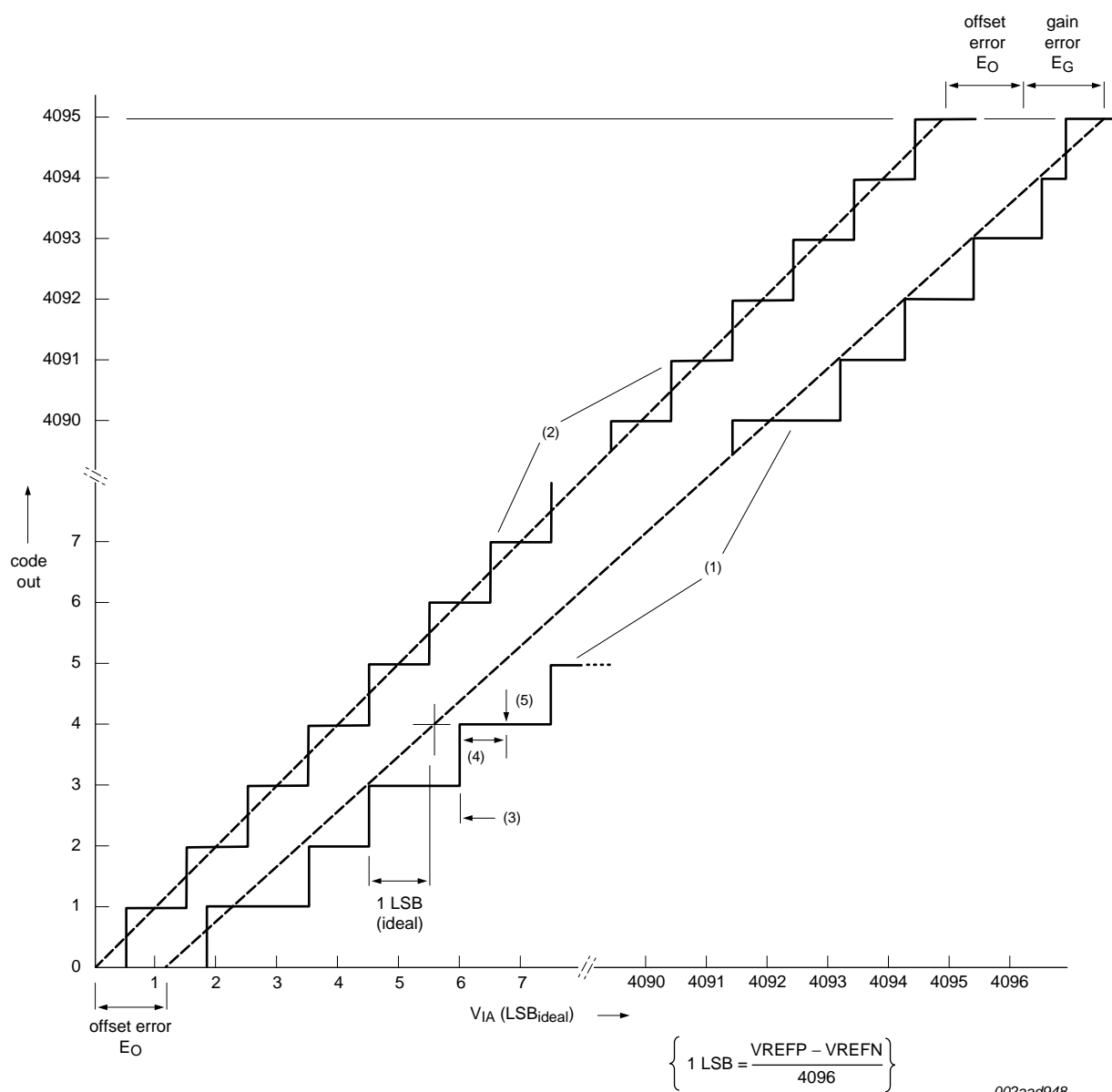
[2] The differential linearity error (E_D) is the difference between the actual step width and the ideal step width. See [Figure 26](#).

[3] The integral non-linearity (E_{L(adj)}) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset errors. See [Figure 26](#).

[4] The offset error (E_O) is the absolute difference between the straight line which fits the actual curve and the straight line which fits the ideal curve. See [Figure 26](#).

[5] The gain error (E_G) is the relative difference in percent between the straight line fitting the actual transfer curve after removing offset error, and the straight line which fits the ideal transfer curve. See [Figure 26](#).

[6] The conversion frequency corresponds to the number of samples per second.



- (1) Example of an actual transfer curve.
- (2) The ideal transfer curve.
- (3) Differential linearity error (E_D).
- (4) Integral non-linearity ($E_{L(adj)}$).
- (5) Center of a step of the actual transfer curve.

Fig 26. 12-bit ADC characteristics

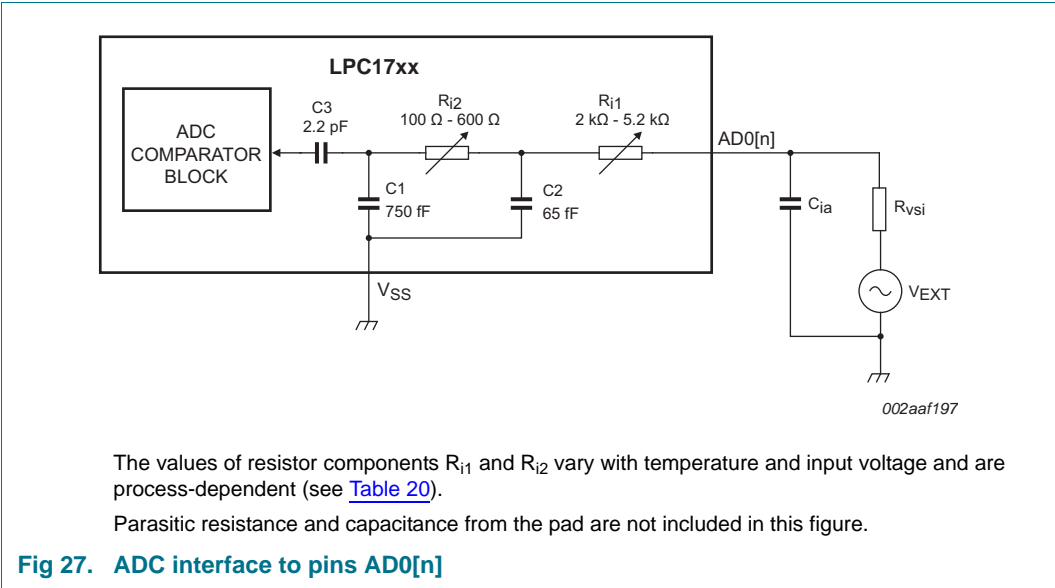


Table 20. ADC interface components

Component	Range	Description
R_{i1}	2 kΩ to 5.2 kΩ	Switch-on resistance for channel selection switch. Varies with temperature, input voltage, and process.
R_{i2}	100 Ω to 600 Ω	Switch-on resistance for the comparator input switch. Varies with temperature, input voltage, and process.
C1	750 fF	Parasitic capacitance from the ADC block level.
C2	65 fF	Parasitic capacitance from the ADC block level.
C3	2.2 pF	Sampling capacitor.

