Aquarius Custos

Grado de Ingeniería Informática

Estudiante
Juan Enrique Gómez Pérez

Consultor
Jordi Bécares Ferrés

16 de Enero de 2016

Esta obra está sujeta a una licencia de
Reconocimiento-NoComercial-SinObraDerivada 3.0
España (CC BY-NC-ND 3.0 ES)

FITXA DEL TREBALL FINAL

Título del trabajo:	*Aquarius Custos*
Nombre del autor:	*Juan Enrique Gómez Pérez*
Nombre del consultor:	*Jordi Bécares Ferrés*
Fecha de entrega (mm/aaaa):	*01/2016*
Área del trabajo final:	*Sistemas encastrados*
Titulación:	*Grado de Ingeniería Informática*

Abstract (in English, 250 words or less):

Today, hobbies are a fundamental part of our society and play a very important role in our personal wellness. Hobbies like taking care of an aquarium can be very time consuming in some circumstances and it requires a constant surveillance in any circumstance.

This is why managing an aquarium can be very satisfactory for the owner, but maintaining it in good condition every day during a full year generates some problems in periods like holidays or weekends. This job looks for creating a solution to this need, automating the maintenance tasks needed for this hobby. This solution will provide a useful system under any circumstance, in the daily maintenance tasks and when the owner is far away from it.

To achieve this objective, the aquarium community has defined a number of elements to be maintained on which we will base this solution. The system will develop tasks to execute maintenance tasks on these elements using a real time system. These tasks will monitor water temperature and pH, environmental lighting and will manage a feeder. The solution will execute all these tasks periodically. It will allow the user to program how frequently these tasks must be executed, and which actions to execute just in case some of the predefined parameters for each element are out of the defined thresholds.

The system will also allow the end user to get real-time information from the elements of the aquarium.

This solution will offer an end to end system which will care for the health on the aquarium, when we are near or far away from it.

Palabras clave (entre 4 y 8):
embebido lpc1769 acuario sensores pH temperatura luz

A mi familia por todo el tiempo robado y
aun así contar con todo su apoyo y paciencia.
Y a mi madre, que siempre nos enseñó
que lo que nos propusiéramos es posible.
Gracias.

Agradecimientos

En primer lugar a mi familia por la paciencia que han tenido conmigo durante todas esas horas dedicadas a estudiar, a elaborar PACs o prácticas, a aguantarme en los malos momentos o compartir los buenos resultados.

En segundo lugar a la UOC por el modelo educativo que ha implantado, el cual sin duda ha hecho que pueda alcanzar esta fase final del Grado. La forma de la evaluación continua, la claridad de los contenidos, la accesibilidad de los consultores, el campus, y así un largo etc. ha hecho que disfrute todos estos años de estudio y francamente mantenga el gusanillo para el futuro.

En tercer lugar a los consultores, los cuales ha sabido transmitir en muchas ocasiones la pasión por la asignatura, que se han preocupado por entender nuestros problemas, y no únicamente del resultado académico, que han sido comprensivos y nos han apoyado en momentos difíciles.

Por último a todos los compañeros estudiantes de la UOC con los que he compartido momentos excepcionales durante todos estos años, nos hemos apoyado, hemos disfrutado y hemos sufrido. Hemos podido sentir un auténtico espíritu universitario.

Gràcies a la gran comunitat de la UOC.

Índice

1. Introducción ... 11
1.1 Justificación ... 11
1.2 Descripción del proyecto ... 12
1.3 Objetivos ... 12
1.4 Enfoque y método seguido ... 13
1.5 Planificación ... 14
1.6 Recursos empleados ... 18
1.7 Productos obtenidos ... 22
1.8 Descripción de capítulos ... 22
2. Antecedentes ... 24
2.1 Estado del arte ... 24
2.2 Estudio de mercado ... 25
3. Descripción funcional ... 28
3.1 Sistema principal ... 28
3.2 Interface de usuario ... 31
3.3 Aplicación del sistema encastrado LPC1769 ... 35
4. Descripción detallada ... 37
4.1 Descripción del Hardware ... 37
4.1.1 Placa LPC1769 ... 37
4.1.2 Conectividad puerto Serie: librerías printf y log, driver uart. ... 39
4.1.3 Sensores analógicos ... 41
4.1.4 Sensor digital ... 44
4.1.5 Relés ... 47
4.2 Software del sistema ... 49
4.2.1 Sistema operativo ... 49
4.2.2 Aplicación del sistema: ... 49
5. Viabilidad técnica ... 56
6. Valoración económica ... 58
7. Conclusiones ... 60
7.1 Conclusiones ... 60
7.2 Propuesta de mejoras ... 61
7.3 Autoevaluación ... 61

8. Glosario 62
9. Bibliografía 64
10. Anexos 65
10.1 Manual de usuario 65
10.1.1 Conexión de dispositivos 65
10.1.2 Uso de la aplicación 66
10.2 Ejecución y compilación 67
10.2.1 Entorno de desarrollo y compilación 67

Índice de figuras

Ilustración 1 – Planificación inicial 16
Ilustración 2 - Planificación final 17
Ilustración 3 - IDE de desarrollo 18
Ilustración 4 - Placa LPC1769 18
Ilustración 5 - Módulo WiFly RX-XV 19
Ilustración 6 – Adaptador DIP xBee 19
Ilustración 7 - Módulo CP2102 19
Ilustración 8 - Relé JQC-3FF 19
Ilustración 9 - Sensor temperatura DF18S20 digital 20
Ilustración 10 - Sonda analógica de pH SEN0161 20
Ilustración 11 - Conversor lógico de nivel BSS138 20
Ilustración 12 - Mini foto sensor 21
Ilustración 13 - Dispensador automático de comida 21
Ilustración 14 - Breadboard para conexiónado del prototipo 21
Ilustración 15 - Diagrama de bloques LPC17xx 24
Ilustración 16 - Mercados de mayor crecimiento hasta 2020 26
Ilustración 17 - Solución para acuario Aquatrónica 27
Ilustración 18 - Diagrama del sistema completo 28
Ilustración 19 - Interacción de elementos 29
Ilustración 20 - Diagrama de bloques 30
Ilustración 21 - Interface de usuario 32
Ilustración 22 - Alarmas en interface de usuario 33
Ilustración 23 - Diagrama de bloques Interface de usuario 34
Ilustración 24 - Diagrama de bloques aplicación del sistema encastrado 35

Ilustración 25 - Diagrama de bloques de la placa LPC1769 37
Ilustración 26 - Esquema general del sistema 39
Ilustración 27 - Conversor USB-Serie 40
Ilustración 28 - Conexión conversores serie-USB 40
Ilustración 29 - Mini fotosensor 42
Ilustración 30 - Divisor de tensión 5v a 3,3v 42
Ilustración 31 - Sonda de pH 43
Ilustración 32 - Sensor de temeratura 1-wire 45
Ilustración 33 - Inicializacion bus DS18S20 45
Ilustración 34 - protocolo de tiempo para envio y recepcion 46
Ilustración 35 - Características rele JQC-3FF 47
Ilustración 36 - Esquema conexión Relé 48
Ilustración 37 - Diagrama de bloques aplicación 50
Ilustración 38 - Diagrama de flujo tareas 51
Ilustración 39 - menu de configuración 52
Ilustración 40 - Mensajes de alarma 54
Ilustración 41 - PINOut de la placa LPCXpresso 65
Ilustración 42 - Esquema de conexión CP2102 66
Ilustración 43 - Importar proyectos en LPCXpresso 68
Ilustración 44 - Selección de proyectos 68
Ilustración 45 - Proyectos disponibles en el IDE 69
Ilustración 46 - Compilación del proyecto 69

Tablas

Tabla 1 - Fabricantes sistemas embebidos 26

Tabla 2 - Correspondencias mV a pH 43

Tabla 3 - Costes de materiales 58

Tabla 4 - Costes mano de obra 59

Tabla 5 - Periféricos soportados 65

1. Introducción

1.1 Justificación

La vida actual, en especial de las personas que viven en ciudades, es por regla general muy exigente y por tanto estresante. Las diferentes obligaciones que nuestro modelo de vida nos impone: el trabajo, los hijos, la compra, etc. demanda una fuerte atención y dedicación durante la semana sin dejarnos apenas tiempo libre para nuestra auto-realización y en un entorno como el descrito, disponer de tiempo libre para dedicar a nosotros mismos se convierte cada vez en un objetivo más importante.

Este tiempo libre se puede dedicar a muchas cosas, pero particularmente a la realización de hobbies que nos satisfagan y produzcan un bienestar personal. El cuidado de animales desde tiempos inmemoriales ha sido bien un trabajo o bien un hobby dependiendo del tipo de animal y objetivo de este. Hoy en día un hobby muy extendido y de una cierta complejidad es la creación, cuidado y mantenimiento de acuarios particulares.

Cuidar un acuario es una actividad muy satisfactoria, pero a la vez muy exigente, debido a que al final nuestro hobby está formado por seres vivos, plantas, peces, crustáceos, etc. lo que hace que sea un hobby que no podemos guardar en una caja y olvidarnos de él hasta que nos vuelva a llamar la atención. Como hobby que nos crea esta obligación diaria de trabajar en su mantenimiento, y no siendo un hobby que nos podamos guardar y llevar con nosotros surge una necesidad importante y es la de analizar cómo podemos mantener correctamente este acuario en momentos en los que no podemos dedicarle tiempo al hobby, viajes de trabajo, periodos de vacaciones, o simplemente un fin de semana fuera de casa.

En este contexto aparece *Aquarius Custos*. *Aquarius Custos* se basará un sistema informático que se responsabilizará de las actividades principales de mantenimiento del acuario, permitiéndonos de esta manera delegar este cuidado, al menos en un formato de mínimos al sistema, el cual podrá tomar decisiones sobre el acuario en función de diversos sensores y configuraciones. De esta manera durante cortos periodos de tiempo *Aquarios Custos* será el responsable del cuidado del acuario, e incluso durante los periodos que el acuario es atendido regularmente ofrecerá valiosa información al usuario facilitándole de esta manera su tarea diaria de mantenimiento.

1.2 Descripción del proyecto

Se construirá un sistema basado en un sistema embebido. Este sistema será capaz de leer diferentes parámetros del medioambiente del acuario a través de varios dispositivos como: sensor de temperatura del agua, sensor de pH y sensor de luz ambiental. Además en función de estos valores el sistema será capaz de tomar algunas decisiones o alertar al usuario del acuario, por ejemplo cuando los valores de temperatura del acuario o del pH se encuentren fuera de los umbrales definidos se lanzará una alarma, o cuando la luz ambiental durante una franja horaria predeterminada esté por debajo de un determinado umbral se realizará el encendido de una luz artificial. Por último el sistema en base a una programación introducida por el usuario será capaz de realizar el proceso de alimentación del acuario.

El sistema tendrá un interface de usuario en el que mostrará un cuadro de mando del estado del acuario, los umbrales definidos y las últimas acciones del mismo. De este modo el usuario podrá de forma sencilla interactuar con el sistema y ajustarlo a sus necesidades.

1.3 Objetivos

- Desarrollar un sistema que mida la temperatura del agua del acuario de forma continua, vigilando que esté dentro de unos umbrales previamente establecidos, y alerte en caso que estos sean sobrepasados por exceso o defecto.
- Medir la acidez del agua (pH) vigilando que se encuentre dentro de unos rangos compatibles con la vida de los animales y plantas que viven en el acuario, alertando si el pH disminuye o supera los umbrales establecidos.
- Simular el ciclo día/noche en los horarios correspondientes de manera artificial.
- Provisión de alimentos automatizada de los animales del acuario. Definiendo las horas del día en que se realizará la alimentación, hasta 10 diferentes, y en cada una de ellas la cantidad de alimentación que se dispensará.
- El usuario podrá configurar los umbrales de los sensores, vía menú disponible en el puerto serie. También se podrán configurar las notificaciones que se desean recibir.
- Control del ciclo de día/noche y la alimentación, personalización de los umbrales y alertas.

- Envío y consulta remota de las estadísticas del acuario: temperaturas históricas, evolución del pH, ciclos de día/noche registrando la intensidad de luz y alimentación dispensada.

1.4 Enfoque y método seguido

El enfoque y método se ha realizado siguiendo el modelo "top-down" utilizado en el desarrollo de software, pero aplicándolo al sistema completo. En el modelo top-down en primer lugar se diseña un modelo objetivo de este a alto nivel, para posteriormente ir profundizando en el mismo. En este caso definimos un sistema para el cuidado y mantenimiento de acuarios, al que fuimos añadiendo componentes en forma de sensores o funcionalidades, hasta llegar a un sistema objetivo final sobre el que comenzamos los desarrollo y realizamos las planificaciones.

Este enfoque era importante, ya que el número de componentes que lo forman es bastante heterogéneo en sus características e interfaces, lo que obligó a realizar una definición genérica del sistema, para después profundizar en el conocimiento de los diferentes dispositivos que forman parte del sistema y así poder construir la solución final.

Para realizar esta aproximación comenzamos fijando unos objetivos claros junto con las tareas que acompañaban a estos objetivos. Con estos objetivos fijados se realizó un primer diseño de la solución con varias iteraciones de la misma, para después realizar una cuidadosa selección de los dispositivos que conformarían la solución final.

A partir de este momento comienza el periodo de profundizar en el conocimiento, tanto del microcontrolador que gobernará todo el sistema, como de los diferentes dispositivos, para ello dimos los siguientes pasos:

- Se analizaron todas las características de la placa LPC1769, microcontrolador elegido como sistema principal del proyecto.
- Se instalaron las herramientas necesarias de desarrollo (entorno IDE) del microcontrolador.
- Se realizaron los primeros test con sistemas de prueba del fabricante para validar su correcto funcionamiento.
- A partir de aquí se empezaron a desarrollar pequeños *drivers* o controladores junto con un grupo de librerías para su utilización, de los diferentes dispositivos junto con pequeños programas que probaban su funcionalidad en diferentes escenarios (ejecutándose como tareas en tiempo real, como un programa secuencial, etc.). Estos fueron:

- GPIO: Driver para el manejo de los diferentes puertos GPIO de la placa LPC1769. Este driver contenía las capacidades necesarias para configurar la funcionalidad de estos puertos como puertos digitales, ADC, UART, etc.
- Puerto serie: gestión del puerto serie que incluye el sistema embebido.
- Wifly: Driver para la gestión de la conectividad Wifi de la placa, así como el envío y recepción de comandos HTTP.
- Implementación de un driver con el protocolo 1-Wire para su uso con la sonda de temperatura.
- Driver para el conversor analógico-digital, ADC.
- Librería de control de relés
- El driver 1-wire, se basa en desarrollos de terceros, lo que exigió no el desarrollo del driver, pero si comprender su funcionamiento, y ajustarlo a las necesidades del proyecto.

Además de estos módulos se fueron construyendo pequeñas piezas de software en forma de librerías que ayudaban a mejora tareas repetitivas a lo largo del proyecto, con utilidades como pequeñas demoras, cálculos, o conversiones de datos.

Una vez desarrolladas todas las piezas y probadas una a una de manera individual se definió la forma de comunicación entre ellas, creando las tareas en un único esqueleto de aplicación responsable de la coordinación de estas así como de la comunicación.

Cada uno de los bloques o puntos finales siempre finaliza con un pack de pruebas del mismo, de forma que se pudiera certificar la calidad del mismo así como su funcionamiento de la manera esperada, terminando con una prueba completa del sistema.

1.5 Planificación

La planificación del proyecto se ha definido en 4 grandes bloques: Inicio o fase de investigación Fase 2 que agrupa todas las tareas específicas del proyecto y sus subsistemas, fase 3 está compuesta por unos objetivos secundarios que agregan un valor adicional al proyecto y para terminar la cuarta fase que se ocupa de la documentación y memoria del proyecto.

- **Fase 1**: esta fase comprende la preparación inicial del proyecto, elaboración de la documentación, establecer objetivos y tareas del proyecto, y desarrollar la planificación. También incluye aspectos técnicos como la instalación del IDE de desarrollo, instalación del *framework* y sistema operativo a utilizar y tareas básicas de pruebas sobre estos.
- **Fase 2**: Durante la fase dos se ha realizado el desarrollo de diferentes controladores y librerías básicas del sistema. Estas tareas cubren elementos generales como el módulo WiFi, los puertos series, o los puertos GPIO del sistema.
- **Fase 3**: En la fase tres se pone atención en el desarrollo y gestión de los dispositivos específicos de este proyecto como sensores o relés. Esta fase contiene dos tareas adicionales que finalmente no fue posible llevar a cabo por falta de tiempo, como es el envío de información a Arp@lab y envíom de alarmas vía correo electrónico.
- **Fase 4**: Para terminar la planificación la última fase de esta agrupa todas las actividades de calidad y formato de código, documentación, elaboración de memorias y vídeos de presentación.

La planificación final ha sufrido modificaciones frente a lo inicialmente descrito. Esto ha sido debido a diferentes factores. En primer lugar los diseños electrónicos de conexión entre los dispositivos y la placa LPCXpresso, debido al desconocimiento de electrónica, han consumido mucho tiempo en la investigación y aprendizaje. Esto también ha supuesto modificar la lista de materiales inicial. Este punto es importante ya que el riesgo de que la placa LPCXpresso se viera dañada por una mala conexión era alto.

En segundo lugar la implementación del driver 1-wire consumió prácticamente una semana completa de trabajo. Este driver se obtuvo de un foro del fabricante NXP con buenas referencias y aparentemente buena calidad de código, sin embargo no funcionaba. Esto requirió una dedicación especial a entender el funcionamiento del protocolo 1-wire, para después revisar el código y corregir los problemas que este presentaba y así hacer que funcionara correctamente.