13. DAC electrical characteristics

Remark: The DAC is available on parts LPC1769/68/67/66/65/63. See [Table 2](#).

Table 21. DAC electrical characteristics

$V_{DDA} = 2.7\text{ V to }3.6\text{ V}$; $T_{amb} = -40\text{ °C to }+85\text{ °C}$ unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
E_D	differential linearity error		-	± 1	-	LSB
$E_{L(adj)}$	integral non-linearity		-	± 1.5	-	LSB
E_O	offset error		-	0.6	-	%
E_G	gain error		-	0.6	-	%
C_L	load capacitance		-	200	-	pF
R_L	load resistance		1	-	-	kΩ

14. Application information

14.1 Suggested USB interface solutions

Remark: The USB controller is available as a device/Host/OTG controller on parts LPC1769/68/66/65 and as device-only controller on part LPC1764.

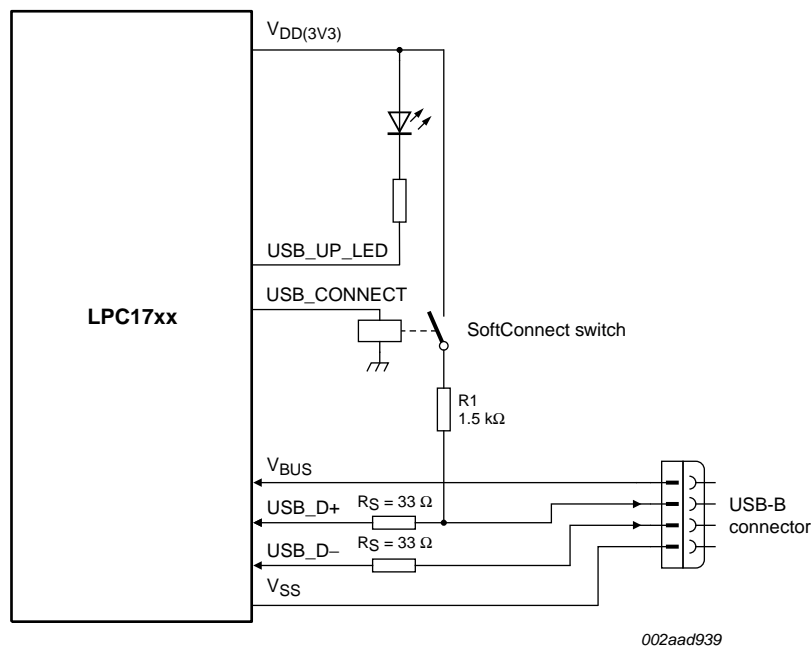


Fig 28. USB interface on a self-powered device

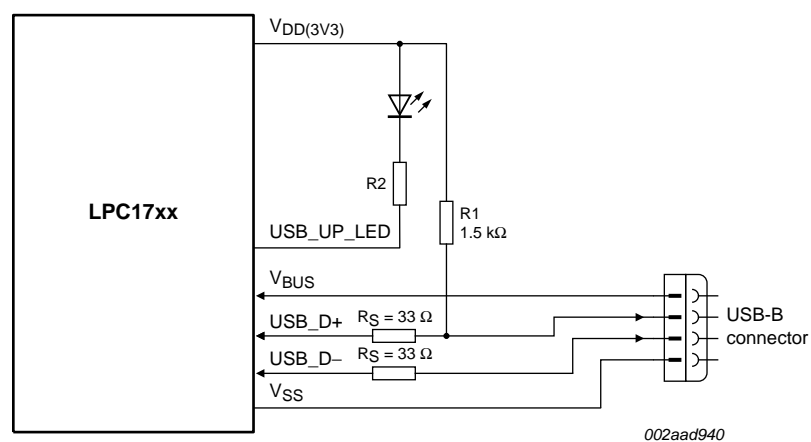


Fig 29. USB interface on a bus-powered device

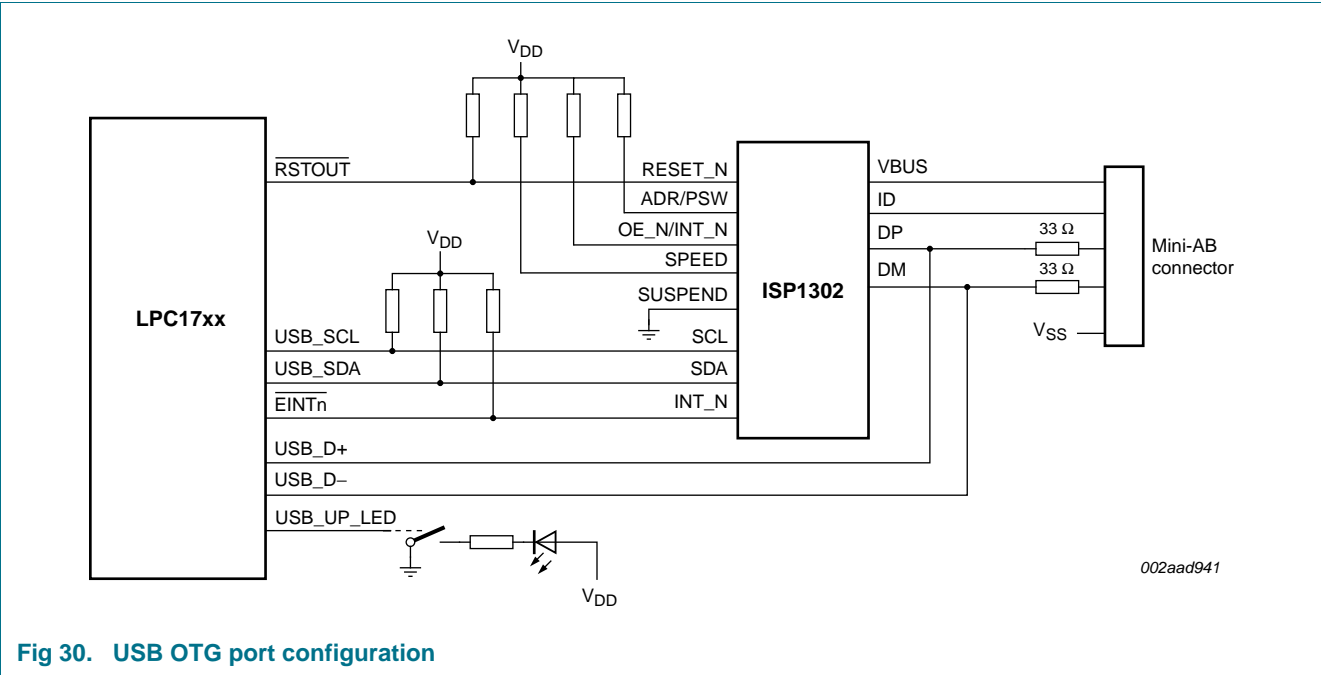


Fig 30. USB OTG port configuration

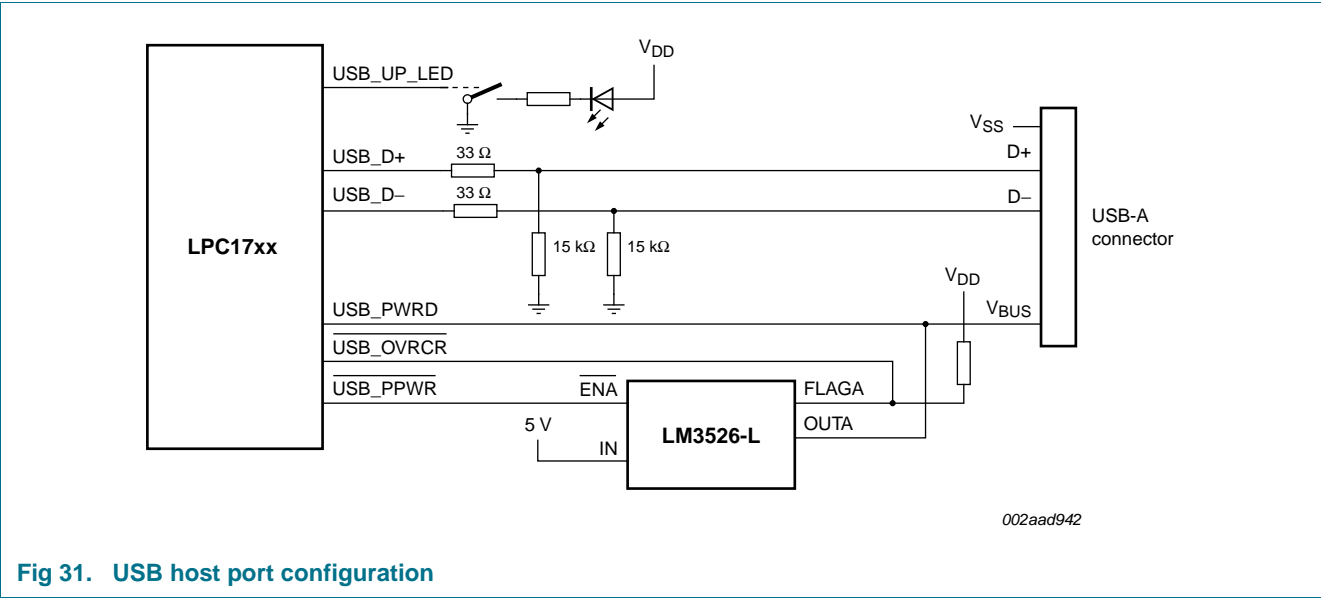
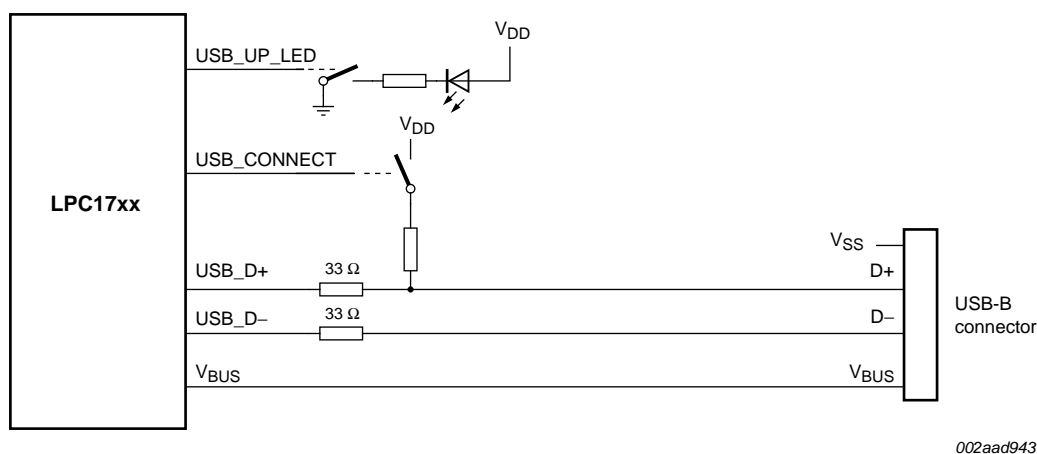


Fig 31. USB host port configuration

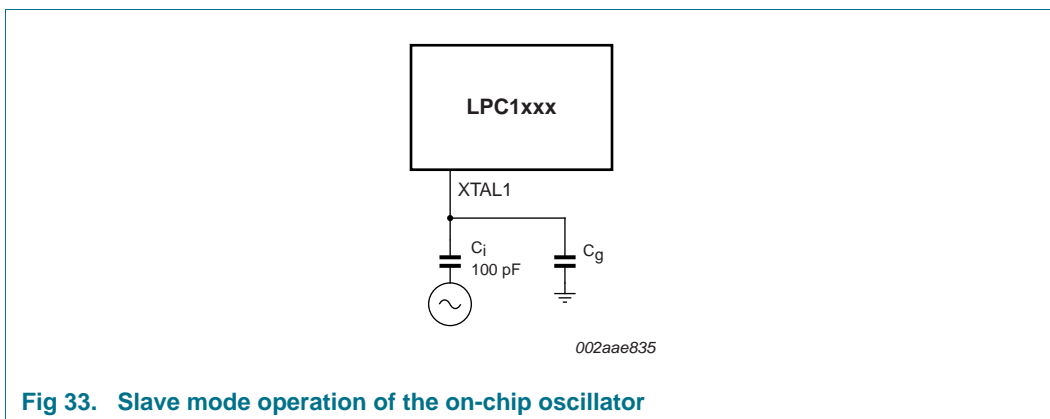


002aad943

Fig 32. USB device port configuration

14.2 Crystal oscillator XTAL input and component selection

The input voltage to the on-chip oscillators is limited to 1.8 V. If the oscillator is driven by a clock in slave mode, it is recommended that the input be coupled through a capacitor with $C_i = 100$ pF. To limit the input voltage to the specified range, choose an additional capacitor to ground C_g which attenuates the input voltage by a factor $C_i/(C_i + C_g)$. In slave mode, a minimum of 200 mV(RMS) is needed.



002aae835

Fig 33. Slave mode operation of the on-chip oscillator

In slave mode the input clock signal should be coupled by means of a capacitor of 100 pF (Figure 33), with an amplitude between 200 mV(RMS) and 1000 mV(RMS). This corresponds to a square wave signal with a signal swing of between 280 mV and 1.4 V. The XTALOUT pin in this configuration can be left unconnected.

External components and models used in oscillation mode are shown in Figure 34 and in Table 22 and Table 23. Since the feedback resistance is integrated on chip, only a crystal and the capacitances C_{X1} and C_{X2} need to be connected externally in case of fundamental mode oscillation (the fundamental frequency is represented by L , C_L and R_S). Capacitance C_P in Figure 34 represents the parallel package capacitance and should not be larger than 7 pF. Parameters F_{OSC} , C_L , R_S and C_P are supplied by the crystal manufacturer.