El último aspecto ha sido la estimación de los tiempos necesarios para los desarrollos relacionados con la gestión del puerto serie, los cuales aunque se finalizaron durante la planificación inicial, después ha necesitado de nuevas funcionalidades las cuales han llevado bastante tiempo su implementación.

A continuación se muestran los diagramas de Gantt correspondientes a la planificación inicial (ilustración 1) y a la planificación final o real (ilustración 2).

Planificación inicial:

	Nombre	Duración	Inicio	Fin
1	**Inicio TFG**	**14.38d**	**16/09/2015**	**30/09/2015**
2	Instalación del entorno de desarrollo LPCIDE con FreeRTOS	2.33d	16/09/2015	18/09/2015
3	Analisis de la versión de FreeRTOS a utilizar en desarrollo	2.33d	28/09/2015	30/09/2015
4	**OBJETIVOS BÁSICOS: Fase 1**	**32.63d**	**01/10/2015**	**02/11/2015**
5	Desarrollo del módulo que gestionará la comunicación vía UA	10d	01/10/2015	11/10/2015
6	Desarrollo del módulo que gestionará los componentes GPIO	3d	11/10/2015	14/10/2015
7	Desarrollo del módulo de comunicaciones WiFy (base)	13d	15/10/2015	28/10/2015
8	Librería de utilidades generales (printf / log)	5.63d	28/10/2015	02/11/2015
9	**OBJETIVOS BÁSICOS: Fase 2**	**40d**	**02/11/2015**	**12/12/2015**
10	Sensor de iluminación	7d	02/11/2015	09/11/2015
11	Sensor de temperatura	7d	09/11/2015	16/11/2015
12	Cuerpo principal de tareas de lectura y monitorización	7d	16/11/2015	23/11/2015
13	Módulo menú configuración	5d	23/11/2015	28/11/2015
14	Sensor de pH	3d	28/11/2015	01/12/2015
15	Módulo de alarmas	4d	01/12/2015	05/12/2015
16	Testing	7d	05/12/2015	12/12/2015
17	**OBJETIVOS SECUNDARIOS: Fase 3**	**10d**	**12/12/2015**	**22/12/2015**
18	Manejo del Alimentador	5d	12/12/2015	17/12/2015
19	Envio información a Arp@lab	2d	17/12/2015	19/12/2015
20	Módulo correo electrónico	2d	19/12/2015	21/12/2015
21	Testing	1d	21/12/2015	22/12/2015
22	**OBJETIVOS BASICOS: Documentación**	**53.38d**	**01/12/2015**	**23/01/2016**
23	Previa de la memoria	11d	01/12/2015	12/12/2015
24	Entrega final del código	26d	07/12/2015	02/01/2016
25	Memoria definitiva	25d	22/12/2015	16/01/2016
26	Presentación	7d	16/01/2016	23/01/2016

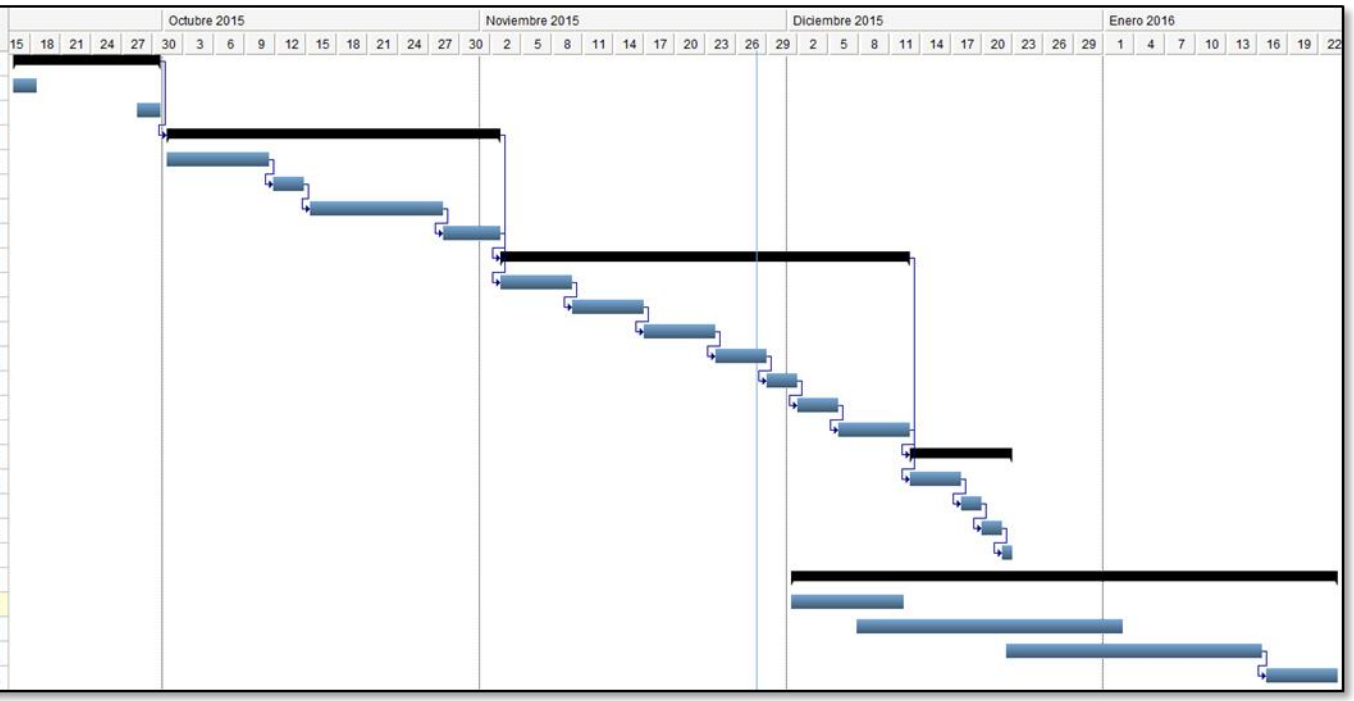


Ilustración 1 – Planificación inicial

Planificación final:

	Nombre	Duración	Inicio	Fin
1	Inicio TFG	14.38d	16/09/2015	30/09/2015
2	Instalación del entorno de desarrollo LPCIDE con FreeRTOS	2.33d	16/09/2015	18/09/2015
3	Analisis de la versión de FreeRTOS a utilizar en desarrollo	2.33d	28/09/2015	30/09/2015
4	OBJETIVOS BÁSICOS: Fase 1	33d	01/10/2015	03/11/2015
5	Desarrollo del módulo que gestionará la comunicación vía UAF	10d	01/10/2015	11/10/2015
6	Desarrollo del módulo que gestionará los componentes GPIO	3d	11/10/2015	14/10/2015
7	Desarrollo del módulo de comunicaciones WiFy (base)	13d	15/10/2015	28/10/2015
8	Librería de utilidades generales (printf / log)	6d	28/10/2015	03/11/2015
9	OBJETIVOS BÁSICOS: Fase 2	49d	11/11/2015	30/12/2015
10	Sensor de iluminación	7d	11/11/2015	18/11/2015
11	Sensor de temperatura	12d	18/11/2015	30/11/2015
12	Cuerpo principal de tareas de lectura y monitorización	10d	30/11/2015	10/12/2015
13	Módulo menú configuración	7d	10/12/2015	17/12/2015
14	Sensor de pH	8d	17/12/2015	25/12/2015
15	Módulo de alarmas	2d	25/12/2015	29/12/2015
16	Testing	1d	29/12/2015	30/12/2015
17	OBJETIVOS SECUNDARIOS: Fase 3	3d	30/12/2015	02/01/2016
18	Manejo del Alimentador	2d	30/12/2015	01/01/2016
19	Envio información a Arp@lab	0d	01/01/2016	01/01/2016
20	Módulo correo electrónico	0d	01/01/2016	01/01/2016
21	Testing	1d	01/01/2016	02/01/2016
22	OBJETIVOS BASICOS: Documentación	53.38d	01/12/2015	23/01/2016
23	Previa de la memoria	11d	01/12/2015	12/12/2015
24	Entrega final del código	26d	07/12/2015	02/01/2016
25	Memoria definitiva	25d	22/12/2015	16/01/2016
26	Presentación	7d	16/01/2016	23/01/2016

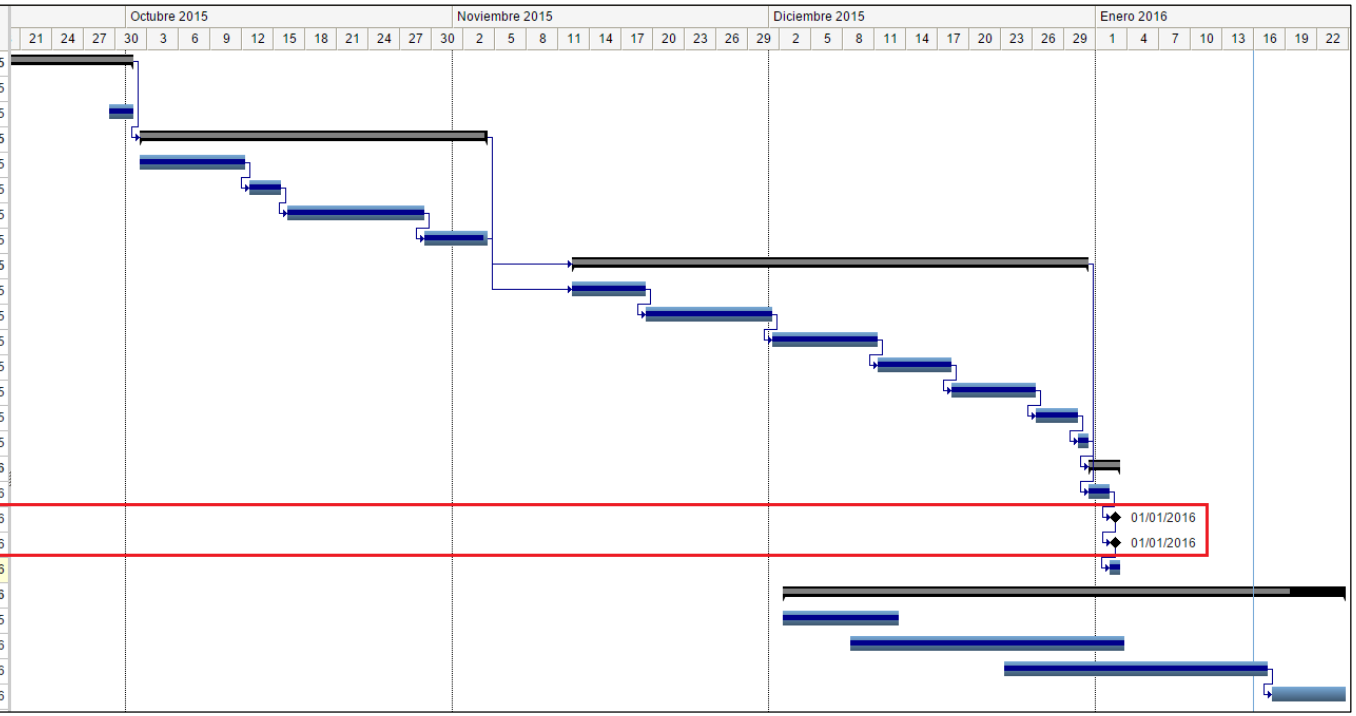


Ilustración 2 - Planificación final

Marcado con un recuadro rojo se pueden ver las tareas que no han sido completadas en la planificación, así como la re colocación de todas las tareas de la Fase 2 en nuevas fechas por los motivos ya descritos.

1.6 Recursos empleados

A continuación se describen todos los recursos que se han empleado para el desarrollo y construcción del sistema:

<u>Elementos principales:</u>

Software:

- Entorno de desarrollo LPCXpresso versión 7.5.0_254

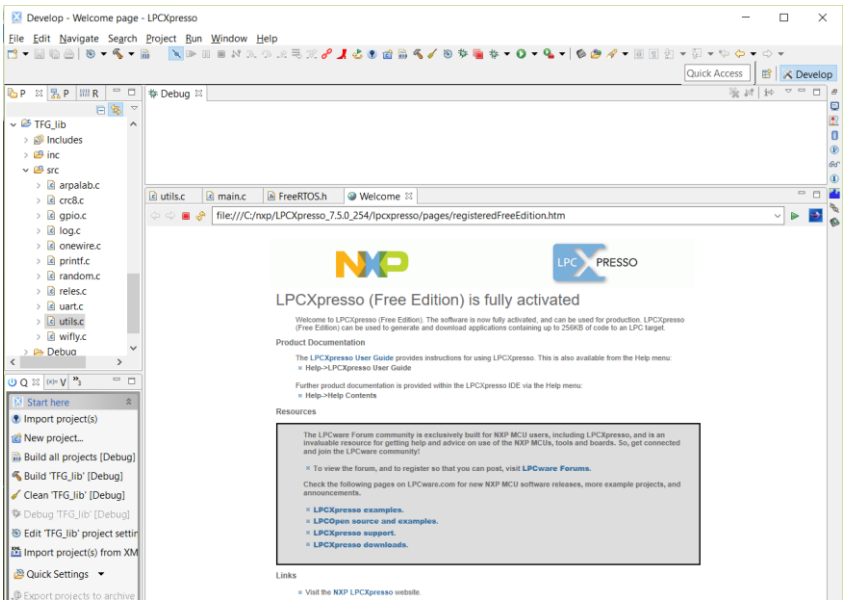


Ilustración 3 - IDE de desarrollo

- Sistema operativo FreeRTOS v7.1.0
- Terminal serie Putty

Hardware:

- Placa NXP de Embedded Artists con procesador LPC1769.

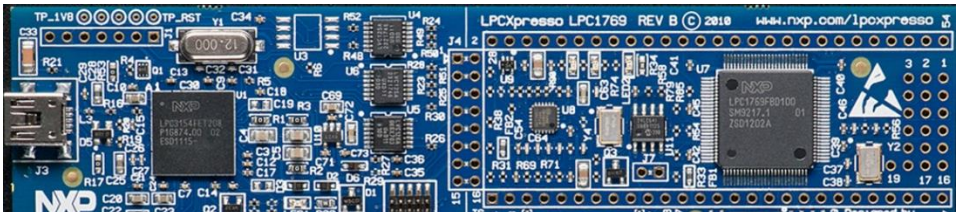


Ilustración 4 - Placa LPC1769

- Módulo Wifly de Roving Networks (actualmente MicroChip) RN-XV ver 3. Con firmware 4.1. Se utiliza un adaptador "*xBee DIP Adapter*" para la conexión.

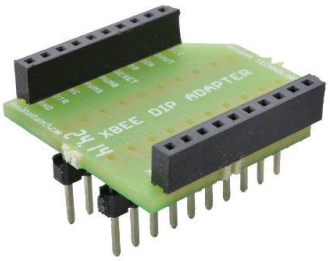


Ilustración 6 – Adaptador DIP xBee

Ilustración 5 - Módulo WiFly RX-XV

- Adaptador USB a serie CP2102

Ilustración 7 - Módulo CP2102

- Relé JQC-3FF del fabricante DFRobot. Relé a 5V y soporta hasta 12ª y 277VAC.

Ilustración 8 - Relé JQC-3FF

- Sonda de temperatura digital, especial para entornos acuáticos. Basada en el sensor DS18S20.

Ilustración 9 - Sensor temperatura DF18S20 digital

- Sonda analógica de medición pH del fabricante DFRobot, modelo SEN0161, especial para entornos acuáticos.

Ilustración 10 - Sonda analógica de pH SEN0161

- Conversor lógico 5v a 3v bidireccional, basado en el transistor BSS138. Se ha usado un pack del fabricante Sparkfun modelo BOB-12009.

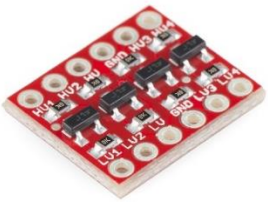


Ilustración 11 - Conversor lógico de nivel BSS138

- Sensor fotosensible FIT0180 del fabricante DFRobot.

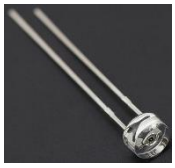


Ilustración 12 - Mini foto sensor

- Alimentador eléctrico para acuarios, RESUN AF-2003.

Ilustración 13 - Dispensador automático de comida

- BreadBoard o placa de pruebas, para conexionado del prototipo inicial.

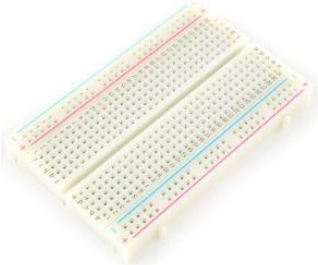


Ilustración 14 - Breadboard para conexiónado del prototipo

- Otros materiales:
    - Transistor BC337
    - Diodos 1N4007
    - Resistencias de diferentes valores: 330ohm, 1k ohm, 100kohm, etc.
    - Cables para interconexión entre las diferentes placas

- Diodos led de color verde y amarillo para la fase de pruebas
- Hub USB para el conexionado de los diferentes elementos, así como su alimentación a 5v.
- Linterna para simular ciclos de día y noche
- Acuario real para las pruebas de mediciones
- Reactivos de pH
- Termómetro profesional de horno para la medición y comparación de temperatura

- PC portátil con 4Gb de memoria, un procesador Core i5 de 4 núcleos y disco SSD de 250Gb.

1.7 Productos obtenidos

Se ha obtenido un sistema que permite realizar el mantenimiento de un acuario en tiempo real. Este sistema principal está compuesto por la placa LPC1769 los sensores conectados a ella para realizar las diferentes lecturas, así como el sistema de control vía relés para la iluminación y alimentación. El sistema contiene un sistema operativo, así como es software desarrollado en este proyecto. Es autónomo, funciona sin intervención humana en base a la programación previa de umbrales y acciones que el usuario haya definido.

En segundo lugar se obtiene un sistema de vigilancia que nos ofrece de forma visual información del acuario a través de un interface de usuario. Este interface es consumible de forma muy sencilla, mediante una conexión vía puerto serie. Este permitirá recibir alarmas cuando se superen determinados umbrales, obtener información sobre el estado del ecosistema del acuario, o configurar parámetros del sistema principal.

1.8 Descripción de capítulos

A continuación detallamos el contenido de cada uno de los capítulos a partir de este punto:

Capítulo 1:

Se realiza una introducción del trabajo realizado justificando en primer lugar el mismo, describiendo brevemente que objetivos buscamos, como lo hemos planificado y que elementos hemos obtenido. Para terminar explicando que productos obtenemos finalmente.

Capítulo 2:

En este capítulo comenzamos describiendo la potencia y avanzadas capacidades que ponen a nuestra disposición hoy en día los pequeños sistemas embebidos. Para seguir describiendo el crecimiento del mercado y aplicabilidad de estos en el año 2015 con una perspectiva hasta 2020.

Capítulo 3:

En este capítulo vamos a hacer una descripción funcional del sistema. Explicando los diferentes módulos, librerías y drivers que lo componen. Sus interrelaciones y modelo de funcionamiento.

Capítulo 4:

Aquí entramos en detalle en la descripción técnica de los puntos descritos desde una perspectiva funcional en el capítulo 3, aquí son descritos técnicamente mediante flujos o diagramas técnicos.

Capítulo 5:

Analizamos la viabilidad técnica de la solución, viendo si existen dispositivos para esta, si son fiables, se pueden mantener, etc.

Capítulo 6:

Se describen todos los componentes que se utilizan en la solución aportando una valoración económica de cada uno de ellos. Se ha dividido en dos bloques, uno dedicado a los componentes físicos (hardware) y un segundo apartado dedicado a las horas hombre de esfuerzo en documentación, pruebas y desarrollo.

Capítulo 7:

Por último en este capítulo se realiza una autoevaluación del trabajo desarrollado revisando los objetivos fijados durante el curso, aspectos mejorables y una autocrítica del trabajo y experiencia obtenida.

El resto de capítulos a partir del número 8 están compuestos por el glosario de términos, bibliografía y anexos que completan el presente documento. Es importante destacar que en los anexos se puede encontrar información necesaria para la compilación del código fuente de la aplicación, así como un manual de uso de la misma para usuarios (no técnico).

2. Antecedentes

2.1 Estado del arte

Un sistema embebido es un conjunto de dispositivos interconectados en una pequeña placa de circuito impreso, los cuales son gobernados por un microprocesador central. Los sistemas embebidos han sido concebidos para que puedan actuar de forma equivalente a otros sistemas más generales pero en un ámbito y con un objetivo muy específico.

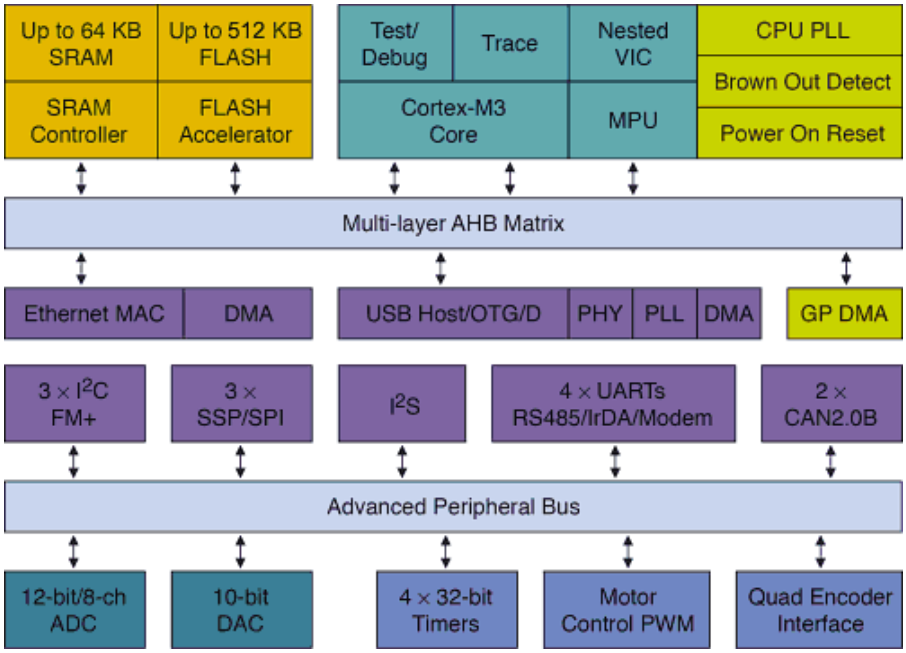


Ilustración 15 - Diagrama de bloques LPC17xx

En la ilustración 15, puede observar el módulo principal de control, basado en una CPU de tipo Cortex-M3, esta ofrece todas las capacidades de un procesador convencional como controlador de interrupciones, procesador matemático, métodos de depuración, etc. el mismo sistema agrupa el resto de periféricos fundamentales para cualquier sistema como son buses de comunicación, controladores de memoria RAM y flash para almacenamiento del software.

Como complemento al sistema agrega una serie de sub-sistemas para la comunicación del sistema embebido con el exterior, como son: conversores analógicos digitales, temporizadores, sistema de red, puertos UART o I2C, etc. Esto permite el uso directo de estos por él sistema, o la expansión a través de diferentes tipos de dispositivos externos.

Todos estos componentes recogidos en un único integrado sería lo que podemos definir como un sistema embebido. Es un sistema especializado y compacto que se ha diseñado para propósitos específicos de manera muy compacta y que frecuentemente es capaz de ejecutar estos propósitos en tiempo real.

2.2 Estudio de mercado

La historia de los sistemas embebidos o empotrados arranca en los años 60 del siglo 20. Fundamentalmente se inicia por dos necesidades concretas: reducir el tamaño de los sistemas, y abaratar los costes. Hasta esa fecha los sistemas de información eran enormes computadoras, y ante circunstancias en las que se necesita reducir el tamaño y peso como es en el envío de una nave al espacio esto se vuelve crítico. De hecho se considera que uno de los primeros sistemas embebidos de la historia fue el ordenador de guiado del cohete *Apollo*, desarrollado por Charles Stark del MIT en el año 1961.

A partir de este momento y demostrando la viabilidad y utilidad de los sistemas embebidos, estos han ido desarrollándose hasta la actualidad en la que podemos encontrarlos en millones de dispositivos en cualquier parte, cajeros de parking, neveras, dispositivos domóticos, pequeños electrodomésticos, etc.

En la actualidad varios factores como la demanda de sistemas multimedia cada vez más portables, el incremento de ventas de dispositivos para consumo personal, la evolución y nuevos desarrollos de sistemas cada vez más pequeños y de menor coste, se ha juntado con la evolución en entornos empresariales e industriales con factores como la mejora exponencial de las comunicaciones máquina a máquina, la implementación de procesos de automatización en sectores industriales de fabricación y procesamiento, o el más reciente y máximo exponente en la actualidad el denominado Internet de las Cosas.

Esto ha generado un mercado que en el año 2020 la consultora Strategyr considera que alcanzará los 233,9$US billones, principalmente en Asia, Europa y Estados Unidos (ver ilustración 16).

Ilustración 16 - Mercados de mayor crecimiento hasta 2020

El mercado está compuesto por 8 fabricantes (ver tabla 1) principales de tecnología para sistema embebidos, en particular microprocesadores y elementos periféricos para la construcción completa del sistema. En el mercado existen más de 57 fabricantes diferentes de nicho que construyen sistemas embebidos específicos para determinados entornos o sistemas.

Fabricantes
ARM Limited
Atmel Corp.
Axiomtek
Fujitsu Limited
HCL Tech Limited
Infiniteon Tech Ltd
Infosys Limited
Intel Corporation

Tabla 1 - Fabricantes sistemas embebidos

La adopción de lenguajes de desarrollo más generalistas ha ayudado también a la amplia adopción de sistemas embebidos, es habitual el uso de lenguajes como C, el más utilizado con casi un 65% de cuota, C++ (20%), ensamblador (5%), java (2%) o incluso .NET (2%), entre otros.

En el ámbito de los sistemas operativos hasta la fecha se han estado utilizando sistemas específicos para estos entornos como QNX, FreeRTOs, entre otros, pero en el estudio de mercado de Strategyr, se muestra como en 2015 hasta un 56% de los proyectos ya utilizan sistemas operativos mucho más generales como puede ser Linux.

El estado del mercado lo podemos resumir en una amplia variedad de soluciones y dispositivos, con un enorme número de periféricos con los que agregarles un valor añadido, y que en los últimos años han facilitado su adopción utilizando herramientas de desarrollo más sencillas y de amplia utilización con lenguajes como C o java, o sistemas operativos como Linux.

Específicamente, en el segmento de acuarios existen varias soluciones, la más parecida a la desarrollada en el presente proyecto es la del fabricante AQ-Aquarium Solutions, el cual dispone de

la solución Aquatronica, que dispone de una serie de módulos conectables a su sistema principal que permiten la vigilancia de umbrales de parámetros físicos y químicos, así como tomar decisiones para solucionar problemas o realizar las tareas de mantenimiento necesarias para el acuario, se puede ver en la ilustración 17 - solución para acuario aquatrónica

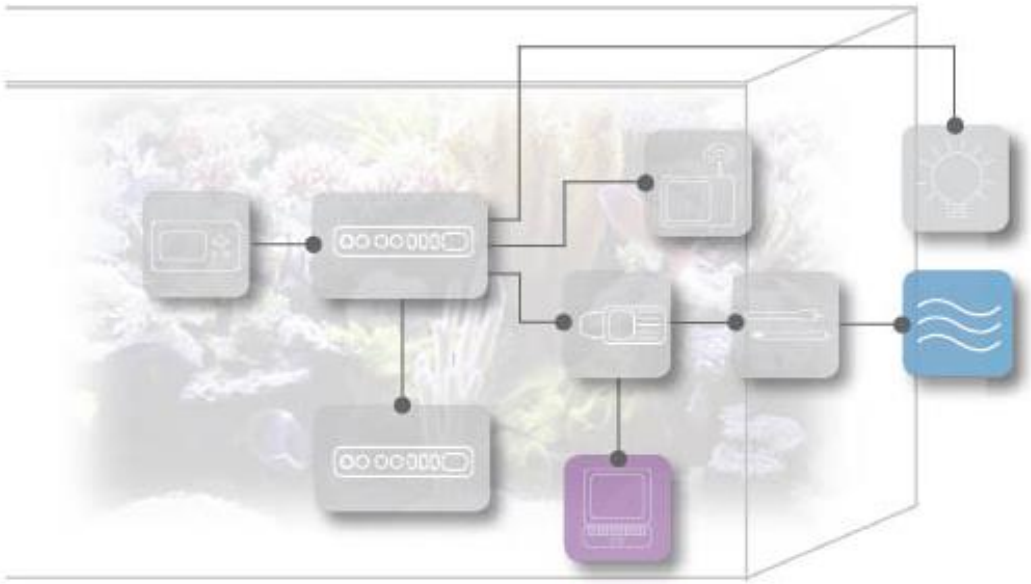


Ilustración 17 - Solución para acuario Aquatrónica

3. Descripción funcional

3.1 Sistema principal

El sistema se ha construido como una única unidad hardware que dispone de un sistema de software embebido ejecutado bajo sistema operativo FreeRTOS. Este sistema conectará todas las piezas y sensores necesarios para desarrollar su actividad. El sistema se ubicará junto al acuario que debe vigilar y mantener, de manera que sus sensores puedan introducirse en el agua, el alimentador dispensar la comida a los animales, y gestionar el encendido y apagado de luces.

El sistema no necesita nada más para funcionar, aunque dispone opcionalmente de una conexión vía puerto serie la cual ofrece un panel de control en el que revisar el estado del acuario, y realizar configuraciones o modificaciones al sistema. Así mismo en esta pantalla se mostrarán las alertas que se produzcan en el mantenimiento del acuario.

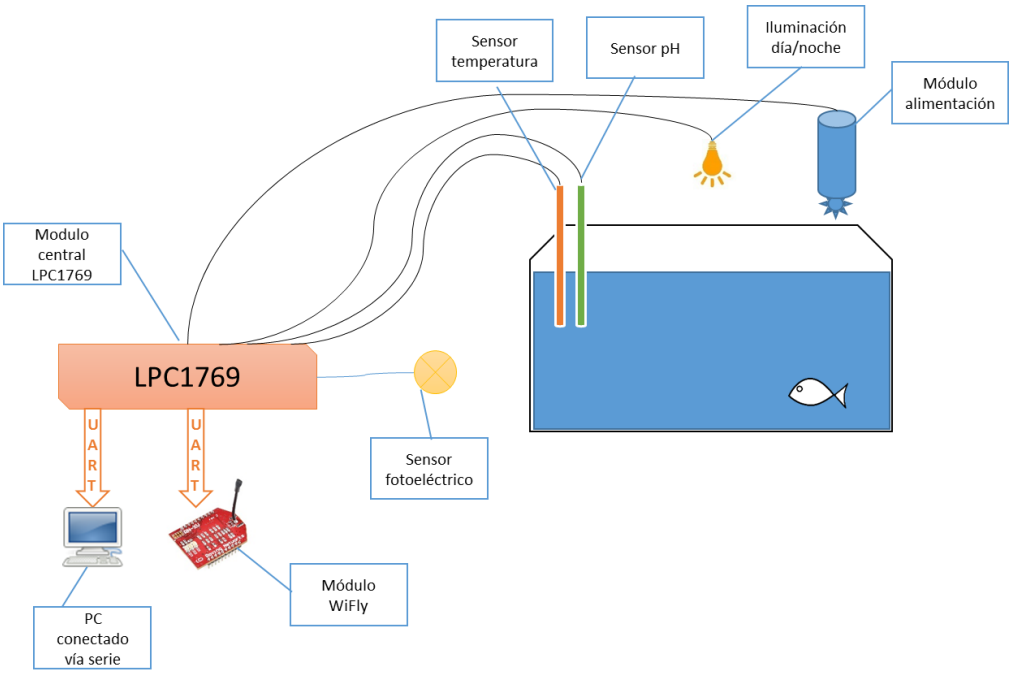


Ilustración 18 - Diagrama del sistema completo

En la ilustración 18, podemos observar todos los elementos físicos que forman parte del sistema y su conexión entre ellos. El sistema dispone de conectividad con el módulo wifi *wifly rn-xv*, de forma

que en un futuro el sistema se pueda extender y acceder a sus datos y configuraciones vía web o consola remota.

La forma de interacción entre los diferentes elementos del sistema se muestra en la ilustración 19 a continuación:

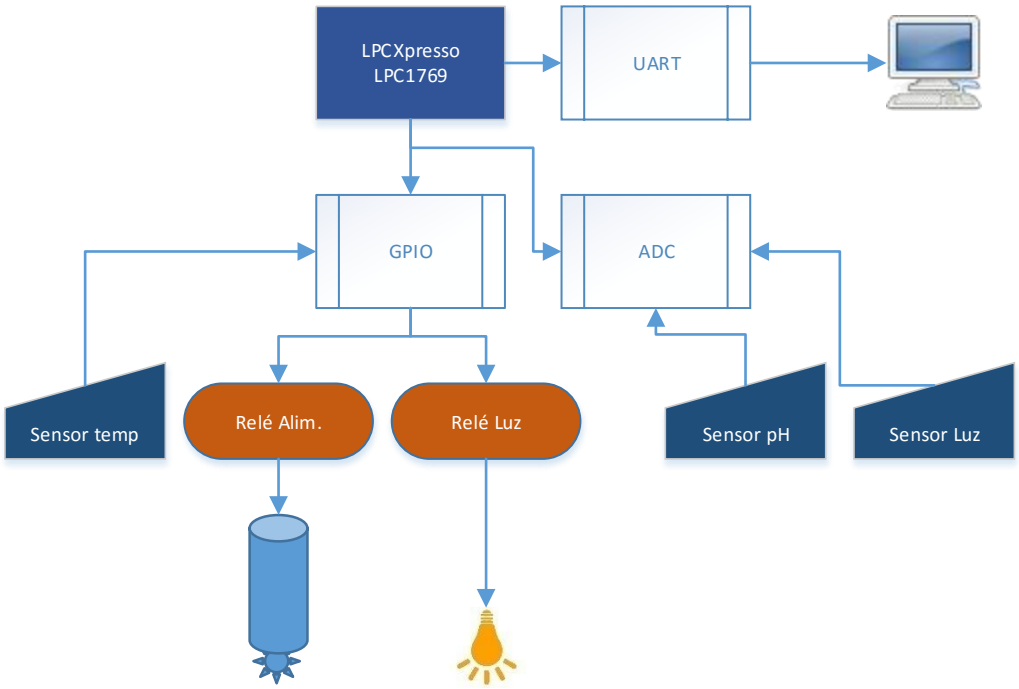


Ilustración 19 - Interacción de elementos

En el siguiente diagrama de bloques (ilustración 20) se puede ver cómo interactúan entre si los componentes del sistema:

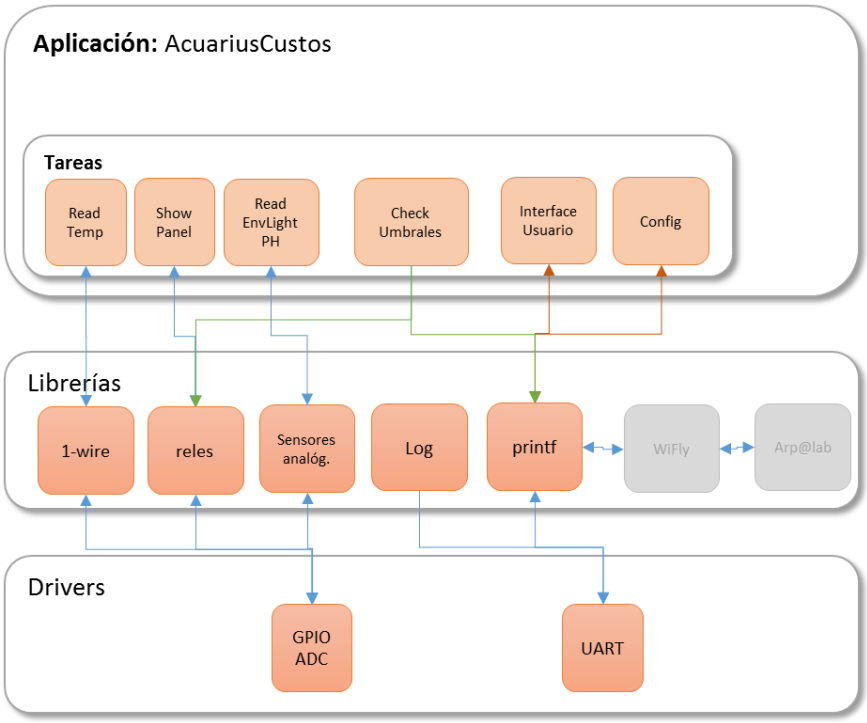


Ilustración 20 - Diagrama de bloques

La capa de drivers es la que más cerca del hardware se encuentra, proporcionando servicios de gestión de los componentes de la placa y el procesador. La abstracción de los diferentes servicios utilizados (o dispositivos) se lleva a cabo a través de las diferentes librerías, para terminar con el cuerpo principal de la aplicación gestionado en diferentes tareas.