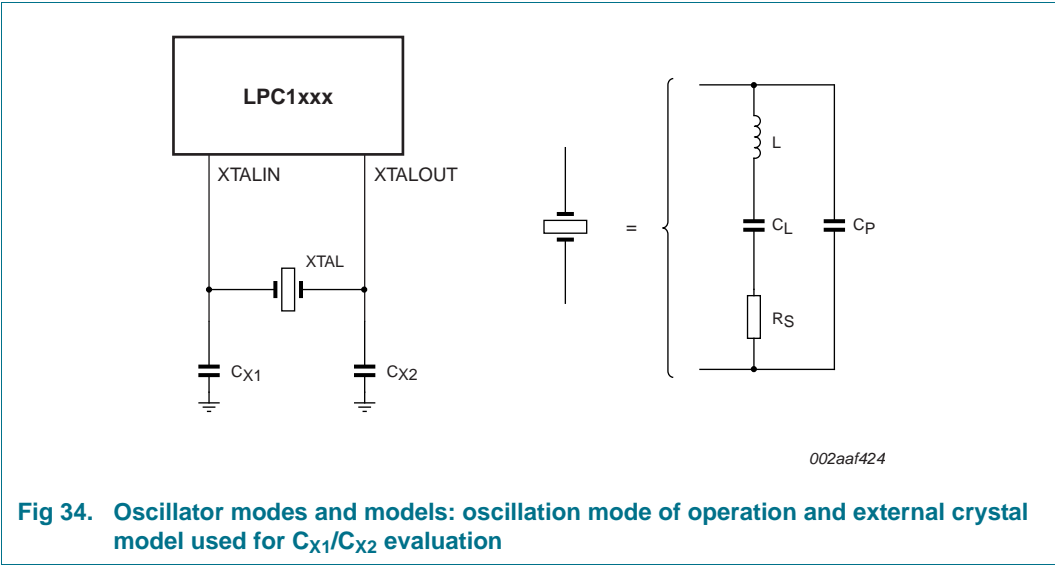


Table 22. Recommended values for C_{x1}/C_{x2} in oscillation mode (crystal and external components parameters): low frequency mode

Fundamental oscillation frequency F _{osc}	Crystal load capacitance C _L	Maximum crystal series resistance R _S	External load capacitors C _{x1} /C _{x2}
1 MHz - 5 MHz	10 pF	< 300 Ω	18 pF, 18 pF
	20 pF	< 300 Ω	39 pF, 39 pF
	30 pF	< 300 Ω	57 pF, 57 pF
5 MHz - 10 MHz	10 pF	< 300 Ω	18 pF, 18 pF
	20 pF	< 200 Ω	39 pF, 39 pF
	30 pF	< 100 Ω	57 pF, 57 pF
10 MHz - 15 MHz	10 pF	< 160 Ω	18 pF, 18 pF
	20 pF	< 60 Ω	39 pF, 39 pF
15 MHz - 20 MHz	10 pF	< 80 Ω	18 pF, 18 pF

Table 23. Recommended values for C_{x1}/C_{x2} in oscillation mode (crystal and external components parameters): high frequency mode

Fundamental oscillation frequency F _{osc}	Crystal load capacitance C _L	Maximum crystal series resistance R _S	External load capacitors C _{x1} , C _{x2}
15 MHz - 20 MHz	10 pF	< 180 Ω	18 pF, 18 pF
	20 pF	< 100 Ω	39 pF, 39 pF
20 MHz - 25 MHz	10 pF	< 160 Ω	18 pF, 18 pF
	20 pF	< 80 Ω	39 pF, 39 pF

14.3 XTAL and RTCX Printed Circuit Board (PCB) layout guidelines

The crystal should be connected on the PCB as close as possible to the oscillator input and output pins of the chip. Take care that the load capacitors C_{x1}, C_{x2}, and C_{x3} in case of third overtone crystal usage have a common ground plane. The external components must also be connected to the ground plain. Loops must be made as small as possible in

order to keep the noise coupled in via the PCB as small as possible. Also parasitics should stay as small as possible. Values of C_{x1} and C_{x2} should be chosen smaller accordingly to the increase in parasitics of the PCB layout.

14.4 Standard I/O pin configuration

Figure 35 shows the possible pin modes for standard I/O pins with analog input function:

- Digital output driver: Open-drain mode enabled/disabled
- Digital input: Pull-up enabled/disabled
- Digital input: Pull-down enabled/disabled
- Digital input: Repeater mode enabled/disabled
- Analog input

The default configuration for standard I/O pins is input with pull-up enabled. The weak MOS devices provide a drive capability equivalent to pull-up and pull-down resistors.