Drivers: realizan funciones específicas del hardware a través de la librería CMISIS. En particular se han construido dos drivers:

- GPIO/ADC: este driver es responsable de la gestión y configuración de los puertos GPIO, siendo capaz de establecer el estado de los diferentes puertos, o leer datos de estos de acuerdo al modo establecido, por ejemplo esperando a que una conversión digital/analógica termine.
- UART: en este caso se realiza mediante un driver desarrollado por un tercero la gestión vía interrupciones de los 4 puertos UART que dispone la placa LPC1769. Esto incluye la configuración de los diferentes puertos, la lectura de datos del puerto, así como su escritura.

Librerías: estas librerías entregan servicios con independencia del hardware utilizado a los programas principales, a continuación referimos las principales librerías:

- Printf/Log: se encarga del envío y recepción de información estructurada por el puerto serie, apoyándose en el driver UART. La librería log apoyándose en printf realiza el envío de información de depuración también vía puerto serie.
- Sensores analógicos: es una librería que permite la lectura de datos en cualquiera de los puertos ADC (conversor analógico digital). Se utilizarán sobre todo para la lectura de datos de luz ambiental y de la sonda de pH, ambos sensores analógicos.
- Relés: librería que apoyándose en el driver GPIO modificará el estado durante un determinado tiempo de un pin digital de forma que este active o desactive el relé conectado a este puerto. Esta librería juega un papel fundamental en la función de alimentación y luz artificial.
- 1-wire: librería dedicada a implementar el protocolo 1-wire del fabricante Microchip y que se utilizará para obtener datos de la sonda de temperatura que implementa este protocolo.

Por último el diagrama de bloques nos muestra el grupo de tareas responsable de la ejecución del sistema, estas las podemos agrupar en tres grandes grupos: lectura de datos, chequeo de situación y ejecución de acciones, gestión del interface de usuario.

- Lectura de datos: este grupo de tareas, leen datos de los diferentes sensores, y los van almacenando en las variables del sistema que manejan esta información, para después ser procesadas y analizadas.
- Chequeo de situación y acciones: esta tarea es responsable de comprobar que los datos leídos se encuentran en los umbrales definidos por el usuario y lanzar alarmas en caso contrario. También comprobarán los horarios e intensidad lumínica para encender o apagar la luz ambiental, o para activar el alimentador.
- Interface de usuario: esta última tarea, es responsable de mostrar los datos del acuario en pantalla para el usuario, así como gestionar la entrada de configuraciones de umbrales, horarios, etc. por parte del usuario.

3.2 Interface de usuario

La interface de usuario que se ha creado está basada en texto ya que de forma estándar esta será publicada vía un puerto serie que solo nos permite la transmisión de caracteres y algunos códigos

de control muy simples como borrado de pantalla, retroceso, cambio de línea o de color de fondo y letras.

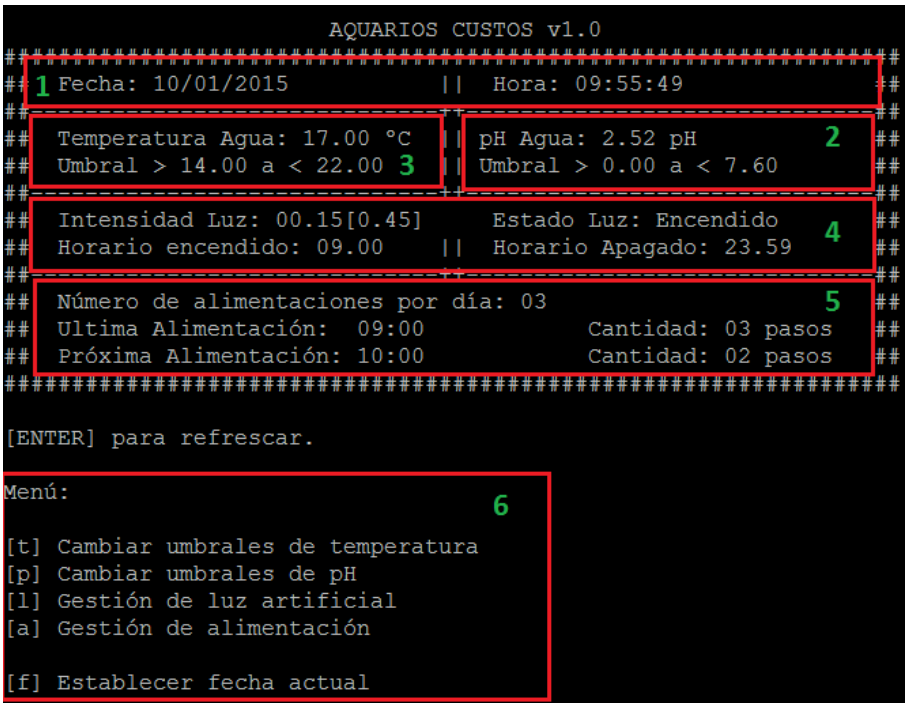


Ilustración 21 - Interface de usuario

En la ilustración 21 se puede ver la interface de usuario, en la que se muestran datos de lecturas actuales, así como las configuraciones relacionadas con estas lecturas.

- Zona 1: Fecha y hora actual del sistema
- Zona 2: Lectura del pH actual del acuario, y umbrales fuera de los que arrojará alarmas para su atención.
- Zona 3: Lectura actual del agua del acuario y umbrales que si son superados lanzarán alarmas.
- Zona 4: Configuración del horario de encendido de la luz artificial del acuario, estado de la luz actualmente, lectura de luz ambiente y entre corchetes la configuración de la luz que si es superada no encenderá la luz ambiente ya que no será necesario.
- Zona 5: Muestra el número de alimentaciones diarias que se realizarán en el acuario, cual es la siguiente alimentación y cuantos pasos de alimento se proporcionarán. También se puede ver cuál fue la última alimentación que se administró al acuario.

- Zona 6: Menú para el usuario que permite tocar las configuraciones o datos del sistema.

Como ya se ha explicado el sistema cuenta con un módulo de alarmas el cual muestra de forma visual en pantalla cuando alguno de los umbrales se supera:

```
                        AQUARIOS CUSTOS v1.0
#####################################################################
##  Fecha: 10/01/2015           ||  Hora: 10:04:07               ##
##-------------------------------++-------------------------------##
##  Temperatura Agua: -1.00 °C  || pH Agua: 2.06 pH              ##
##  Umbral > 14.00 a < 22.00    || Umbral > 7.00 a < 7.80        ##
##-------------------------------++-------------------------------##
##  Intensidad Luz: 00.23[0.45]     Estado Luz: Encendido        ##
##  Horario encendido: 09.00    ||  Horario Apagado: 23.59       ##
##-------------------------------++-------------------------------##
##  Número de alimentaciones por día: 03                         ##
##  Ultima Alimentación:  10:00           Cantidad: 02 pasos     ##
##  Próxima Alimentación: 18:00           Cantidad: 05 pasos     ##
#####################################################################

[ENTER] para refrescar.

Menú:

[t] Cambiar umbrales de temperatura
[p] Cambiar umbrales de pH
[l] Gestión de luz artificial
[a] Gestión de alimentación

[f] Establecer fecha actual



[AVISO] El sensor de temperatura definido no se ha encontrado

[AVISO] El umbral de pH está fuera de rango. pH actual: 2.47, pH. mínimo: 7.00, pH.
 máximo: 7.80
```

Ilustración 22 - Alarmas en interface de usuario

En la ilustración 22 se puede ver dos de las alarmas posibles, en concreto una porque el sensor de temperatura está desconectado, y una segunda por que la lectura de pH está fuera de los umbrales definidos previamente.

Las alarmas que el sistema soporta en la actualidad son:

- Sensor de temperatura no detectado
- Sensor de pH no detectado
- Umbral de temperatura superado (inferior o superior)
- Umbral de pH superado (inferior o superior)
- Alimentación en curso (número de alimentación enviado)

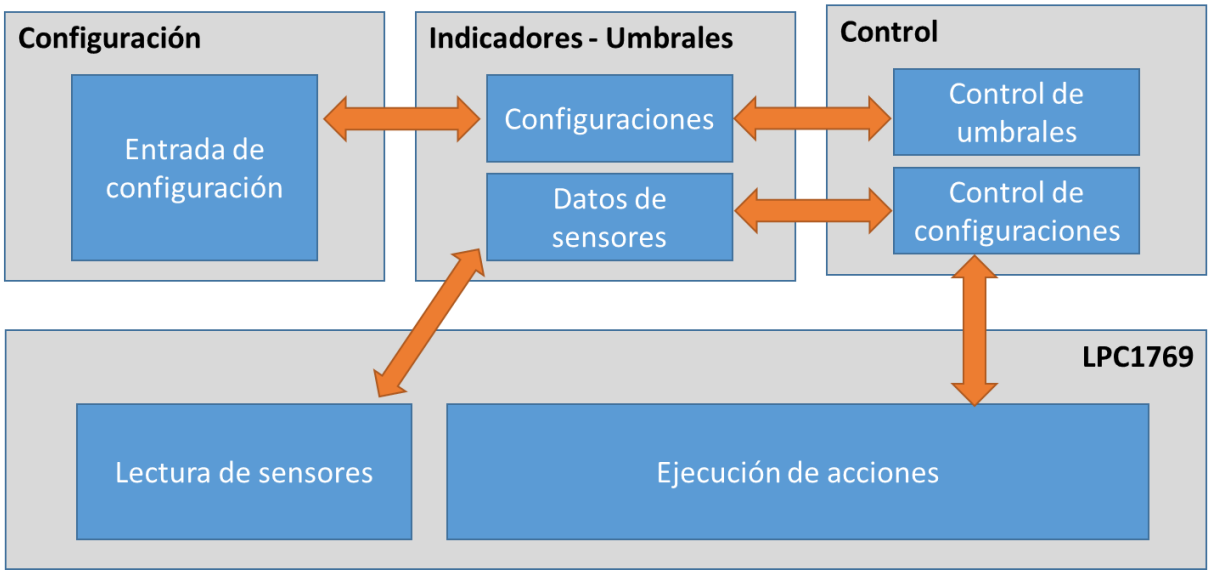


Ilustración 23 - Diagrama de bloques Interface de usuario

En la ilustración 23 se puede ver el diagrama de bloques del interface de usuario, como se muestran en los indicadores/umbrales los diferentes datos y parámetros.

El módulo de configuración se ocupará de modificar estos valores y umbrales, que posteriormente serán utilizados por el módulo de control en comparación con los datos de los sensores, para comprobar los rangos o umbrales de estos y solicitar al LPCXpresso la ejecución de las diferentes acciones.

De manera continua la placa LPC1769 está realizando la lectura de los diferentes sensores conectados y almacenándolos de forma centralizada para su posterior procesado por el sistema de control.

3.3 Aplicación del sistema encastrado LPC1769

La aplicación principal ejecuta sobre el sistema LPC1769, alma de todo el proyecto, a continuación en la ilustración 24 se muestra su diagrama de bloques.

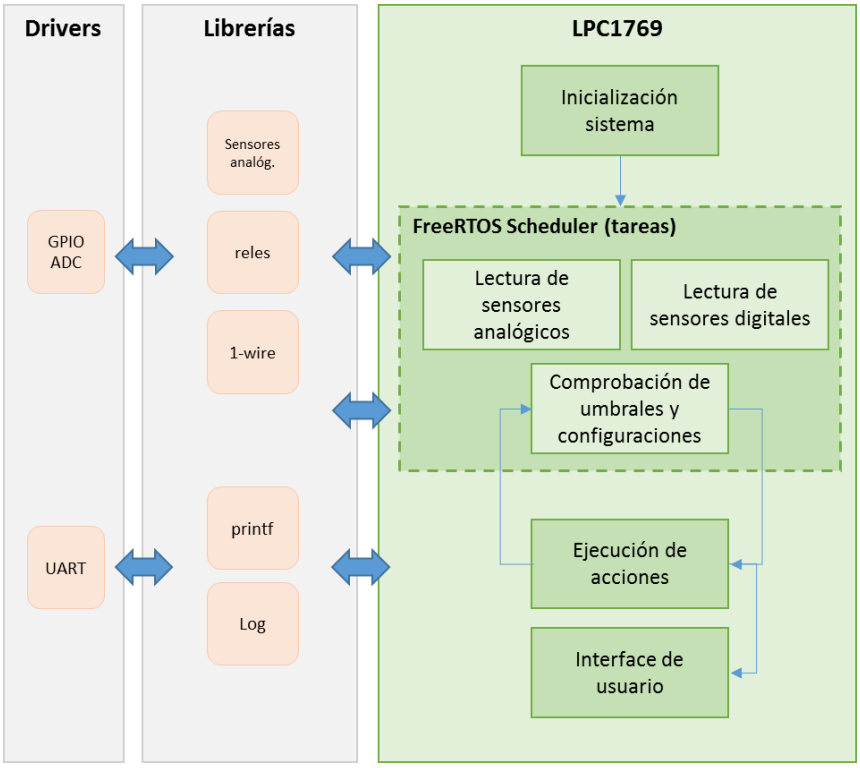


Ilustración 24 - Diagrama de bloques aplicación del sistema encastrado

Esta aplicación se apoya en los módulos previamente descritos que facilitan la gestión de los diferentes dispositivos hardware que el sistema tiene conectados. La aplicación principal dispone de un módulo de inicialización, que se ocupa de iniciar todos los puertos y sensores que serán necesarios posteriormente, además de establecer los valores de umbrales definidos previamente con objeto que la aplicación en su arranque pueda comenzar su tarea.

Un grupo de tareas gestionadas por el *scheduler* del sistema operativo, FreeRTOS será el responsable de las diferentes actividades de vigilancia y ejecución de acciones programadas previamente. Estas tareas se agrupan en 4 grandes bloques, interface de usuario, lectura de sensores digitales, lectura de sensores analógicos y chequeo de umbrales junto con ejecución de acciones o alarmas.

El interface de usuario será el responsable de informar a este de lo que está sucediendo en tiempo real en un formato inteligible por un ser humano. Destacan datos como los indicadores de los sensores, las alarmas activas, o determinadas acciones como la alimentación o estado de la luz artificial que son mostradas en este interface. Es importante resaltar que el sistema es 100% autónomo y no necesita intervención humana alguna, salvo para modificar umbrales o parámetros del mismo.

El interface es accesible en cualquier momento vía puerto serie o USB, mostrando de forma inmediata toda la información en tiempo real. Adicionalmente el sistema cuenta con un mecanismo para refrescar la información en la pantalla simplemente pulsando la tecla ENTER.

4. Descripción detallada

En el siguiente apartado vamos a describir cada uno de los componentes del sistema en profundidad desde una perspectiva técnica.

4.1 Descripción del Hardware

4.1.1 Placa LPC1769

La placa LPCXpresso para el procesador LPC1769 está basada en la arquitectura Cortex-M3 de ARM. NXP como fabricante de la placa LPCXpresso LPC 1769, utilizada en este proyecto, ha aprovechado un alto nivel de integración de los componentes otorgados por este procesador, para entregar un sistema muy sencillo de programar, depurar, y que dispone de todos los elementos necesarios en un sistema IT como se muestra en la ilustración 25.