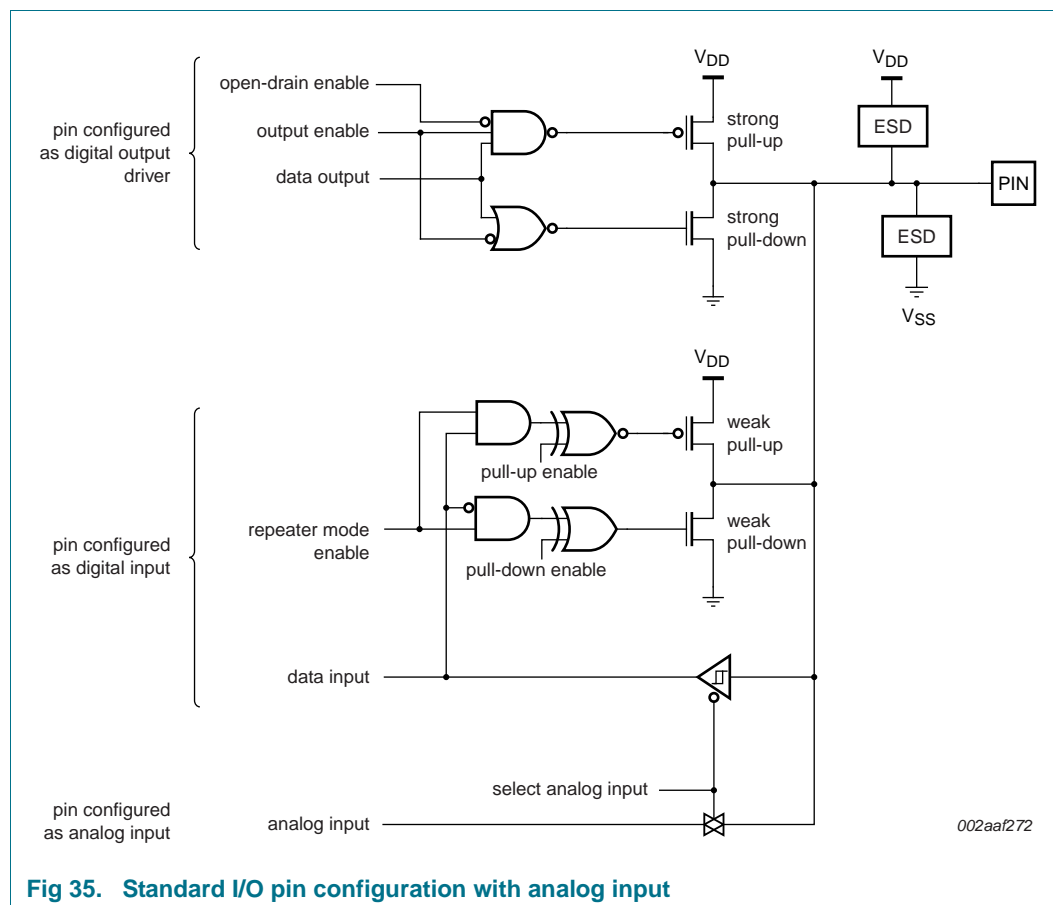
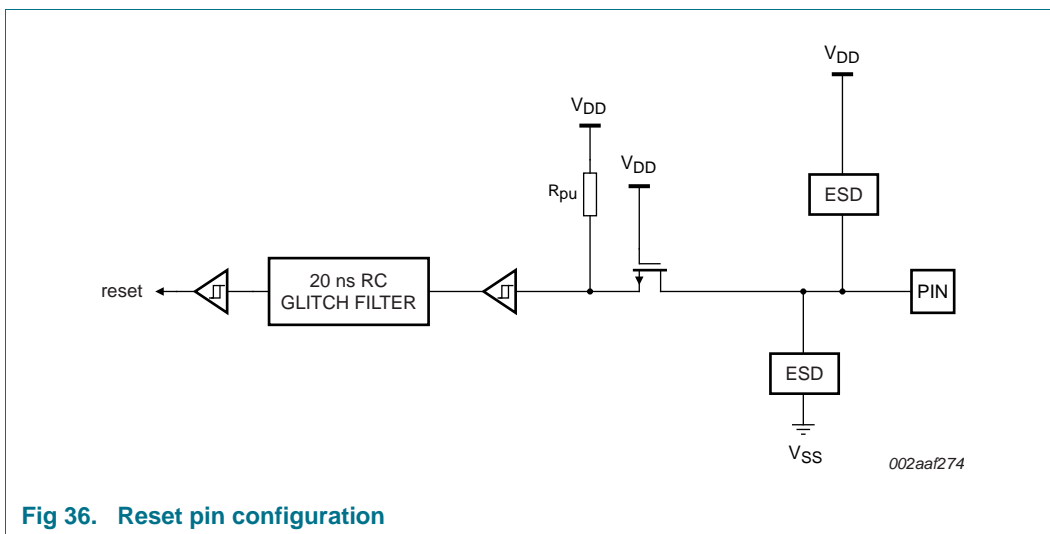


Fig 35. Standard I/O pin configuration with analog input

14.5 Reset pin configuration



14.6 ElectroMagnetic Compatibility (EMC)

Radiated emission measurements according to the IEC61967-2 standard using the TEM-cell method are shown for part LPC1768.

Table 24. ElectroMagnetic Compatibility (EMC) for part LPC1768 (TEM-cell method)

$V_{DD} = 3.3\text{ V}$; $T_{amb} = 25\text{ °C}$.

Parameter	Frequency band	System clock =					Unit
		12 MHz	24 MHz	48 MHz	72 MHz	100 MHz	
Input clock: IRC (4 MHz)							
maximum peak level	150 kHz - 30 MHz	−7	−6	−4	−7	−7	dBμV
	30 MHz - 150 MHz	+1	+5	+11	+16	+9	dBμV
	150 MHz - 1 GHz	−2	+4	+11	+12	+19	dBμV
IEC level ^[1]	-	O	O	N	M	L	-
Input clock: crystal oscillator (12 MHz)							
maximum peak level	150 kHz - 30 MHz	−5	−4	−4	−7	−8	dBμV
	30 MHz - 150 MHz	−1	+5	+10	+15	+7	dBμV
	150 MHz - 1 GHz	−1	+6	+11	+10	+16	dBμV
IEC level ^[1]	-	O	O	N	M	M	-

[1] IEC levels refer to Appendix D in the IEC61967-2 Specification.

15. Package outline

LQFP100: plastic low profile quad flat package; 100 leads; body 14 x 14 x 1.4 mmSOT407-1

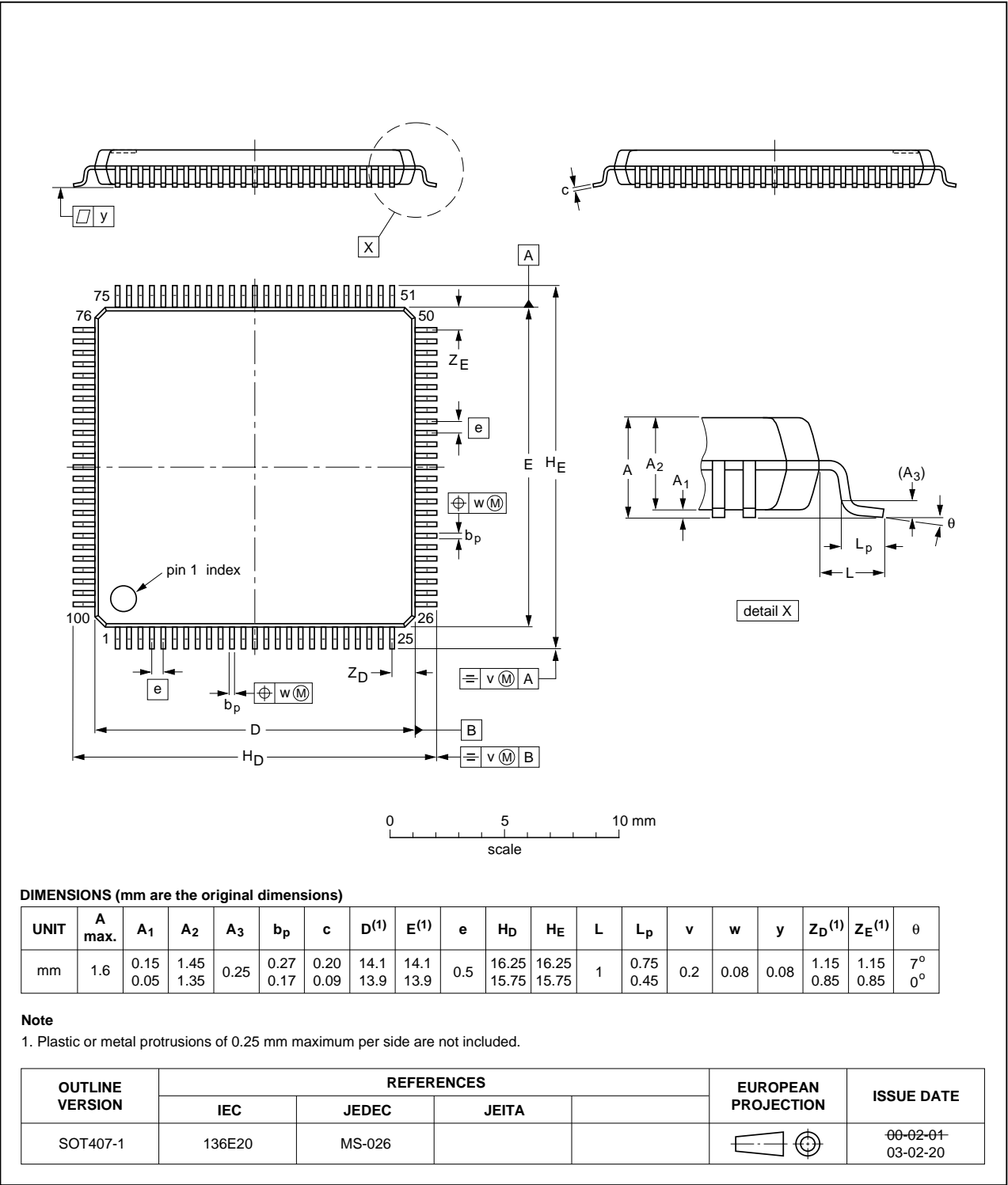


Fig 37. Package outline SOT407-1 (LQFP100)

TFBGA100: plastic thin fine-pitch ball grid array package; 100 balls; body 9 x 9 x 0.7 mm

SOT926-1

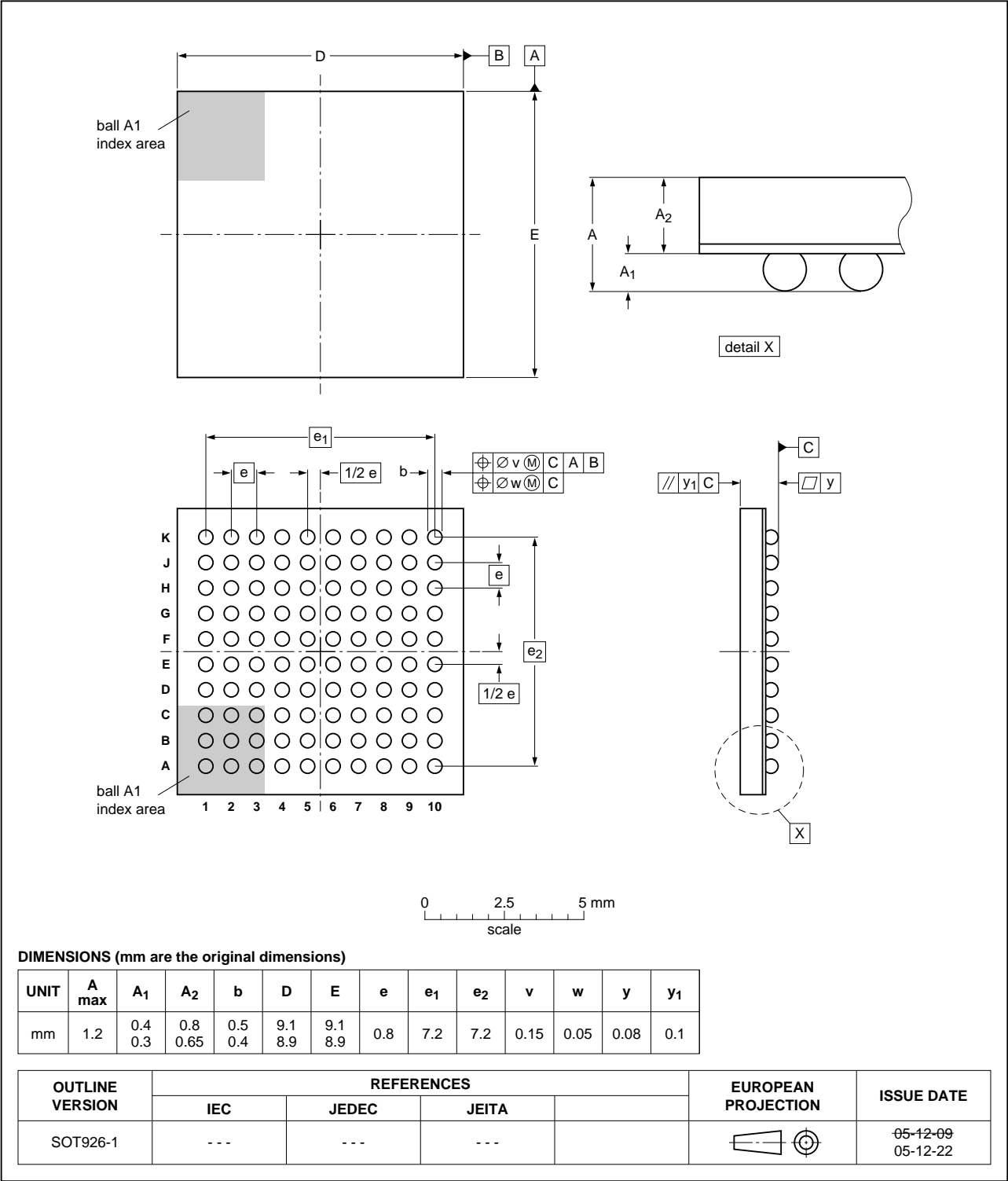


Fig 38. Package outline SOT926-1 (TFBGA100)