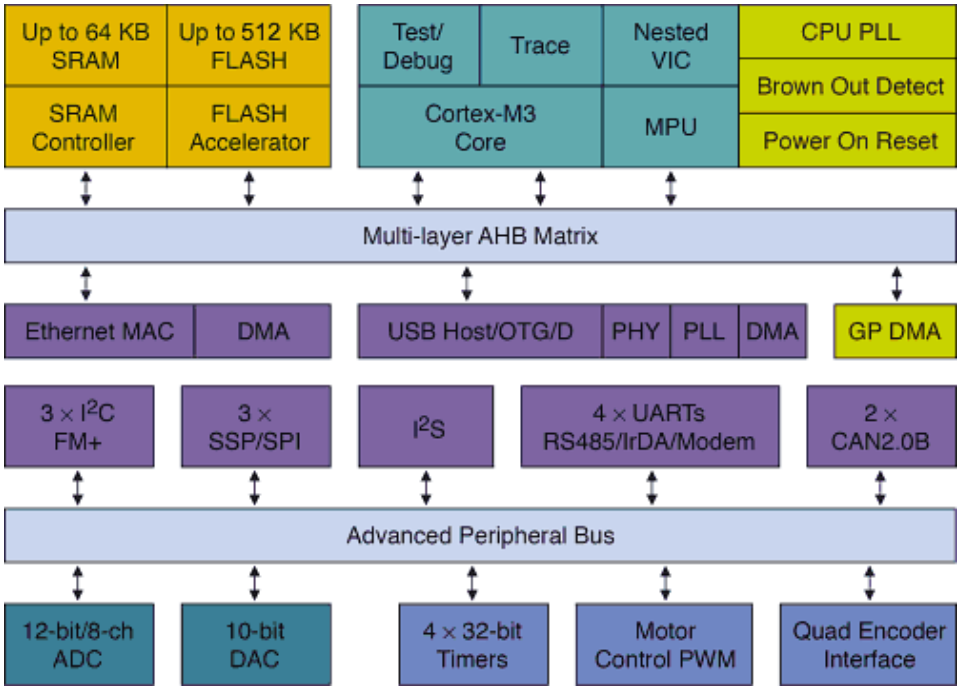


Ilustración 25 - Diagrama de bloques de la placa LPC1769

La unidad LPC1769 dispone de 64kb de memoria SRAM, 512kB de memoria Flash, 4 puertos UART, 3 puertos I2C, SPI, 2 SSP, 2 controladores de red de área local (CAN), control de motores PWM, puerto USB, un RTC de bajo consumo con posibilidad de alimentarlo con batería de forma externa, un puerto Ethernet, I2S, un conversor analógico digital de 8 canales de 12bit, un conversor digital analógico de 10bits, además de hasta 70 pins de propósito general de entrada y salida. Puede funcionar con velocidades hasta 120Mhz. Incorpora un gestor de interrupciones (NVIC) y 8 canales tipo DMA.

El sistema puede alimentarse por el puerto JTAG también incluido en la placa, con una tensión entre 3,15v a 3,3v. Dispone de diferentes modos de operación destinados a reducir el consumo eléctrico del sistema, estos modos son: *Sleep, Deep Sleep, Power-Down y Deep Power Down*.

El sistema dispone de varias facilidades de depuración, entre las que destaca la principal a través de un puerto de tipo JTAG para pruebas y depuración.

Dispone de un completo IDE de desarrollo para C y C++ y placas LPCXpresso. Este mismo entorno IDE dispone de todas las herramientas necesarias para la depuración de la ejecución del software que corre en la placa, desde ejecución paso a paso, análisis de registros y variables, o estados de los diferentes indicadores de la placa.

El fabricante NXP, entrega diferentes códigos de ejemplo de ámbito general que muestran las funcionalidades de la placa LPCXpresso. Ofrece un modelo de soporte a través de foros públicos en su página WEB, en los que una amplia comunidad de desarrolladores ayudan o incluso publican versiones de sus librerías o programas.

Sobre la placa LPCXpresso se ha realizado el diseño de la solución que se muestra en la ilustración 26:

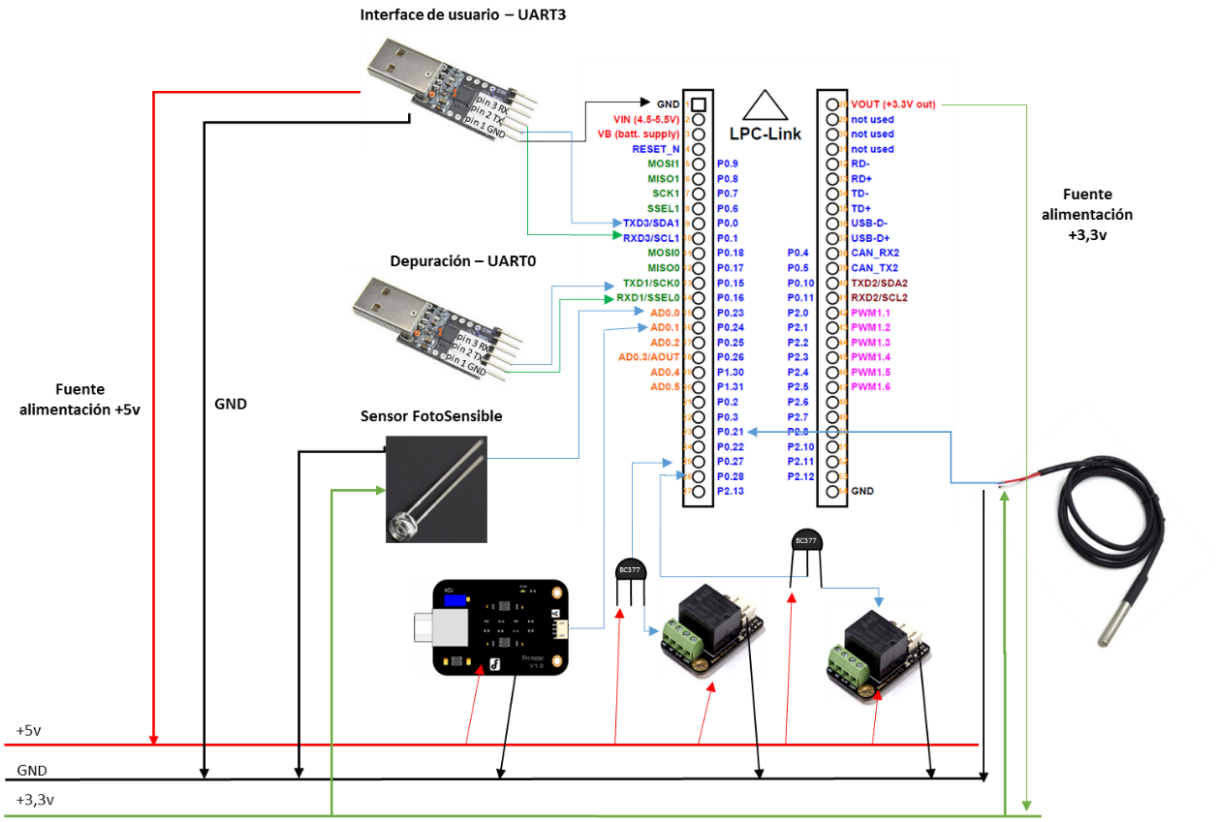


Ilustración 26 - Esquema general del sistema

4.1.2 Conectividad puerto Serie: librerías printf y log, driver uart.

La placa principal LPCExpreso como se ha descrito contiene 4 UARTs disponibles a través del puerto de propósito general de entrada y salida.

El presente proyecto soporta el uso de 3 puertos serie para diferentes propósitos (UARTS: 0, 1 y 3). Hemos utilizado el adaptador de puerto serie-usb CP2102 para acceder al interface de usuario, así como para tener acceso a la consola de depuración del sistema.

Ilustración 27 - Conversor USB-Serie

El sistema realiza una conversión de los datos serie recibidos en los pines RX y TX a USB los cuales pueden ser recibidos en el PC o dispositivo compatible que mediante un terminal serie configurado a la velocidad requerida podrá visualizar los datos enviados por el sistema LPC.

El esquema de conexiones de los dos conversores serie utilizados son como sigue a continuación en la ilustración 28:

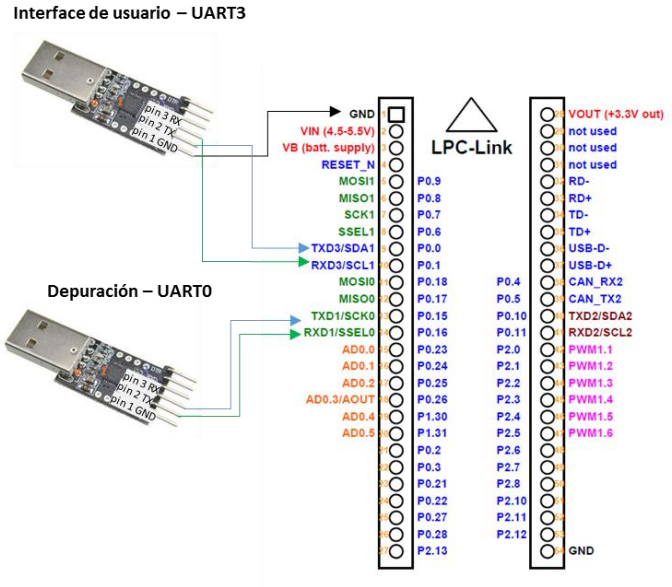


Ilustración 28 - Conexión conversores serie-USB

Para la gestión de los puertos UART junto con los conversores conectados para su acceso desde dispositivos compatibles se han desarrollado las siguientes librerías y funcionalidades:

Driver UART:

Este driver ha sido desarrollado por el propio fabricante NXP Semiconductors, y gestiona los puertos UART de la placa 0, 1 y 3. El driver funciona en base a interrupciones que generan los propios puertos y que debe gestionar el programa en el caso de la recepción y envío de información. Estas son sus principales funciones:

- UARTInit: indicando el número de puerto y velocidad, el sistema establecerá la configuración de este puerto a la velocidad indicada.
- UARTSend: es la función responsable que utilizarán las diferentes librerías para el envío de información al puerto serie desde la placa LPCXpresso.

Librería printf:

Esta librería es la responsable de enviar de forma ordenada y con formato información a un puerto serie, así como de recibir la entrada desde el mismo, para ello existen dos funciones principales:

- Printf_Start: se apoya en el driver UART para inicializar el puerto serie que con posterioridad envían información el resto de comandos, así como la velocidad de este.
- Printf_Print: permite el envío de información a un puerto serie en forma de caracteres o códigos de control, utilizando las características de formateo de cadenas que nos permite el lenguaje C.
- Printf_Scan: esta función nos permite leer caracteres en la entrada del puerto serie con objeto de capturar la pulsación de teclas, textos o números desde el puerto serie, o por parte de dispositivos como el módulo wifly.

Librería log:

Esta librería es responsable de generar un log en formato que un humano pueda leer utilizando un puerto UART. La librería de log se apoya en la librería printf para la comunicación con el puerto serie, esta se ocupa de mantener hora y fecha de la entrada de depuración así como número correlativo de la línea de log. Las dos principales funciones son:

- Log_Start: inicializa la comunicación de los logs de depuración vía puerto serie. En esta se indicará el puerto y velocidad que se utilizará en el mismo.
- Log_Log: esta función se ocupa de enviar al puerto serie que inicialicemos, la línea formateada siguiendo la nomenclatura del lenguaje C con la información de depuración.

4.1.3 Sensores analógicos

En el proyecto se utilizan dos sensores analógicos, sensor de pH y de luz ambiental, a continuación se describen los dispositivos, esquemas de conexión de estos, así como los módulos que los gestionan.

Sensor fotosensible:

Este sensor funciona como una resistencia que cambiará su impedancia en función de la cantidad de luz que la célula fotoeléctrica reciba en su sensor. Desde la placa del LPCXpresso seremos capaces en un pin de tipo ADC de leer los diferentes voltajes que nos llegan a este pin.

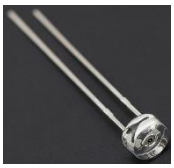


Ilustración 29 - Mini fotosensor

El sensor funciona a 5V y consume 70mW. Es de un tamaño pequeño y su parte superior tan solo ocupa 5mm.

Debido al voltaje de operación de la célula hemos utilizado un divisor de tensión basado en una pequeña resistencia de 66k ohms, para conseguir que el valor máximo que nos llegue al pin analógico al realizar las lecturas no supere los 3,3v máximo.

El esquema mostrado en la ilustración 30 muestra la conexión del sensor de luz con la resistencia de 66k ohm que actua como divisor de la tensión de salida de 5v, convirtiéndola en la máxima tensión tolerada por la plata LPCXpresso, 3,3v.

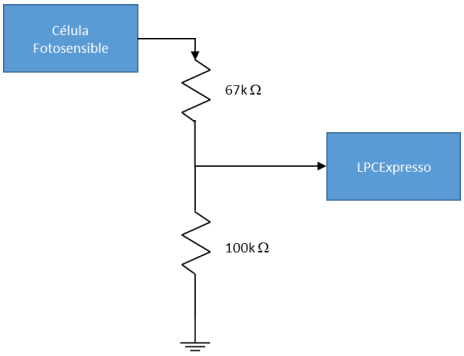


Ilustración 30 - Divisor de tensión 5v a 3,3v

Este divisor de tensión se utilizará también en las lecturas del sensor de pH, el cual también opera a 5v.

Sensor de pH:

Este sensor de pH es especial para soportar el estar sumergido durante largas temporadas en un medio acuático como puede ser un acuario. Este modelo especifico sumergible se puede observar en la ilustración 31

Ilustración 31 - Sonda de pH

Está compuesto por dos partes diferenciadas una primera parte en forma de placa con un conector BNC (ilustración 30) para la sonda, y 3 pines de conexión para la alimentación y envío de información al pin analógico. Este pin variará el voltaje entre 0,00 y 414,12 miliVoltios. Siguiendo la tabla 2 se puede ver los valores de conversión.

VOLTAGE (mV)	pH value	VOLTAGE (mV)	pH value
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

Tabla 2 - Correspondencias mV a pH

La placa dispone de un conector BNC al que se conectará la sonda que generará esta variación de miliVoltios en función de cómo reaccione el material sensible al pH, que incluye en su punta y que provoca estas variaciones de voltaje. Esta sonda debe estar sumergida en el acuario de forma constante y la sonda tiene una vida media de 6 meses tras los que deberá ser sustituido.

Las características técnicas del dispositivo son: alimentación de 5v, capacidad para realizar lecturas de pH entre 0 y 14, rango de error <0,1pH en condiciones de 25ºC, potenciómetro para ajustar la ganancia.

De manera regular la sonda de pH debe ser calibrada con un medio externo, este puede ser comparando el resultado contra otro método de medición (por ejemplo tiras reactivas), o bien utilizando una solución con un pH conocido previamente e introduciendo la sonda en este y así ajustando el potenciómetro para llegar al valor esperado.

Librería ADC:

Estos dos sensores se apoyan en la librería para la gestión del conversor Analógico-Digital. La librería utiliza el driver GPIO para realizar la configuración de diferentes pines de la placa LPC como entradas ADC. En concreto en el presente proyecto se utilizan los pin 15 y 16 que corresponden a AD0.0 y AD0.1.

- ADC_Init: esta función se ocupa de recibir el número de puerto del conversor analógico a digital e inicializar el pin correspondiente en el GPIO con esta configuración. Además inicia la operación del ADC.
- ADC_Read: Solicita la lectura de un dato del pin correspondiente al conversor ADC, espera a que el ADC haya terminado la conversión, y devuelve este dato tal cual lo entrega el ADC sin ningún tipo de conversión. El receptor del dato deberá ser quien convierta el dato a mV o la lectura final que necesite.

4.1.4 Sensor digital

Para la medición de la temperatura del agua del acuario se buscó una sonda que permitiera permanecer sumergida largos periodos de tiempo en el líquido. El fabricante DFRobot dispone de una sonda específica para este medio basada en el integrado de Maxim DS18S20. Este integrado utiliza un protocolo digital denominado 1-wire que permite interactuar mediante una única conexión con el integrado DS18S20.

Ilustración 32 - Sensor de temeratura 1-wire

Para la gestión de este protocolo se ha utilizado una librería de terceros publicada por NXP en sus foros.

La sonda de temperatura opera en un intervalo de entre 3,0v y 5,0v. Es capaz de leer temperaturas entre -55ºC y +125ºC con un error menor a 0,5ºC para lecturas entre -10ºC y +85ºC. Dispone de un número serie único cada integrado de forma que se puedan conectar diferentes sensores en el mismo bus.

La implementación del protocolo 1-wire sigue un sistema equivalente a un modelo half-duplex en el que los mensajes solo pueden viajar en un sentido por vez. Para ello define un formato de comunicación en el que se especifican tiempos de cada señal (o bit de dato), tiempos de espera y tiempos de recepción de datos.

Previo a cualquier comunicación siempre necesitaremos inicializar el bus, siguiendo el esquema de tiempos que se puede ver en la ilustración 33:

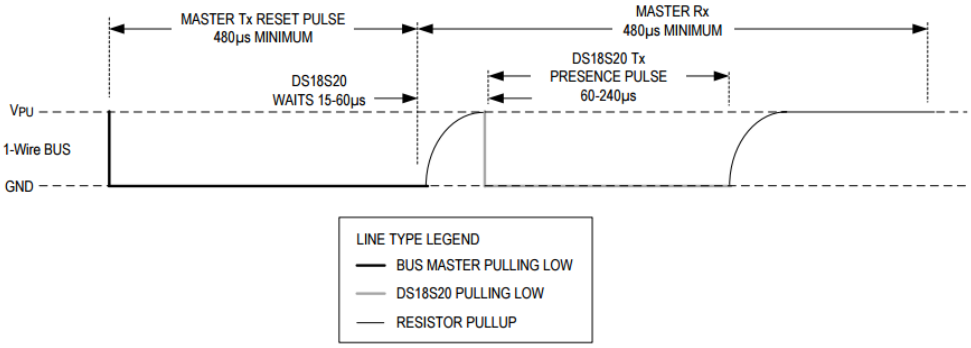


Ilustración 33 - Inicializacion bus DS18S20

A partir de este momento el bus está listo para el envío de comandos por parte de nuestro sistema, y la lectura de la respuesta de estos conforme al formato mostrado en la ilustración 34:

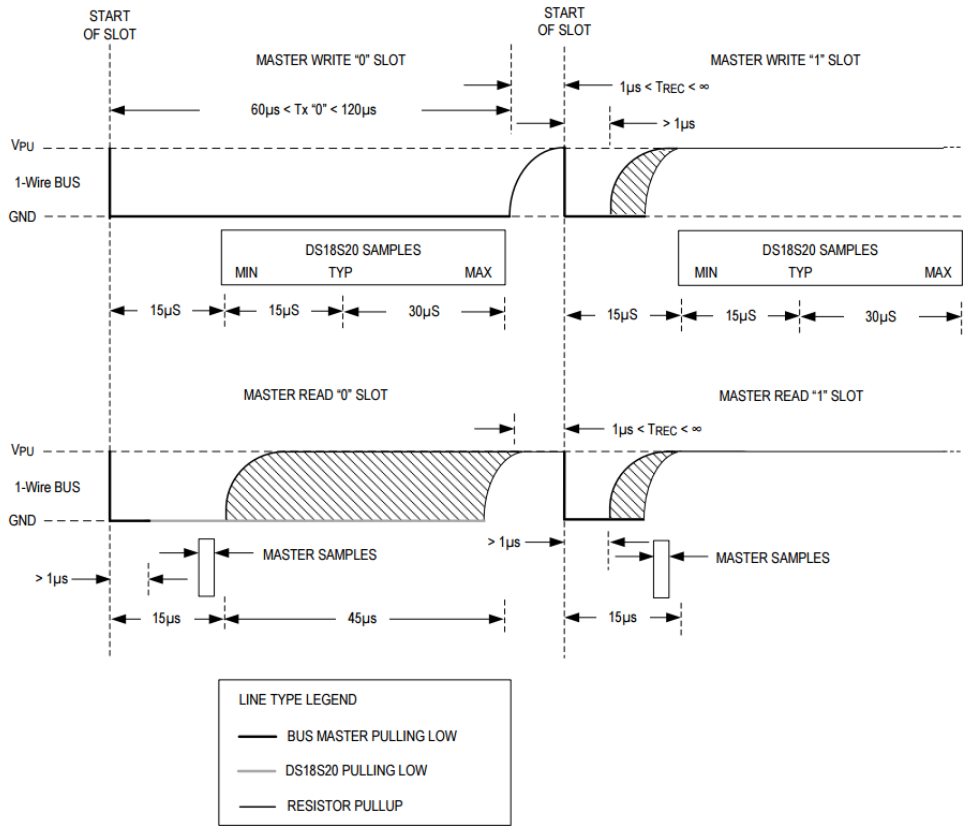


Ilustración 34 - protocolo de tiempo para envio y recepcion

Librería 1-Wire:

La implementación y soporte a este protocolo se ha realizado en base a una librería existente en los foros del fabricante LPC a la cual se le ha realizado varias modificaciones ya que tras un exhaustivo análisis se descubrió un bug en la misma al utilizar una única sonda de temperatura. Estas modificaciones están documentadas en el código fuente de la aplicación.

- ow_Init: es responsable de enviar una señal de reset a los dispositivos conectados al bus 1-wire para su inicialización, y así poder realizar el envío de comandos posteriormente. También es responsable de inicializar el puerto GPIO al que se conectará el pin del bus 1-wire.
- ow_find_devices: esta función permite buscar los dispositivos (por número de serie) que están conectados al bus 1-wire. También permite añadir un parámetro con el número serie del sensor o sensores que tenemos conectado con objeto de comprobar que están presentes en el bus, y si no fuera así lanzar una alerta.

- ow_read_temp: función responsable de lanzar una petición de lectura al dispositivo vía bus 1-wire. La lectura que devuelve el sensor está establecida en pasos de 0,5ºC.

4.1.5 Relés

El sistema incluye dos relés destinados a controlar dos elementos principales del sistema. En primer lugar serán responsables de la iluminación artificial en caso de esta ser necesaria, encendiendo una lámpara especial para acuario durante las horas diurnas, y apagándola en las horas nocturnas.

En segundo lugar el otro relé será el responsable de accionar el alimentador automático para la pecera el número de veces y a la hora programada.

Para esto hemos utilizado relés del fabricante DFRobot basados en el componente JQC-3FF de Futurlec. En la Ilustración 35se muestran sus características técnicas:

File No.:E134517
File No.:40025218
File No.:R50148356
File No.:CQC08002027861

Features

- 15A switching capability
- 1 Form A and 1 Form C configurations
- Subminiature, standard PCB layout
- Plastic sealed and flux proofed types available
- Environmental friendly product (RoHS compliant)
- Outline Dimensions: (19.0 x 15.2 x 15.5) mm

COIL

Coil power	5VDC to 24VDC: 360mW; 48VDC: 510mW

COIL DATA at 23˚C

Nominal Voltage VDC	Pick-up Voltage VDC	Drop-out Voltage VDC	Max. Allowable Voltage VDC	Coil Resistance Ω
5	3.80	0.5	6.5	70 x (1±10%)
6	4.50	0.6	7.8	100 x (1±10%)
9	6.80	0.9	11.7	225 x (1±10%)
12	9.00	1.2	15.6	400 x (1±10%)
18	13.5	1.8	23.4	900 x (1±10%)
24	18.0	2.4	31.2	1600 x (1±10%)
48	36.0	4.8	62.4	4500 x (1±10%)
48 1)	36.0	4.8	62.4	6400 x (1±10%)

Notes: 1) There are 2 types for 48V--510mW and 360mW. The coil resistance for 510mW type is 4500ohm while for that for 360mW type is 6400ohm. If 360mW type is required, please add a special suffix (068) in the ordering information.

CONTACT DATA

Contact arrangement	1A	1C
Contact resistance	100mΩ (at 1A 6VDC)	
Contact material	$AgSnO_2$, AgCdO	
Contact rating (Res. load)	10A 277VAC/28VDC	
Max. switching voltage	277VAC / 30VDC	
Max. switching current	15A	10A
Max. switching power	2770VA / 210W	
Mechanical endurance	1×10^7OPS	
Electrical endurance 1)	1×10^5OPS (NO, at 7A 250VAC) 5×10^4OPS (NO, at 10A 250VAC)	

CHARACTERISTICS

Insulation resistance		100MΩ (at 500VDC)
Dielectric strength	Between coil & contacts	1500VAC 1min
	Between open contacts	750VAC 1min
Operate time (at nomi. volt.)		10ms max.
Release time (at nomi. volt.)		5ms max.
Shock resistance	Functional	98m/s²
	Destructive	980m/s²
Vibration resistance		10Hz to 55Hz 1.5mm DA
Humidity		35% to 85% RH
Ambient temperature		-40˚C to 70˚C
Termination		PCB
Unit weight		Approx. 10g
Construction		Plastic sealed, Flux proofed

Notes: 1) For sealed type, the vent-hole cover should be excised.
2) The data shown above are initial values.
3) Please find coil temperature curve in the characteristic curves below.

Ilustración 35 - Características rele JQC-3FF

Este relé actúa con una tensión de 5V. La operación a 5V de estos relés ha supuesto realizar una serie de adaptaciones al circuito ya que la placa LPCXpresso en los pin GPIO solo es capaz de llegar hasta 3,3v y esta tensión no sería suficiente para accionar la bobina. Para conseguir este objetivo se ha utilizado un pequeño circuito con un transistor y un diodo, la ilustración 36 muestra el esquema de conexión utilizado:

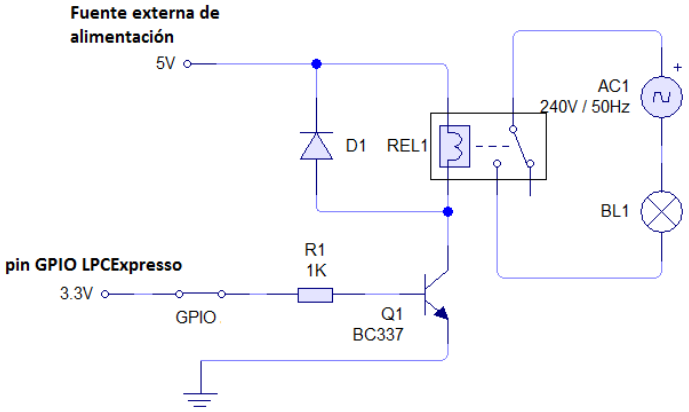


Ilustración 36 - Esquema conexión Relé

Como se observa en la ilustración 29, utilizamos un transistor BC337 al que conectaremos el pin GPIO del LPCXpresso a su pata base. La patilla emisora recibirá la tensión de 5v que además alimenta al relé y que protegeremos con un diodo 1N004. De esta forma cuando el pin GPIO de nuestro sistema se ponga en estado alto, el transistor cerrará el circuito y circularán los 5V por la bobina del relé activándolo, y por ende el dispositivo de iluminación o de alimentación.

Librería Relés:

El sistema cuenta con dos sencillas funciones: Ligths_Switch y Feeder_Switch. Ambas funciones reciben como parámetro 0 o 1. Si se recibe un 0 el sistema pone el pin de GPIO en estado bajo, desactivando el relé y de este modo apagando la luz o parando el alimentador. Si recibe un 1 realiza una acción opuesta, activando el relé y por tanto encendiendo la luz o activando el alimentador.

4.2 Software del sistema

4.2.1 Sistema operativo

Se ha elegido el sistema operativo FreeRTOS para el desarrollo del presente sistema. Ciertamente no tenemos ningún elemento en el entorno que requiera características exclusivas de un sistema operativo en tiempo real como FreeRTOS. Sin embargo su gran integración con las arquitecturas CORTEX, el soporte y amplia adopción del mismo, y la gran cantidad de documentación existente lo hace ideal para un proyecto de este estilo.

FreeRTOS es un sistema operativo en tiempo real para sistemas embebidos. Este agrega las principales características de un sistema como este, con un potente *scheduler* para gestión de tareas que además es configurable en su modo de operación (pre-emptive o cooperativo). Es independiente del hardware sobre el que corre, soportando más de 35 procesadores embebidos diferentes. Mecanismos para el paso de mensajes, gestión de colas, bloqueos de tareas, mutex, semáforos y un largo etc. de capacidades.

CMSIS:

CMSIS o Cortex Microcontroller Software Interface Standard, es una pieza de software que nos permite abstraer nuestro código de gestión de hardware del propio hardware que se ejecuta debajo del sistema. De esta forma conseguiremos una portabilidad excelente de nuestro código a cualquier otro sistema con una arquitectura de tipo Cortex-M.

El software CMSIS está pensado para utilizar el hardware de manera muy sencilla por los programadores, es totalmente compatible con los sistemas operativos Real Time, y proporciona las herramientas necesarias para la depuración de este.

Nuestro sistema se basa en la versión 2.0 del CMSIS

4.2.2 Aplicación del sistema:

Siguiendo la metodología descrita al comienzo del presente documento, se ha comenzado el desarrollo de las diferentes piezas que de forma conjunta conformarán el sistema completo.

Siempre se ha mantenido un modelo de capas en mente, en las que se ha diferenciado una capa de drivers o controladores del hardware todos ellos siempre basados en la capa de abstracción CMSIS, sobre esta se ha construido un grupo de librerías responsable de la gestión de los diferentes dispositivos para en la capa superior construir el grupo de tareas responsable de la funcionalidad del sistema. En la ilustración 37 se muestra un diagrama de bloques:

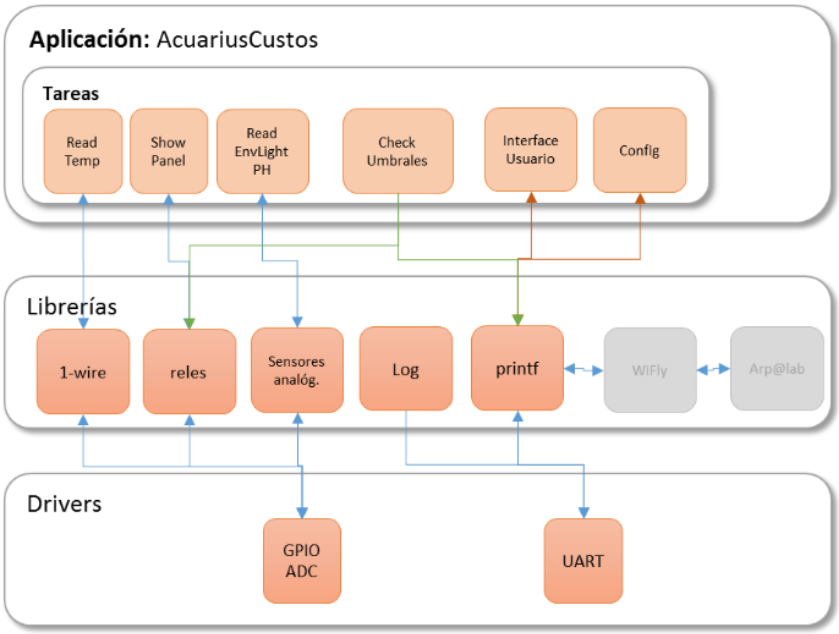


Ilustración 37 - Diagrama de bloques aplicación

La aplicación *Aquarius Custos* ha sido desarrollada con el IDE LPCXpresso de NXP en lenguaje C, usando la versión 7.5 de FreeRTOS y el CMSIS 2.0. Esta aplicación cuenta con 4 tareas principales (ver ilustración 33): una tarea responsable de gestionar el interface de usuario y manejar los menús y entradas desde el puerto serie, dos tareas responsable de las lecturas de los diferentes sensores, y una última tarea responsable de comprobar los umbrales de los sensores, y configuraciones realizadas y ejecutar las acciones pre-programadas.

- prvShowPanel: Tarea que gestiona el interface de usuario.
- prvReadEnvLightPH: Tarea responsable de leer sensores de tipo analógico, en concreto el sensor de luz ambiental y pH del agua.
- prvReadTemp: Tarea responsable de leer sensores digitales, en concreto la temperatura de la sonda DS18S20
- prvCheck: Tarea responsable de la comprobación de umbrales, y revisión de configuraciones. Es responsable de lanzar las alarmas y acciones programadas según la configuración.

En la ilustración 38 se muestra el flujo de la aplicación basado en tareas definidas en estas y las relaciones que entre ellas tienen.

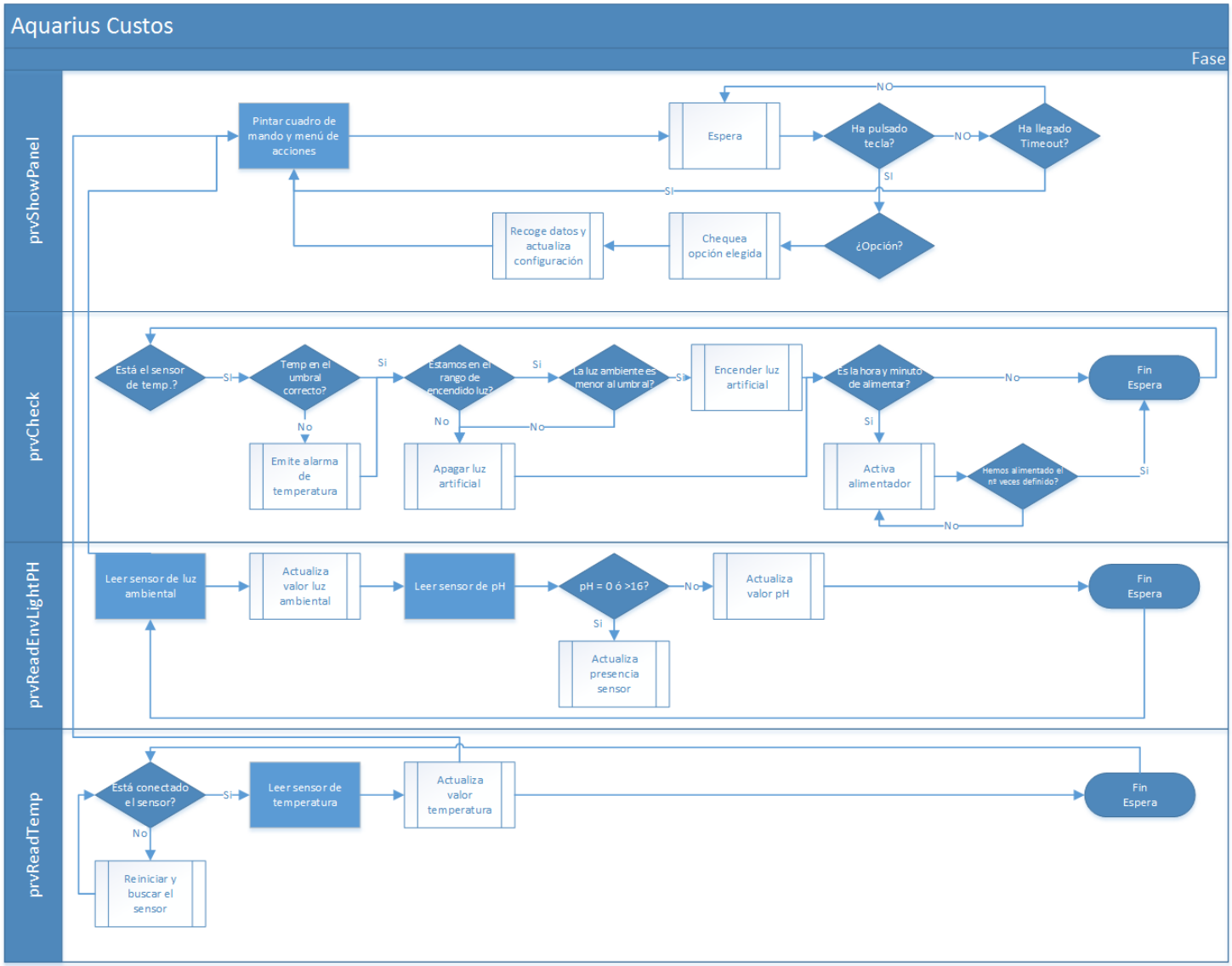


Ilustración 38 - Diagrama de flujo tareas

Interface de usuario:

Este interface de usuario está compuesto por una salida a través de puerto serie, el cual muestra utilizando códigos de control y caracteres ASCII un interface dibujado en pantalla con diferentes zonas. Este interface muestra información actual de diferentes parámetros, así como las diferentes configuraciones en vigor en ese momento.