16. Soldering

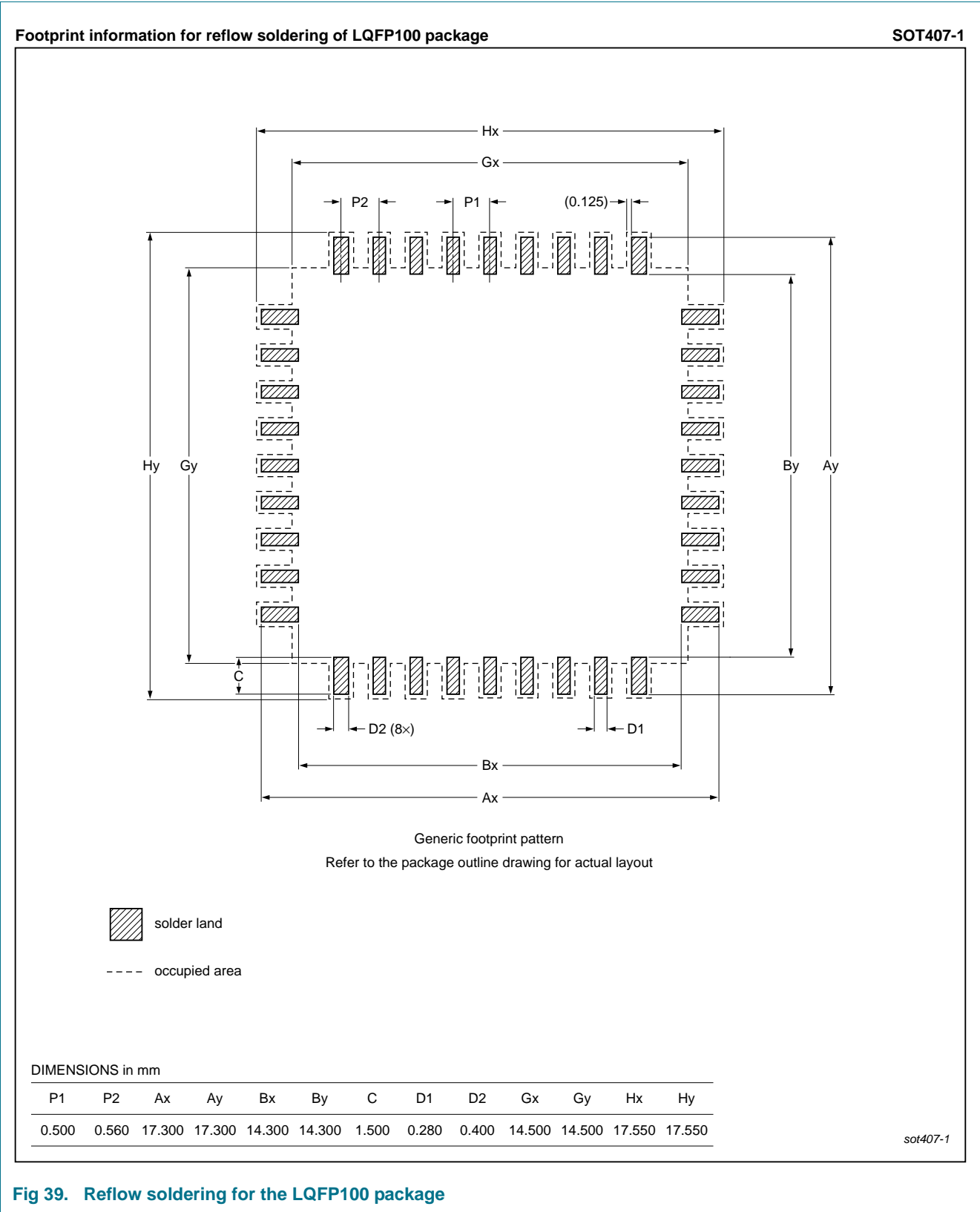
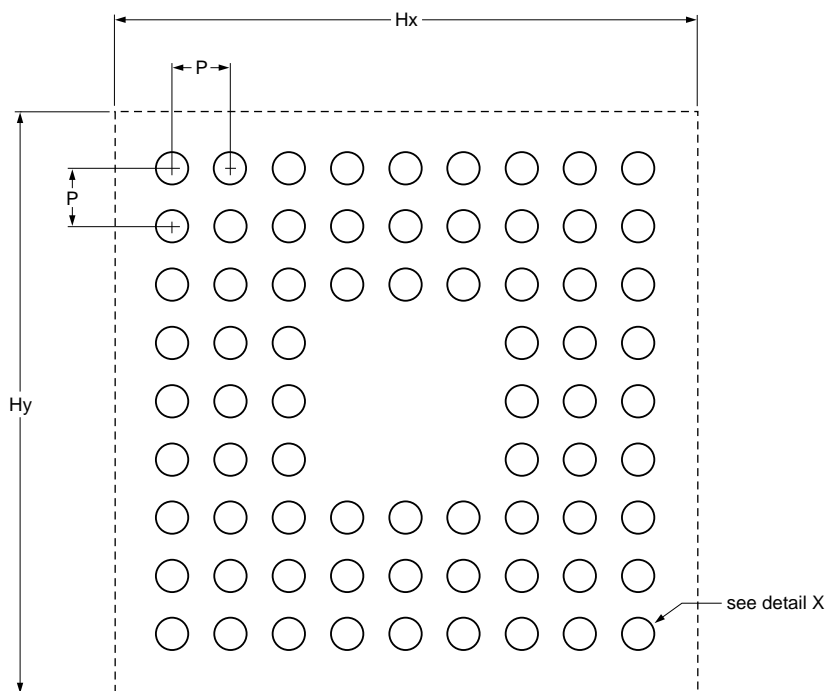


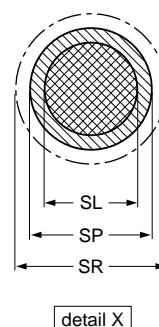
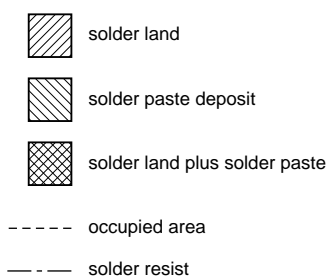
Fig 39. Reflow soldering for the LQFP100 package

Footprint information for reflow soldering of TFBGA100 package

SOT926-1



Generic footprint pattern
Refer to the package outline drawing for actual layout



DIMENSIONS in mm

P	SL	SP	SR	Hx	Hy
0.80	0.330	0.400	0.480	9.400	9.400

sot926-1 fr

Fig 40. Reflow soldering of the TFBGA100 package

17. Abbreviations

Table 25. Abbreviations

Acronym	Description
ADC	Analog-to-Digital Converter
AHB	Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
BOD	BrownOut Detection
CAN	Controller Area Network
DAC	Digital-to-Analog Converter
DMA	Direct Memory Access
EOP	End Of Packet
GPIO	General Purpose Input/Output
IRC	Internal RC
IrDA	Infrared Data Association
JTAG	Joint Test Action Group
MAC	Media Access Control
MIIM	Media Independent Interface Management
OHCI	Open Host Controller Interface
OTG	On-The-Go
PHY	Physical Layer
PLL	Phase-Locked Loop
PWM	Pulse Width Modulator
RIT	Repetitive Interrupt Timer
RMII	Reduced Media Independent Interface
SE0	Single Ended Zero
SPI	Serial Peripheral Interface
SSI	Serial Synchronous Interface
SSP	Synchronous Serial Port
TCM	Tightly Coupled Memory
TTL	Transistor-Transistor Logic
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

18. Revision history

Table 26. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
LPC1769_68_67_66_65_64_63 v.9	20120810	Product data sheet	-	LPC1769_68_67_66_65_64 v.8
Modifications:	<ul style="list-style-type: none"> Remove table note "The peak current is limited to 25 times the corresponding maximum current." from Table 5 "Limiting values". Change $V_{DD(3V3)}$ to $V_{DD(REG)(3V3)}$ in Section 11.3 "Internal oscillators". Glitch filter constant changed to 10 ns in Table note 6 in Table 4. Description of $\overline{\text{RESET}}$ function updated in Table 4. Pull-up value added for GPIO pins in Table 4. Pin configuration diagram for LQFP100 package corrected (Figure 2). 			
LPC1769_68_67_66_65_64_63 v.8	20111114	Product data sheet	-	LPC1769_68_67_66_65_64 v.7
Modifications:	<ul style="list-style-type: none"> Pin description of USB_UP_LED pin updated in Table 4. R_{i1} and R_{i2} labels in Figure 27 updated. Part LPC1765FET100 added. Table note 10 updated in Table 4. Table note 1 updated in Table 12. Pin description of STCLK pin updated in Table 4. Electromagnetic compatibility data added in Section 14.6. Section 16 added. 			
LPC1769_68_67_66_65_64_63 v.7	20110405	Product data sheet	-	LPC1769_68_67_66_65_64 v.6
Modifications:	<ul style="list-style-type: none"> Pin description of pins P0[29] and P0[30] updated in Table note 5 of Table 4. Pins are not 5 V tolerant. Typical value for Parameter N_{endu} added in Table 9. Parameter V_{hys} for I²C bus pins: typical value corrected $V_{\text{hys}} = 0.05V_{DD(3V3)}$ in Table 7. Condition $3.0\text{ V} \leq V_{DD(3V3)} \leq 3.6\text{ V}$ added in Table 16. Typical values for parameters $I_{DD(REG)(3V3)}$ and I_{BAT} with condition Deep power-down mode corrected in Table 7 and Table note 9, Table note 10, and Table note 11 updated. For Deep power-down mode, Figure 9 updated and Figure 10 added. 			
LPC1769_68_67_66_65_64_63 v.6	20100825	Product data sheet	-	LPC1769_68_67_66_65_64 v.5
Modifications:	<ul style="list-style-type: none"> Part LPC1768TFBGA added. Section 7.30.2; BOD level corrected. Added Section 10.2. 			
LPC1769_68_67_66_65_64_63 v.5	20100716	Product data sheet	-	LPC1769_68_67_66_65_64 v.4
LPC1769_68_67_66_65_64 v.4	20100201	Product data sheet	-	LPC1768_67_66_65_64 v.3
LPC1768_67_66_65_64 v.3	20091119	Product data sheet	-	LPC1768_66_65_64 v.2
LPC1768_66_65_64 v.2	20090211	Objective data sheet	-	LPC1768_66_65_64 v.1
LPC1768_66_65_64 v.1	20090115	Objective data sheet	-	-

19. Legal information

19.1 Data sheet status

Document status ^{[1][2]}	Product status ^[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term 'short data sheet' is explained in section "Definitions".

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL <http://www.nxp.com>.

19.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

Product specification — The information and data provided in a Product data sheet shall define the specification of the product as agreed between NXP Semiconductors and its customer, unless NXP Semiconductors and customer have explicitly agreed otherwise in writing. In no event however, shall an agreement be valid in which the NXP Semiconductors product is deemed to offer functions and qualities beyond those described in the Product data sheet.

19.3 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <http://www.nxp.com/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b)

whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

19.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I²C-bus — logo is a trademark of NXP B.V.

20. Contact information

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

21. Contents

1	General description	1	7.20	I ² S-bus serial I/O controllers	29
2	Features and benefits	1	7.20.1	Features	29
3	Applications	4	7.21	General purpose 32-bit timers/external event counters	29
4	Ordering information	4	7.21.1	Features	29
4.1	Ordering options	4	7.22	Pulse width modulator	30
5	Block diagram	5	7.22.1	Features	30
6	Pinning information	6	7.23	Motor control PWM	31
6.1	Pinning	6	7.24	Quadrature Encoder Interface (QEI)	31
6.2	Pin description	8	7.24.1	Features	31
7	Functional description	18	7.25	Repetitive Interrupt (RI) timer	32
7.1	Architectural overview	18	7.25.1	Features	32
7.2	ARM Cortex-M3 processor	18	7.26	ARM Cortex-M3 system tick timer	32
7.3	On-chip flash program memory	18	7.27	Watchdog timer	32
7.4	On-chip SRAM	18	7.27.1	Features	32
7.5	Memory Protection Unit (MPU)	18	7.28	RTC and backup registers	33
7.6	Memory map	19	7.28.1	Features	33
7.7	Nested Vectored Interrupt Controller (NVIC)	21	7.29	Clocking and power control	33
7.7.1	Features	21	7.29.1	Crystal oscillators	33
7.7.2	Interrupt sources	21	7.29.1.1	Internal RC oscillator	34
7.8	Pin connect block	21	7.29.1.2	Main oscillator	34
7.9	General purpose DMA controller	21	7.29.1.3	RTC oscillator	34
7.9.1	Features	22	7.29.2	Main PLL (PLL0)	35
7.10	Fast general purpose parallel I/O	22	7.29.3	USB PLL (PLL1)	35
7.10.1	Features	23	7.29.4	RTC clock output	35
7.11	Ethernet	23	7.29.5	Wake-up timer	35
7.11.1	Features	23	7.29.6	Power control	36
7.12	USB interface	24	7.29.6.1	Sleep mode	36
7.12.1	USB device controller	24	7.29.6.2	Deep-sleep mode	36
7.12.1.1	Features	24	7.29.6.3	Power-down mode	37
7.12.2	USB host controller	25	7.29.6.4	Deep power-down mode	37
7.12.2.1	Features	25	7.29.6.5	Wake-up interrupt controller	37
7.12.3	USB OTG controller	25	7.29.7	Peripheral power control	37
7.12.3.1	Features	25	7.29.8	Power domains	38
7.13	CAN controller and acceptance filters	25	7.30	System control	39
7.13.1	Features	26	7.30.1	Reset	39
7.14	12-bit ADC	26	7.30.2	Brownout detection	40
7.14.1	Features	26	7.30.3	Code security (Code Read Protection - CRP)	40
7.15	10-bit DAC	26	7.30.4	APB interface	40
7.15.1	Features	26	7.30.5	AHB multilayer matrix	41
7.16	UARTs	27	7.30.6	External interrupt inputs	41
7.16.1	Features	27	7.30.7	Memory mapping control	41
7.17	SPI serial I/O controller	27	7.31	Emulation and debugging	41
7.17.1	Features	27	8	Limiting values	42
7.18	SSP serial I/O controller	27	9	Thermal characteristics	43
7.18.1	Features	28	10	Static characteristics	44
7.19	I ² C-bus serial I/O controllers	28	10.1	Power consumption	47
7.19.1	Features	28			

continued >>

10.2	Peripheral power consumption	50
10.3	Electrical pin characteristics	51
11	Dynamic characteristics	53
11.1	Flash memory	53
11.2	External clock	53
11.3	Internal oscillators	54
11.4	I/O pins	54
11.5	I ² C-bus	55
11.6	I ² S-bus interface	56
11.7	SSP interface	58
11.8	USB interface	59
11.9	SPI	60
12	ADC electrical characteristics	62
13	DAC electrical characteristics	65
14	Application information	66
14.1	Suggested USB interface solutions	66
14.2	Crystal oscillator XTAL input and component selection	68
14.3	XTAL and RTCX Printed Circuit Board (PCB) layout guidelines	69
14.4	Standard I/O pin configuration	70
14.5	Reset pin configuration	71
14.6	ElectroMagnetic Compatibility (EMC)	72
15	Package outline	73
16	Soldering	75
17	Abbreviations	77
18	Revision history	78
19	Legal information	79
19.1	Data sheet status	79
19.2	Definitions	79
19.3	Disclaimers	79
19.4	Trademarks	80
20	Contact information	80
21	Contents	81

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© NXP B.V. 2012.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 10 August 2012

Document identifier: LPC1769_68_67_66_65_64_63