El sistema cuenta con un menú en la parte inferior de la pantalla, el cual permite cambiar diferentes configuraciones del sistema, las opciones disponibles se muestran en la siguiente pantalla de la ilustración 39:

```
Menú:

[t] Cambiar umbrales de temperatura
[p] Cambiar umbrales de pH
[l] Gestión de luz artificial
[a] Gestión de alimentación

[f] Establecer fecha actual
```

Ilustración 39 - menu de configuración

Las opciones disponibles son:

- Tecla f: Permite establecer la fecha actual, así como la hora y minuto. A partir de este momento se utilizará el reloj RTC incluido en la placa LPCXpresso para mantener un reloj en tiempo real.
- Tecla t: Permite cambiar el umbral mínimo de temperatura desde 0 hasta 99,99ºC. El sistema admite la introducción de hasta cuatro caracteres, pensado para utilizar dos dígitos enteros, una coma o punto decimal y dos dígitos decimales.
- Tecla p: Permite cambiar los umbrales mínimo y máximo de pH ideales para el acuario. Este admite valores entre 0 y 14, con hasta dos dígitos decimales.
- Tecla l: Esta tecla permite configurar la iluminación artificial del acuario. En primer lugar nos solicitará desde que hora y minuto debe encenderse la luz artificial si es necesario, hasta que hora debe apagarse esta luz. Adicionalmente nos solicitará la cantidad de luz mínima a partir de la cual no es necesario encender la luz artificial ya que la de ambiente supera esa cantidad.
- Tecla a: Desde aquí configuraremos la alimentación de nuestro acuario. En primer lugar nos solicitará el número de veces que deseamos enviar alimentación al acuario en 24 horas naturales. Una vez introducido este dato, nos preguntará tantas veces como alimentaciones a qué hora y minuto queremos realizar la alimentación y cuantos pasos deseamos.

El alimentador utilizado es un sistema que proporciona una cantidad determinada de alimento en cada petición que se le hace y que hemos denominado “paso”. De esta manera, cada vez que se llega a esa hora y minuto el sistema ejecutará tantas alimentaciones como tengamos configurado. Este proceso se producirá de manera diaria.

Las diferentes configuraciones bien de umbrales, horas, pasos, etc. se guardan en variables globales o en registros de la propia placa LPCXpresso. Debido al uso único de valores no se ha estimado importante el uso de colas u otros mecanismos de comunicación entre procesos. Una vez realizada las acciones definidas en el menú estos valores se guardarán en una variable global con el valor mínimo, o máximo o cantidad configurada. Estas variables serán después consultadas por otras tareas como la responsable de las diferentes comprobaciones.

Lecturas de sensores:

Existen dos tareas diferenciadas para la lectura de sensores. Una destinada a la lectura de sensores analógicos y otra para sensores digitales. Las lecturas de sensores analógicos se realizan con una periodicidad de 1 segundo, y las de sensores digitales cada 10 segundos. Estas periodicidades son configurables en tiempo de compilación del proyecto.

La tarea de sensores analógicos es responsable de leer los sensores de pH y de luz ambiental. Esta tarea realiza la lectura de estos en un formato “raw” es decir sin conversión, el sensor del ADC entregará una serie de pasos en la lectura que deberemos convertir en función del número de bits del ADC así como de la tensión máxima que nos arrojará el sensor.

Para el sensor fotosensible de luz ambiente utilizaremos una tensión de 3,3v y una precisión de 4096 o 12bits, es decir cada unidad de la lectura del sensor ADC corresponde a (3,3v/4096) miliVolts. Esta lectura es la que utilizaremos como referencia para decidir si hay que encender o no la luz artificial posteriormente en función de los umbrales definidos.

El sensor de pH utiliza una lógica similar pero en base a una tabla de correspondencias de pH y miliVolts que se ha publicado como referencia en la ilustración 28.

En el caso del sensor de pH se ha utilizado un divisor de tensión basado en un transistor BSS-138, que nos entrega el voltaje de 3,3v ya que el dispositivo utilizado opera a 5v.

Comprobación:

Esta tarea tiene la responsabilidad principal de la vigilancia de los parámetros del acuario además de controlar las configuraciones definidas y llevar a cabo las acciones que correspondan en cada

caso. Aunque configurable en código, la aplicación realiza estas comprobaciones cada 2 segundos. Estas situaciones se describen a continuación:

- Notificación de alarmas: Las alarmas son mensajes de texto que se muestran en el interface de usuario en color rojo para denotar su urgencia o importancia. Son mensajes que se muestran de manera recurrente, de forma que aunque el interface se vuelva a pintar la alarma no desaparece de la pantalla. Se puede ver un ejemplo de alarmas en la siguiente ilustración 40.

```
[AVISO] El sensor de temperatura definido no se ha encontrado

[AVISO] El umbral de pH está fuera de rango. pH actual: 2.47, pH. mínimo: 7.00, pH.
 máximo: 7.80
```

Ilustración 40 - Mensajes de alarma

- Problemas con dispositivos: el sistema durante las lecturas de sensores controla la no presencia de este sensor en el caso de los sensores digitales, o de lecturas anómalas en el caso de sensores analógicos. Para ello dispone de una variable global por cada uno de los sensores que determina si este está operando correctamente o no. El sistema de comprobación realiza un chequeo en esta variable y en caso de error lo reporta mediante el sistema de alarmas.
- Alarma de temperatura: el sistema comprueba el valor de la última lectura almacenada en la variable global. Si esta está fuera del rango definido lanzará una alarma vía el interface de usuario.
- Alarma de pH: el sistema comprueba el último valor de la lectura de pH almacenada en la variable global. Si esta está fuera del rango definido lanzará una alarma vía el interface de usuario.
- Alimentación. Cuando llega el momento programado para activar el alimentador, se mostrará en pantalla el paso de alimentación en forma de mensaje: “Enviando alimentación X de Y”, indicándonos cuantas alimentaciones se van a realizar (Y) y que alimentación se está realizando en ese momento. Cada paso de alimentación dura 2 segundos.

Todas las tareas disponen de un #define global denominado DEBUG + nombre del módulo al que corresponde. Activando a 1 este #define todas las tareas apoyadas en la librería “log” enviarán mensajes de depuración al puerto serie configurado en los #define LOGSERIALPORT y LOGSERIALSPEED.

El sistema cuenta con un fichero de configuración modificable en fase de compilación, el cual permite definir los siguientes parámetros:

- Puerto serie y velocidad para el interface de usuario
- Puerto serie y velocidad para la salida de log de depuración
- Número de sensores digitales
- Puertos GPIO asignados a los sensores analógicos
- Número de serie para los sensores 1-wire
- Frecuencia de ejecución para cada tarea en milisegundos
- Número máximo de alimentaciones diarias por el sistema

5. Viabilidad técnica

El primer análisis de soluciones de mercado similares a la propuesta en el presente proyecto mostró que no existen muchas soluciones completas de características similares. Por el contrario si existen diferentes dispositivos que de manera independiente pueden conformar una solución completa: sensores de temperatura, pH, alimentadores, boyas de nivel, etc. sin embargo la no existencia de soluciones completas arrojaba ciertas dudas.

Durante este primer análisis en primer lugar se identificaron los componentes mínimos que debían formar parte de este sistema y su compatibilidad con la placa LPCXpreso elegida. Encontrándose todos ellos ya disponibles en un formato electrónico y 100% gestionables a través de los diferentes interfaces de conexión que nos ofrece la placa se realizó este análisis de viabilidad técnica.

Debido a las reducidas dimensiones de un Acuario es importante considerar otros aspectos de diseño como puede ser el tamaño del mismo o también aspectos como la potencial autonomía eléctrica de este.

Con todos estos factores se revisaron los siguientes aspectos:

- Funcionalidad de los elementos, nos ofrecen los datos que necesitamos y con la fiabilidad adecuada.
- Adecuación al medio, es un acuario y muchos deben funcionar en medio acuático
- Características técnicas de los elementos, el voltaje es adecuado, tengo capacidad para leer el dato con la precisión necesaria, existe soporte o drivers para él, etc.
- Coste del elemento, es un coste adecuado para una solución de este tipo.
- Mantenimiento y duración del elemento, habrá que reponerlo con frecuencia o se podrá utilizar a largo plazo.
- Existe ya un elemento que realice tareas complejas y simplifique nuestro sistema. Por ejemplo el alimentador era un caso adecuado, construimos uno en base a motores, o simplemente activamos uno existente.

La conclusión es que el sistema tiene una viabilidad técnica frente a todos los componentes que se han considerado parte del sistema.

Aspectos clave:

- Sencillez de integración de los sensores en la placa
- Sencillez de desarrollo y mantenimiento de código

- Ser un sistema en tiempo real
- Sistema ampliable con nuevos sensores
- Componentes muy económicos y de fácil reposición

Aspectos a mejorar:

- Acceso remoto a la plataforma vía internet
- Mantener un histórico de medidas
- Sensores adicionales como boyas de nivel, cloro o amoniaco.
- Cambiar el modelo de sensor de pH por su constante necesidad de re-calibración
- Alarmas luminosas
- Automatizar procesos como llenado, o corrección de algunos parámetros
- Mejorar el código y la gestión de errores y alarmas

6. Valoración económica

Los costes del proyecto se generan en dos ámbitos diferentes: uno primero dedicado a la adquisición de hardware necesario para las pruebas de concepto y ejecución del proyecto. Un segundo bloque basado en horas hombre dedicadas para la producción del código, pruebas de concepto, pruebas y documentación.

Hardware:

Los elementos descritos en la siguiente tabla, son elementos adquiridos en fabricantes de referencia del sector, y todos los precios son por cada unidad. También es importante entender que muchos de estos elementos han sido adquiridos a fabricantes Norte Americanos por lo que los precios son en $US y se ha realizado una conversión a € para mostrarlos en la siguiente tabla.

A continuación se puede encontrar un detalle de cada uno de los conceptos:

Descripción	Uds.	Precio Ud.	Sub-Total
Placa LPC1769 Rev. B	1	27,58€	27,58€
Placa de pruebas – 64 filas	2	6,91€	13,82€
Sensor fotoeléctrico	1	1,11€	1,11€
Sonda de temperatura resistente en agua DS18S20	1	6,35€	6,35€
Sensor pH SEN0169 DFRobot	1	52,44€	52,44€
Módulo Relay v3.1 JQC-3CC	2	4,42€	4,42€
Conversor lógico de niveles (5v-3,3v)	1	2,72€	2,72€
Dispensador de comida para peces RESUN AF-2003	1	8,50€	8,50€
Packs de cables (analógicos, digitales), cables cocodrilo, cables tipo jumper para placa de pruebas.	1	13,44€	13,44€
Resistencias (diferentes valores) / Transistores y Diodos	n/a	20€	20#
		TOTAL	**150,65€**

Tabla 3 - Costes de materiales

Para esta estimación se ha utilizado un factor de conversión de dólares a euros de 0,920843.

Mano de obra:

El esfuerzo realizado en mano de obra se describe en la siguiente tabla:

Descripción	Uds.	Precio Ud.	Sub-Total
Horas de desarrollo	240 h.	25,00€	27,58€
Pruebas y validación de entorno	60 h.	25,00€	13,82€
		TOTAL	**7.500,00€**

Tabla 4 - Costes mano de obra

Las horas de desarrollo se han considerado todas por el mismo importe, aunque la realidad es que en un proyecto real habrían sido diferentes perfiles (analista, programador, testing, etc.) con diferentes costes por hora de cada uno de ellos.

La tarifa hora elegida se ha realizado para una persona con un salario de 32.352,94€. El cálculo de la tarifa aproximada se ve en el siguiente cuadro:

Concepto	Importe
Salario Bruto	32.352,94€
Costes Seg. Social empresa (36% aprox.)	11.647,96€
Total coste anual	44.000,00€
Coste hora (estimamos 1760 horas año)	**25,00€/hora**

Los precios aquí mostrados son precios para el producto desarrollado como parte de este trabajo. En un modelo de producción masiva del sistema se considera que se podrían conseguir precios menores en todos los conceptos industrializando la producción de entre un 30% a un 40%.

7. Conclusiones

7.1 Conclusiones

Como inicio del proyecto se plantearon una serie de requisitos y objetivos de proyecto a cumplir. De estos objetivos detallamos a continuación los que se han finalizado con éxito:

- Desarrollar un sistema que mida la temperatura del agua del acuario de forma continua, vigilando que esté dentro de unos umbrales previamente establecidos, y alerte en caso que estos sean sobrepasados por exceso o defecto.
- Medir la acidez del agua (pH) vigilando que se encuentre dentro de unos rangos compatibles con la vida de los animales y plantas que viven en el acuario, alertando si el pH disminuye o supera los umbrales establecidos.
- Simular el ciclo día/noche en los horarios correspondientes de manera artificial.
- Provisión de alimentos automatizada de los animales del acuario.
- El usuario podrá configurar los umbrales vía interface disponible en el puerto serie. También se podrán configurar las notificaciones que se desean recibir.

De los objetivos secundarios se ha cumplido el siguiente:

- Control remoto del ciclo de día/noche y la alimentación, personalización de los umbrales y alertas.

Debido a la falta de tiempo han quedado pendientes del desarrollo de dos objetivos secundarios marcados al inicio del proyecto:

- Envío y consulta remota de las estadísticas del acuario: temperaturas históricas, evolución del pH, ciclos de día/noche registrando la intensidad de luz y alimentación dispensada.
- Envío de alarmas vía correo electrónico.

La principal enseñanza de este TFG es que sin un buen análisis y planificación inicial es complicado conseguir un resultado excelente. La planificación ha sido extremadamente útil para conseguir entregar en los plazos acordados y poder ir organizando el trabajo a desarrollar tanto en el tiempo como en el orden a ejecutarlo.

7.2 Propuesta de mejoras

El sistema sin duda es muy mejorable en varios aspectos, pero destacar tres mejoras fundamentales:

- Almacenar la configuración definida por el usuario. Cada vez que el sistema LPC pierde la alimentación, este pierde toda la configuración de umbrales, programaciones, etc. Sería importante que el sistema al arrancar sea capaz de recuperar la última configuración valida.
- Emitir alarmas visuales. El sistema además del interface de usuario disponible en el puerto serie, debería permitir visualmente ver la existencia de alarmas mediante leds de colores. De otra forma es complicado disponer de un equipo conectado al puerto serie al lado de un acuario de forma continua y darse cuenta de estas alarmas.
- Control remoto del sistema. Además del interface serie, sería importante poder acceder a los indicadores del sistema vía remota (web, aplicación móvil, o incluso telnet/ssh) lo que le dota de una utilidad muy superior.

7.3 Autoevaluación

Francamente ha sido un trabajo de fin de grado duro y con muchísima carga de trabajo. Pero por el contrario ha sido muy auto satisfactorio ya que ha permitido conocer toda la realidad de ejecutar un proyecto completo en primera persona, y no solo las tareas típicas de planificación o diseño, sino su construcción desde pequeños prototipos hasta configurar el sistema completo.

Ha servido para valorar aspectos que en ocasiones se le dan menor importancia de la que realmente tiene como puede ser un análisis funcional o casos de uso, con el objeto de realizar un muy buen diseño desde el inicio y detectar carencias del sistema como las descritas en el punto 7.2

Personalmente me apasiona la programación con microcontroladores, y añadir el factor de sistema en tiempo real y gestionar dispositivos conectados a él de varios tipos, como digo ha sido simplemente apasionante y muy motivante, hasta el punto que las mejoras detectadas en el punto 7.2 las terminaré de implementar en mi tiempo libre, e incluiré mejoras como los sensores de nivel de agua.

8. Glosario

- **1-wire:** protocolo de comunicaciones digital para dispositivos de un solo hilo.
- **ADC:** Conversor analógico a digital.
- **ARM:** Diseñador de arquiteturas para procesadores y microcontroladores.
- ASCII: (American Standard Code for Information Interchage) Estándar Americano codificado para el intercambio de información.
- **BaudRate:** velocidad de comunicación de un puerto serie.
- **BreadBoard:** placa de pruebas que permite de forma sencilla interconectar dispositivos electrónicos.
- **Cortex-M3:** Arquitectura de procesadores del fabricante ARM.
- **CPU:** Unidad central de proceso, es el punto central de procesamiento de cualquier dispositivo embebido o informático.
- **CAN:** (Controller Area Network) Controlador de red de área local.
- **CMSIS:** (Cortex Microcontroller Software Interface Standard) capa de abstracción para la gestión del hardware de procesadores con arquitectura Cortex de ARM.
- **Cooperativo:** Modelo de organización de tiempo de procesador para diferentes tareas. Cada tarea es responsable de ceder el tiempo de procesador.
- **DIP:** formato de interconexión de dispositivos electrónicos.
- **Diodo:** dispositivo electrónico de dos electrodos que permite el paso de tensión en un único sentido. Estos electrodos pueden ser luminosos.
- **DMA:** (Direct Memory Access) modelo de acceso a memoria directo desde periféricos del sistema.
- **Drivers:** software que actúa como intermediario especifico de un periférico y una funcionalidad del sistema.
- **Ensamblador:** Lenguaje de programación a través de nombres predefinidos de instrucciones del lenguaje máquina, permite la programación de procesadores.
- **Framework:** entorno de desarrollo que permite entregar una funcionalidad predefinida de forma personalizada.
- **FreeRTOS:** Sistema operative para sistemas embebidos que es capaz de operar en tiempo real.
- **GPIO:** (General Purpouse Input Output) Puertos de acceso de la placa de propósito general de entrada y salida. Estos puertos pueden ser configurados con diferentes funciones.

- **I2C:** bus de comunicaciones de tipo serie.
- **IDE:** (Integrated development environment) entorno de desarrollo integrado
- **LPC1769:** Modelo de procesador con arquitectura Cortex diseñada por ARM
- **LPCXpresso:** Tipo de placa para ejecutar procesadores de tipo Cortex fabricada por NXP, con capacidades especiales para la depuración de software.
- **Microcontrolador:** Circuito capaz de ejecutar instrucciones almacenadas en una memoria no volátil y con capacidad para conectar periféricos.
- **Microprocesador:** dispone de las mismas características de un microcontrolador, pero agrega unidades de mayor complejidad como una unidad aritmética lógica, control de buses, modelos de acceso directo a memoria, interrupciones, etc.
- **Mutex:** (Mutual Exclusion) método aplicable por los desarrolladores para evitar situaciones de concurrencia e inconsistencia de datos, mediante la exclusión de acceso puntualmente.
- **NVIC:** (Nested Vector Interrupt Controller) Vector Controlador de Interrupciones Anidado.
- **NXP:** Fabricante de circuitos integrados.
- **pH:** Indicador que marca la acidez o basicidad de una solución acuosa.
- **Pre-emptivo:** modelo de gestión del tiempo de procesador en que el sistema operativo es responsable de asignar el tiempo de este.
- **Putty:** terminal serie
- **Relé:** dispositivo electrónico que abre/Cierra un circuito en base a la aplicación de una tensión.
- **Reset:** Reinicio de un sistema
- **Scheduler:** (Planificador) sistema responsable de asignar tiempo de procesador a diferentes tareas.
- **SSD:** (Solid State Disk) disco de estado solido.
- **Transistor:** dispositivo electrónico semiconductor que entrega una señal de salida en función a una señal de entrada.
- **Top-Down:** metodología de desarrollo
- **UART:** (Universal Asynchronous Receiver-Transmiter) receptor transimisor universal asincrono. También se conocen como Puerto serie.

9. Bibliografía

Libros:

- NXP Systems (rev. 3.1 – 02/04/2014), **LPC_User_Manual-UM10360: Manual de fabricante para usuarios del procesador LPC176x/5x**
- José Daniel Muñoz Frías (Primera edición, 2009), **Sistemas empotrados en tiempo real**
  ISBN: 978-84-612-9902-7
- Richard Barry – Real Time Engenieers © (2011), **Using the FreeRTOS Real Time Kernel - A Practical Guide**
  ISBN: ISBN 978-1-4461-6997-1
- Microchip Technology Incorporated (2014), **WiFly Command Reference, Advanced Features and Applications User's Guide**
  ISBN: 978-1-6207-7841-8
- Maxim Integrated Products, Inc. (2015), **DS18S20, High-Precision 1-Wire Digital Thermometer**

Páginas WEB:

- **Historia del mercado de sistemas embebidos:**
  https://en.wikipedia.org/wiki/Embedded_system
- **Estudio de mercado sistemas embebidos:**
  http://www.strategyr.com/MarketResearch/Embedded_Systems_Market_Trends.asp
- **NXP LPC Microcontrollers Datasheet:**
  http://www.nxp.com/documents/brochure/75017372.pdf
- **LPCWare Foros:**
  https://www.lpcware.com/forum
- **FreeRTOS:**
  http://www.freertos.org/RTOS.html
- **CMSIS:**
  http://www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php
- **Guías UOC:**
  http://materials.cv.uoc.edu/continguts/PID_00158467/index.html?ajax=true
  http://cv.uoc.edu/webapps/xwiki/wiki/matembeddedsystemslabhome/view/Material/IniciCortexM3

10. Anexos

10.1 Manual de usuario

10.1.1 Conexión de dispositivos

El sistema soporta en la actualidad los siguientes dispositivos:

Dispositivo	Descripción	PIN	Dispositivo homologado
Sensor de temperatura	Sensor de temperatura operando a 3,3v y con un único cable de datos en formato digital y soporte para protocolo de comunicaciones 1-wire		DS18S20
Sensor pH	Sensor de pH operando a 5,5v con un cable de datos de tipo analógico, operando entre 0v y 3,3v.		SEN0161 de DFRobot
Rele alimentación	Relé operando a 5v la bobina, y con capacidad para gestionar hasta 240v y 10A		Cualquiera
Relé luz artificial	Relé operando a 5v la bobina, y con capacidad para gestionar hasta 240v y 10ª		Cualquiera
Alimentador eléctrico	Alimentador eléctrico para peceras. Debe ser capaz de alimentar cada vez que recibe tensión.	Al relé de Alim.	RESUN AF-2003

Tabla 5 - Periféricos soportados

El sistema dispone de una conexión de 5v y de 3,3v en los que se conectarán respetando la polaridad los cables de alimentación de cada uno de los periféricos.

La placa LPCXpresso dispone de un sistema de conexionado con la siguiente correspondencia en el PINOUT, se utilizará este esquema para conectar el PIN de datos del dispositivo (ver Tabla 3) al PIN correspondiente de la placa LPCXpresso:

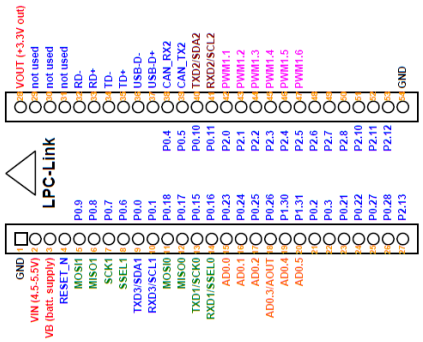


Ilustración 41 - PINOut de la placa LPCXpresso

Si se desea acceder al interface de usuario, se deberá conectar al puerto serie configurado (ver apartado 10.2) un puerto serie desde el PC del usuario, o bien conectar el dispositivo CP2102 que realizará la conversión de puerto serie a USB. Los PIN de conexionado para cada puerto se pueden ver en la ilustración 37.

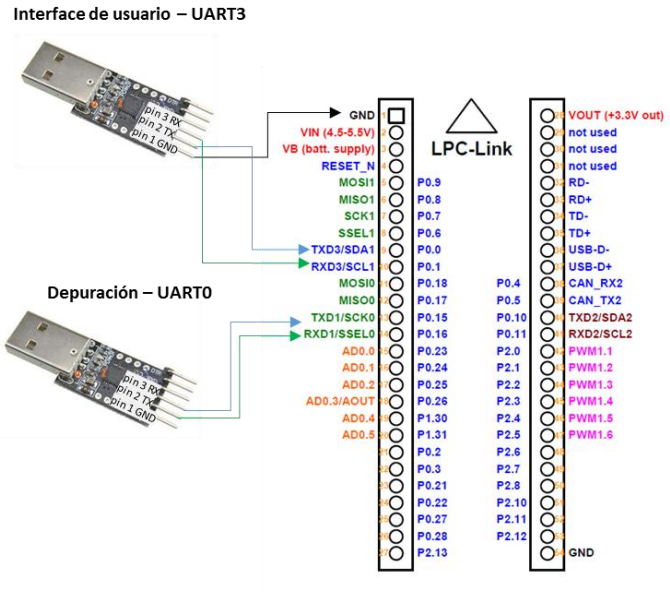


Ilustración 42 - Esquema de conexión CP2102

10.1.2 Uso de la aplicación

El uso del sistema una vez subida la aplicación a la placa LPCXpresso es bastante sencillo siguiendo estos pasos:

1. Configuración de fecha y hora: utilizando la opción “Tecla f”, introduciremos por este orden los datos de año, mes, día, hora y minuto actual. Una vez introducido podremos ver en la parte superior la hora actual. Esta hora se refresca con cada refresco de pantalla.
2. Configuración del sistema de luz artificial. Dependiendo de la ubicación del acuario tendremos que configurar una sensibilidad a la luz ya que variará mucho si está al lado de una ventana con luz natural o si está en una zona oscura.

   Pulsaremos la “Tecla l” la cual no solicitará en primer lugar la cantidad de luz a partir de la cual el sistema no debe encenderse aunque se encuentre en el rango horario de encendido. A continuación solicitará la hora y minuto de encendido y la hora y minuto de apagado.

3. Configuración de la alimentación. Para finalizar utilizaremos la “Tecla a” la cual nos permitirá configurar como vamos a alimentar a nuestro acuario.
   En primer lugar nos solicitará cuantas alimentaciones queremos enviar al cabo del día. Después por cada alimentación nos pedirá tres datos: hora, minuto y número de pasos. Cada paso corresponde a una cantidad pre-definida de comida en el alimentador.

A partir de este momento el sistema está 100% configurado y operando. Cada perdida de alimentación de la placa deberemos realizar estos tres primeros pasos.

Opcionalmente el sistema nos permite configurar los umbrales de vigilancia para otros parámetros que se describen a continuación:

4. Pulsando la “Tecla t” nos solicitará la temperatura mínima y la temperatura máxima que si son traspasadas el sistema nos alertará. Los valores permitidos son de 0 a 99 con dos decimales.
5. Pulsando la “Tecla p” nos solicitará el valor mínimo del pH y máximo del mismo. Los valores permitidos son desde 0 a 14 con dos decimales.

10.2 Ejecución y compilación

10.2.1 Entorno de desarrollo y compilación

Se ha utilizado el IDE de desarrollo de NXP denominado LPCXpresso. El proyecto de software se ha comprobado su funcionamiento con los IDE de LPCXpresso versiones de 7.5.0 hasta la 8.0.0.

En primer lugar deberemos importar el proyecto dentro del IDE, para ello seguiremos los siguientes pasos:

Elegiremos en el menú *File* la opción *Import*, ofreciéndonos el siguiente menú del que escogeremos la opción “Existing projects into WorkSpace”:

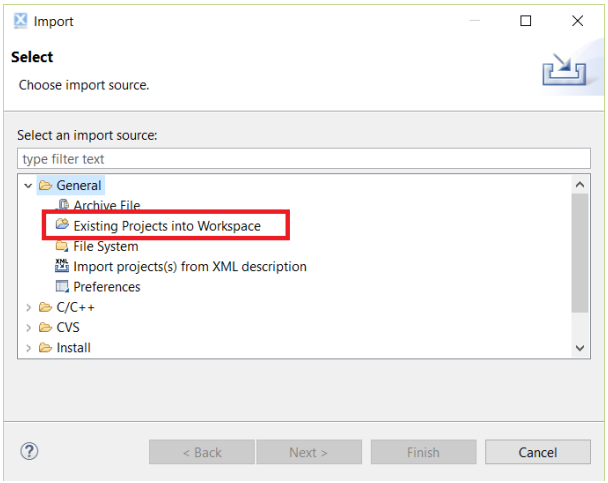


Ilustración 43 - Importar proyectos en LPCXpresso

A continuación elegiremos la opción “Select archive file” y escogeremos el archivo comprimido con el código del proyecto a importar. Tras este punto nos mostrará todos los proyectos que hay en el archivo comprimido. Seleccionaremos todos los proyectos antes de pulsar “Finish”:

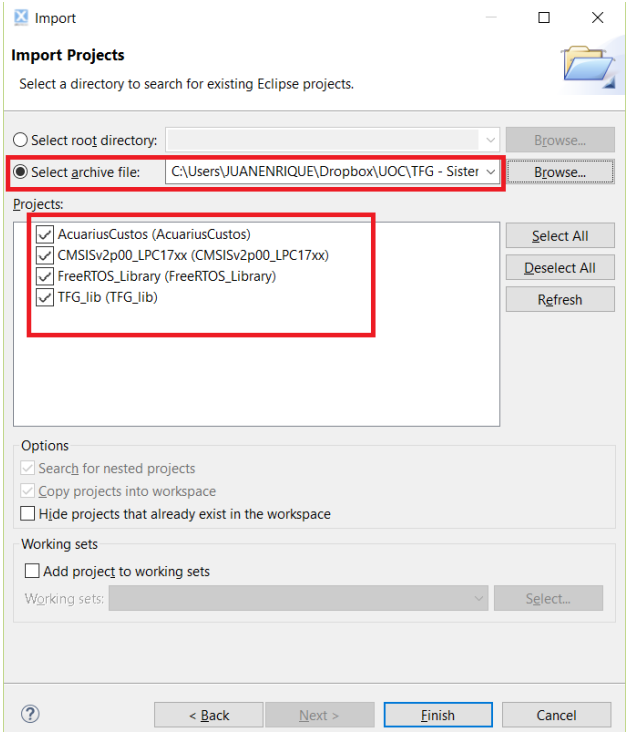


Ilustración 44 - Selección de proyectos

A partir de este momento podremos ver en el apartado “Project Space” los diferentes proyectos que hemos importado:

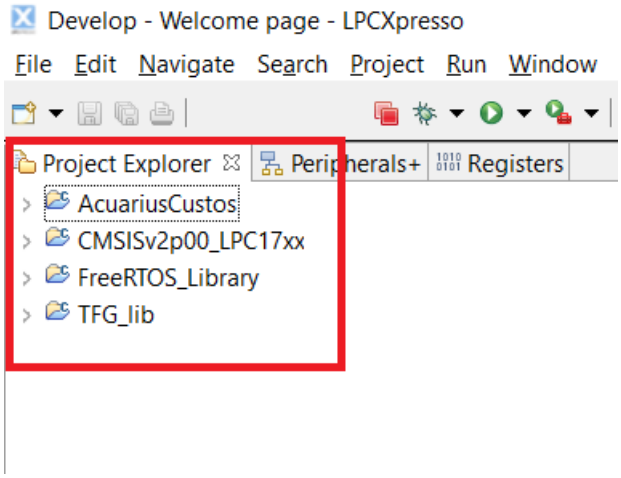


Ilustración 45 - Proyectos disponibles en el IDE

Compilación:

Para proceder a compilar el proyecto, sobre la línea “AcuariusCustos” en la ventana Project Explorer pulsaremos el botón secundario del ratón y sobre el menú contextual elegiremos la opción “Build Project”:

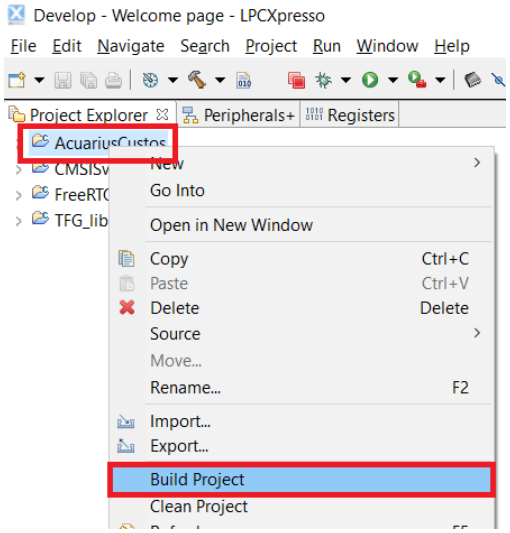
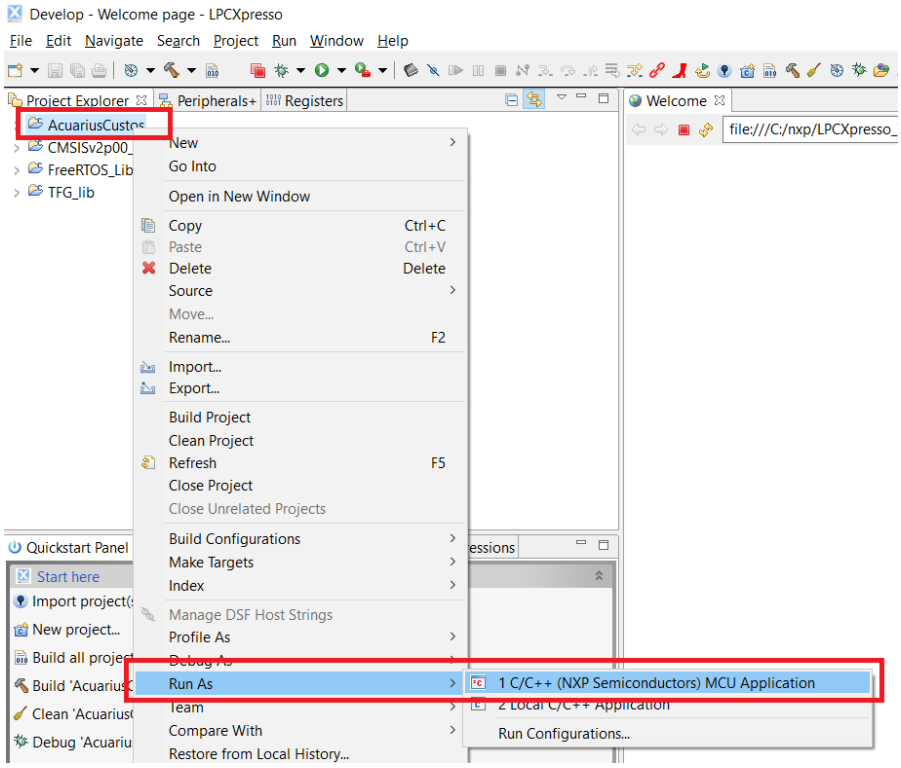


Ilustración 46 - Compilación del proyecto

Esta opción compilará todos los módulos que forman parte del proyecto, librería *CMSIS*, sistema operativo FreeRTOS, librería TFG_lib y por último el software *Acuarius Custos*.

Ejecutar:

Para ejecutar el código, en primer lugar el sistema subirá la aplicación en formato binario ya compilada y linkada a la placa LPCXpresso para su ejecución. De esta manera la próxima vez que la placa se reinicie se ejecutará este código.

Para grabar el software en la placa y ejecutar este, elegiremos en la ventana de “Project Explorer” el proyecto Acuarius Custos y con el botón secundario del ratón elegiremos la opción: “Run As” y la opción secundaria “1 C/C++ (NXP Semiconductors) MCU Application”. Se muestra el menú en la imagen 48: