

UNIVERSITAT OBERTA DE CATALUNYA

Ingeniería Técnica de Telecomunicación

Especialidad Telemática

DISEÑO DE UN SUPERVISOR REMOTO DE TEMPERATURA

VÍA BLUETOOTH

MEMORIA

Alumno: Fco. Javier Hernando García

Dirigido por: Carlos Pacheco

CURSO 2015-16 (Febrero/Septiembre)

Índice de contenidos

1.	Introducción	6
1.1.	Justificación del TFC	7
1.2.	Objetivos	7
1.3.	Enfoque	8
1.4.	Planificación	9
1.4.1.	Calendario de trabajo	9
1.4.2.	Hitos	10
1.4.3.	Tareas y actividades	11
1.4.4.	Diagrama de Gantt	13
1.5.	Riesgos y plan de contingencia	15
1.5.1.	Incidencias con el punto de trabajo	15
1.5.2.	Dificultad del TFC.....	15
1.5.3.	Modificación de las condiciones laborales.....	15
1.5.4.	Indisposición médica	15
1.6.	Productos obtenidos	16
1.7.	Breve descripción de la memoria.....	16
2.	El módulo Bluetooth	17
2.1.	Introducción a la tecnología Bluetooth.....	17
2.2.	Análisis de mercado	19
2.3.	Nuestro módulo Bluetooth	20
3.	El sensor de temperatura y circuito de adaptación	21
3.1.	Introducción a los sensores de temperatura	21
3.2.	Análisis de mercado	22
3.3.	Nuestro sensor de temperatura.....	22
3.4.	Diseño del circuito de adaptación de señales.....	23
4.	El microcontrolador.....	29
4.1.	Introducción a los microcontroladores	29
4.2.	Estudio de mercado	29
4.3.	Nuestro microcontrolador	30
4.4.	Diseño del programa de control	31
4.5.	Simulación del programa de control.....	35
5.	La fuente de alimentación.....	38
5.1.	Introducción a las fuentes de alimentación.....	38
5.2.	Análisis de consumo.....	38

5.3.	Diseño de la fuente de alimentación	39
5.3.1.	Diseño de la fuente de + 15 V	39
5.3.2.	Diseño de la fuente de - 15V	40
5.3.3.	Diseño de la fuente de 3,3 V	41
6.	Diseño del <i>layout</i> de la PCB.....	43
6.1.	Esquemático	44
6.2.	<i>Layouts</i> de la PCB	45
6.2.1.	Disposición de los componentes en la PCB.....	51
6.2.2.	Trazado de las pistas	52
6.3.	Valoración económica	53
7.	Conclusiones y ampliaciones.....	54
7.1.	Reflexión sobre los objetivos	54
7.2.	Problemas surgidos	54
7.3.	Soluciones aportadas	55
7.4.	Lecciones aprendidas	55
7.5.	Líneas de ampliación futuras	56
	Glosario	57
	Bibliografía	59

Índice de tablas

Tabla 1: Calendario de trabajo	10
Tabla 2: Hitos.....	10
Tabla 3: Tareas y actividades.....	12
Tabla 4: Clases de Bluetooth	17
Tabla 5: Versiones de Bluetooth	17
Tabla 6: Análisis de consumo	38
Tabla 7: Pineado del RN42XVP	49
Tabla 8: Valoración económica del supervisor remoto de temperatura vía Bluetooth	53

Índice de ilustraciones

Ilustración 1: Diagrama de bloques.....	8
Ilustración 2: Diagrama de Gantt	14
Ilustración 3: Módulo Bluetooth Microchip RN42XVP-I/RM	20
Ilustración 4: RTD PT100	21
Ilustración 5: Termostato bimetalico	21
Ilustración 6: Termistores PTC y NTC	21
Ilustración 7: LM35.....	21
Ilustración 8: Sonda de temperatura basada en termopar.....	22
Ilustración 9: Sensor de temperatura	22
Ilustración 10: Simulación del puente de Wheatstone a la temperatura mínima.....	24
Ilustración 11: Simulación del puente de Wheatstone a la temperatura máxima	24
Ilustración 12: Amplificador operacional en configuración de ganancia diferencial.....	25
Ilustración 13: Simulación de nuestro primer circuito de adaptación de señal de temperatura	27
Ilustración 14: Amplificador operacional en modo seguidor de tensión.....	27
Ilustración 15: Simulación del circuito de adaptación de señales a la temperatura máxima.....	28
Ilustración 16: Simulación del circuito de adaptación de señales a la temperatura mínima	28
Ilustración 17: Microchip PIC16F723	31
Ilustración 18: Programa de control	35
Ilustración 19: Simulación en Proteus del programa de control.....	36
Ilustración 20: Esquema de la fuente de alimentación de + 15 V.....	39
Ilustración 21: Eficiencia de la fuente de + 15 V	40
Ilustración 22: Esquema de la fuente de alimentación de - 15 V.....	41

Ilustración 23: Eficiencia de la fuente de - 15 V	41
Ilustración 24: Esquema de la fuente de alimentación de 3,3 V.....	42
Ilustración 25: Eficiencia de la fuente de 3,3 V	42
Ilustración 26: Esquemático de nuestro supervisor remoto de temperatura	44
Ilustración 27: Dimensiones en pulgadas (y en milímetros) del LM2674	45
Ilustración 28: Pineado del LM2674.....	45
Ilustración 29: LM2674.....	45
Ilustración 30: Dimensiones en milímetros del LMR16006	46
Ilustración 31: Pineado del LMR16006	46
Ilustración 32: LMR16006	46
Ilustración 33: LMZ34002.....	46
Ilustración 34: Pineado del LMZ34002.....	46
Ilustración 35: Dimensiones en milímetros del LMZ34002.....	47
Ilustración 36: Dimensiones en pulgadas del PIC16F723.....	48
Ilustración 37: Pineado del PIC16F723.....	48
Ilustración 38: Dimensiones en milímetros del RN42XVP.....	49
Ilustración 39: Diseño de la PCB de nuestro supervisor de temperatura	50
Ilustración 40: Disposición de los componentes en la PCB de nuestro supervisor remoto de temperatura	51
Ilustración 41: Trazado de las pistas en la PCB de nuestro supervisor remoto de temperatura	52

1. Introducciónⁱ ⁱⁱ

Este documento constituye la última de las cuatro entregas evaluables de la asignatura denominada Trabajo Final de Carrera, en adelante TFC, que forma parte del plan de estudios de la Ingeniería Técnica en Telecomunicaciones (especialidad en Telemática), en adelante ITTT.

En esta asignatura hemos llevado a cabo un proyecto en el que nosotros mismos hemos sido el único miembro del equipo de trabajo que ha tenido que desempeñar todos los roles. Hemos sido partícipes de las cinco fases del ciclo de vida de un proyecto: iniciación, planificación, ejecución, control y seguimiento, y cierre.

La primera entrega respondía a la necesidad de establecer un plan de trabajo que nos marcara el camino a seguir: qué se hace y qué no se hace, cuándo, cómo y por qué. Así pues, comenzamos hablando de los objetivos que se pretendían alcanzar con la realización de este proyecto.

A continuación comentamos de una manera breve el equipo que teníamos que diseñar en este proyecto, la estructura del contenido de la memoria final y cómo sería su distribución en páginas según los requisitos del encargo.

Después planificamos nuestro proyecto en el tiempo, establecimos un calendario de trabajo, identificamos los principales hitos y tareas, y descompusimos estas últimas en actividades a las que asignamos un tiempo y fechas de realización.

Para finalizar la primera entrega, enumeramos los materiales necesarios para la realización de este proyecto, identificamos los riesgos que podían poner en peligro nuestra planificación temporal y propusimos medidas que mitigasen o corrigiesen los efectos que pudieran causar.

La segunda entrega partía de ese documento, del que extrajimos las partes más importantes correspondientes a la planificación y añadimos los contenidos planificados para la PEC2: una pequeña introducción a la tecnología Bluetooth y la elección del módulo Bluetooth para nuestro supervisor; un repaso de los distintos sensores de temperatura, la elección de uno de ellos y el diseño del circuito de adaptación de señales previo al conversor analógico-digital; y una breve introducción a los microcontroladores y la elección de uno de ellos.

En la tercera entrega añadimos a la anterior los contenidos planificados para la PEC 3: acabamos el capítulo dedicado al microcontrolador implementado el programa de control y realizando la simulación del mismo, realizamos el diseño de la fuente de alimentación del supervisor y finalizamos el documento con el diseño del *layout* de la PCB.

Para elaborar esta memoria final hemos partido de esa tercera entrega a la que hemos añadido el capítulo de las conclusiones y hemos realizado una profunda revisión de contenidos y estilos.

1.1. Justificación del TFC

Tecnologías inalámbricas como WiFi, WiMax, GPRS, HSDPA o LTE, nos permiten tener acceso a Internet en cualquier momento y lugar, algo que se ha convertido en imprescindible en nuestro estilo de vida en el mundo actual en el que nos ha tocado vivir.

Otras tecnologías inalámbricas de corto alcance, como Bluetooth o NFC, también nos permiten la conexión a otros equipos para el acceso a la información o el intercambio de datos de muy diversa índole.

Este proyecto consiste en diseñar un equipo que recogerá información de temperatura y la transmitirá utilizando la tecnología Bluetooth.

1.2. Objetivos

Los objetivos de esta asignatura son:

- Aplicar en el ámbito práctico los conocimientos teóricos adquiridos de las diferentes asignaturas del plan de estudios de la ITTT.
- Desarrollar la habilidad de la comunicación escrita mediante la producción de escritos de longitud media que combinen textos y elementos gráficos que faciliten la comprensión del lector y capten su atención.
- Aprender a identificar los objetivos de un proyecto cualquiera que sea su ámbito y conseguir su resolución.
- Analizar los resultados obtenidos y los problemas encontrados, extraer conclusiones, aportar soluciones y proponer líneas de ampliación futuras.

Los objetivos de este TFC en concreto son:

- Diseñar un supervisor remoto de temperatura vía Bluetooth.
- Diseñar circuitos analógicos que realicen funciones de amplificación, comparación y generación de señal.
- Diseñar circuitos de adaptación de las entradas y salidas a niveles del microcontrolador.
- Diseñar fuentes de alimentación óptimas.
- Programar procesadores digitales de la señal, en su lenguaje de programación, para la manipulación y procesamiento de las señales.

1.3. Enfoque

Como ya hemos comentado en el punto 1.1, vamos a diseñar un supervisor remoto de temperatura vía Bluetooth que monitoriza continuamente la temperatura y permite el acceso remoto a esos datos.

Nuestro supervisor se compondrá de los siguientes bloques:

- Un bloque de alimentación que recibirá la tensión de una fuente externa y que generará las tensiones de alimentación que necesiten el resto de bloques.
- Un bloque de adaptación de señales que adaptará la señal que reciba del sensor de temperatura para que el microcontrolador la pueda procesar.
- Un microcontrolador que gestiona el procesado de la temperatura y el control del módulo Bluetooth.
- Un módulo Bluetooth que transmitirá los datos de temperatura.

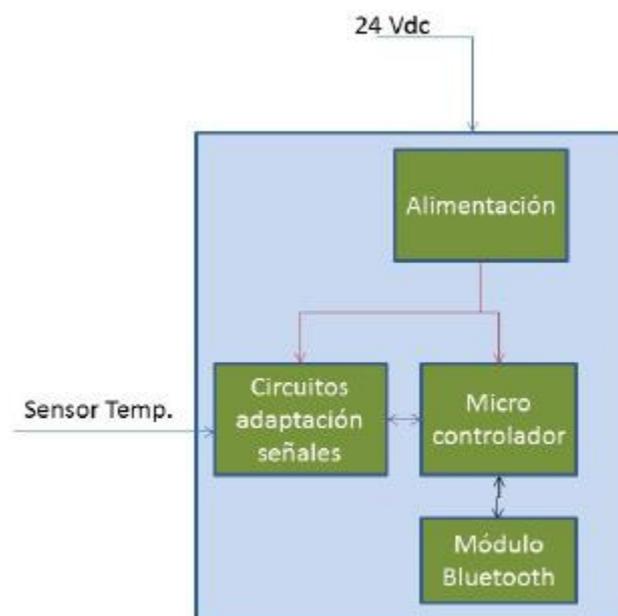


Ilustración 1: Diagrama de bloques

1.4. Planificación^{iii iv v vi}

Para una correcta planificación de nuestro proyecto vamos a elaborar un calendario de trabajo que contemple las horas y días previstos de dedicación, así como aquellos días festivos o vacacionales en los que ya sabemos de antemano que no vamos a dedicar nada de tiempo a la ejecución del TFC.

A continuación identificaremos los principales hitos y tareas del proyecto, y las dividiremos en subtareas para hacer una estimación del tiempo que nos llevará cada una de ellas.

1.4.1. Calendario de trabajo

Siempre intentaremos conciliar la vida laboral, familiar y académica, y dado que el TFC es la única asignatura de la que nos hemos matriculado este semestre, prevemos dedicar a la misma un tiempo que, como mínimo, será el siguiente:

- De lunes a viernes: 2 horas al día.
- Sábados, domingos y otros festivos: 4 horas al día.

Esto supone una media de 18 horas semanales, cantidad que puede verse aumentada o disminuida en función de los festivos que caigan de lunes a viernes o si tenemos previsto ausentarnos por vacaciones.

Como estamos hablando de tiempos mínimos que nos comprometemos a cumplir, esto nos permitirá dedicar más horas en momentos puntuales en los que por cualquier circunstancia no hayamos podido cumplir el horario previsto.

El calendario de trabajo será el siguiente:

Febrero							
L	M	X	J	V	S	D	Horas
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	14
29							2

Marzo							
L	M	X	J	V	S	D	Horas
	1	2	3	4	5	6	16
7	8	9	10	11	12	13	18
14	15	16	17	18	19	20	18
21	22	23	24	25	26	27	6
28	29	30	31				6

Abril							
L	M	X	J	V	S	D	Horas
				1	2	3	10
4	5	6	7	8	9	10	18
11	12	13	14	15	16	17	18
18	19	20	21	22	23	24	18
25	26	27	28	29	30		10

Mayo							
L	M	X	J	V	S	D	Horas
						1	
2	3	4	5	6	7	8	16
9	10	11	12	13	14	15	18
16	17	18	19	20	21	22	18
23	24	25	26	27	28	29	18
30	31						4

Junio							
L	M	X	J	V	S	D	Horas
		1	2	3	4	5	14
6	7	8	9	10	11	12	18
13	14	15	16	17	18	19	4
20	21	22	23	24	25	26	
27	28	29	30				

Total	264
--------------	------------

	Inicio del TFC
	Días libres o vacaciones
	Entrega de la memoria y presentación

Tabla 1: Calendario de trabajo

1.4.2. Hitos

Los hitos se corresponden con cada una de las entregas parciales y finales programadas en este TFC.

Aprovecharemos la posibilidad que tenemos de poder fijar las fechas de entrega de las PEC2 y PEC3, para desplazarlas una semana por detrás respecto a las fechas previstas originalmente.

Aunque nos quitemos tiempo para la preparación de la memoria final y de la presentación, podremos dedicar más tiempo a la parte práctica del TFC que es mucho más compleja.

Hitos	Fecha
Inicio TFC	24/02/2016
Entrega del borrador del Plan de Trabajo	06/03/2016
Entrega del Plan de Trabajo	08/03/2016
Entrega del borrador de la PEC2	10/04/2016
Entrega de la PEC2	12/04/2016
Entrega del borrador de la PEC3	15/05/2016
Entrega de la PEC3	17/05/2016
Entrega del borrador de la memoria y presentación	12/06/2016
Entrega de la memoria y presentación	14/06/2016
Inicio debate virtual	29/06/2016
Fin debate virtual	01/07/2016
Fin TFC	01/07/2016

Tabla 2: Hitos

1.4.3. Tareas y actividades

A continuación vamos a descomponer cada una de las tareas propuestas en el enunciado en otras subtareas o actividades de menor nivel e indicaremos para cada una de ellas: el tiempo estimado en horas para su realización, las fechas de inicio y de fin estimadas para cada una, y cuáles son las actividades predecesoras que debemos completar para poder iniciarla.

Nº	Tarea / Actividades	Predecesora	Horas	Fecha inicio	Fecha final
1	Definición del TFC			24/02/2016	29/02/2016
1.1	Descarga de los materiales del aula virtual		1	24/02/2016	24/02/2016
1.2	Lectura del plan de estudios, enunciado del TFC y otros materiales	1.1	10	25/02/2016	28/02/2016
1.3	Encuentro inicial	1.2	1	29/02/2016	29/02/2016
2	Plan de trabajo (PEC1)			01/03/2016	08/03/2016
2.1	Redacción del borrador	1	11	01/04/2016	04/03/2016
2.2	Entrega del borrador	2.1	1	05/03/2016	05/03/2016
2.3	Corrección del borrador	2.2	5	06/03/2016	07/03/2016
2.4	Entrega del Plan de trabajo	2.3	1	08/03/2016	08/03/2016
3	Estudio de mercado de los módulos Bluetooth			09/03/2016	20/03/2016
3.1	Búsqueda de información sobre la tecnología Bluetooth	2	12	09/03/2016	14/03/2016
3.2	Análisis de mercado de módulos Bluetooth	3.1	10	15/03/2016	19/03/2016
3.3	Elección del módulo Bluetooth	3.2	2	20/03/2016	20/03/2016
4	Diseño del circuito de adaptación de temperatura			21/03/2016	04/04/2016
4.1	Análisis de mercado de sensores de temperatura	3	6	21/03/2016	23/03/2016
4.2	Elección del sensor de temperatura	4.1	2	29/03/2016	29/03/2016
4.3	Instalación y estudio de la herramienta de simulación de circuitos	4.2	4	30/03/2016	31/03/2016
4.4	Diseño del circuito de adaptación	4.3	8	01/04/2016	04/04/2016
5	Elección del procesador e implementación del programa de control			05/04/2016	07/05/2016
5.1	Análisis de mercado de microprocesadores	4	10	05/04/2016	09/04/2016
5.2	Elección del microprocesador	5.1	2	10/04/2016	10/04/2016
6	PEC2			09/03/2016	12/04/2016
6.1	Redacción del borrador	3, 4, 5.1, 5.2	11	09/03/2016	10/04/2016
6.2	Entrega del borrador	6.1	1	10/04/2016	10/04/2016
6.3	Corrección del borrador	6.2	5	11/04/2016	11/04/2016
6.4	Entrega de la PEC2	6.3	1	12/04/2016	12/04/2016

Nº	Tarea / Actividades	Predecesora	Horas	Fecha inicio	Fecha final
7	Diseño de la fuente de alimentación			13/04/2016	21/04/2016
7.1	Estudio de la fuente de alimentación	6	6	13/04/2016	15/04/2016
7.2	Análisis de consumos	7.1	6	16/04/2016	18/04/2016
7.3	Diseño de la fuente	7.2	6	19/04/2016	21/04/2016
5.3	Instalación y estudio de la herramienta de programación de PIC	5.2, 7	8	22/04/2016	24/04/2016
5.4	Diseño y simulación del programa de control	5.3	20	25/04/2016	07/05/2016
8	Diseño del <i>layout</i> de la placa PCB			08/05/2016	15/05/2016
8.1	Instalación y estudio de la herramienta de diseño de placas PCB	7	8	08/05/2016	11/05/2016
8.2	Diseño del <i>layout</i>	8.1	8	12/05/2016	15/05/2016
9	PEC3			13/04/2016	17/05/2016
9.1	Redacción del borrador	5.3, 5.4, 7, 8	11	13/04/2016	14/05/2016
9.2	Entrega del borrador	9.1	1	15/05/2016	15/05/2016
9.3	Corrección del borrador	9.2	5	16/05/2016	16/05/2016
9.4	Entrega de la PEC3	9.3	1	17/05/2016	17/05/2016
10	Conclusiones y ampliaciones			18/05/2016	19/05/2016
10.1	Reflexión sobre los objetivos del proyecto	9	1	18/05/2016	18/05/2016
10.2	Problemas surgidos	10.1	1	18/05/2016	18/05/2016
10.3	Lecciones aprendidas	10.2	1	18/05/2016	18/05/2016
10.4	Soluciones aportadas	10.3	1	19/05/2016	19/05/2016
10.5	Posibles ampliaciones futuras	10.4	1	19/05/2016	19/05/2016
11	Memoria del TFC			18/05/2016	14/06/2016
11.1	Redacción del borrador de la memoria	10	28	18/05/2016	11/06/2016
11.2	Entrega del borrador	11.1	1	12/06/2016	12/06/2016
11.3	Corrección del borrador	11.2	7	13/06/2016	13/06/2016
11.4	Entrega de la memoria	11.3	1	14/06/2016	14/06/2016
12	Presentación del TFC			18/05/2016	14/06/2016
12.1	Elaboración del borrador de la presentación	10	28	18/05/2016	11/06/2016
12.2	Entrega del borrador	12.1	1	12/06/2016	12/06/2016
12.4	Corrección del borrador	12.2	7	13/06/2016	13/06/2016
12.5	Entrega de la presentación	12.3	1	14/06/2016	14/06/2016
13	Debate virtual	11, 12	6	29/06/2016	01/07/2016
Total			269		

Tabla 3: Tareas y actividades

1.4.4. Diagrama de Gantt

Representaremos toda la información del apartado 1.4.3 a lo largo de una línea de tiempo a través de un diagrama de Gantt.

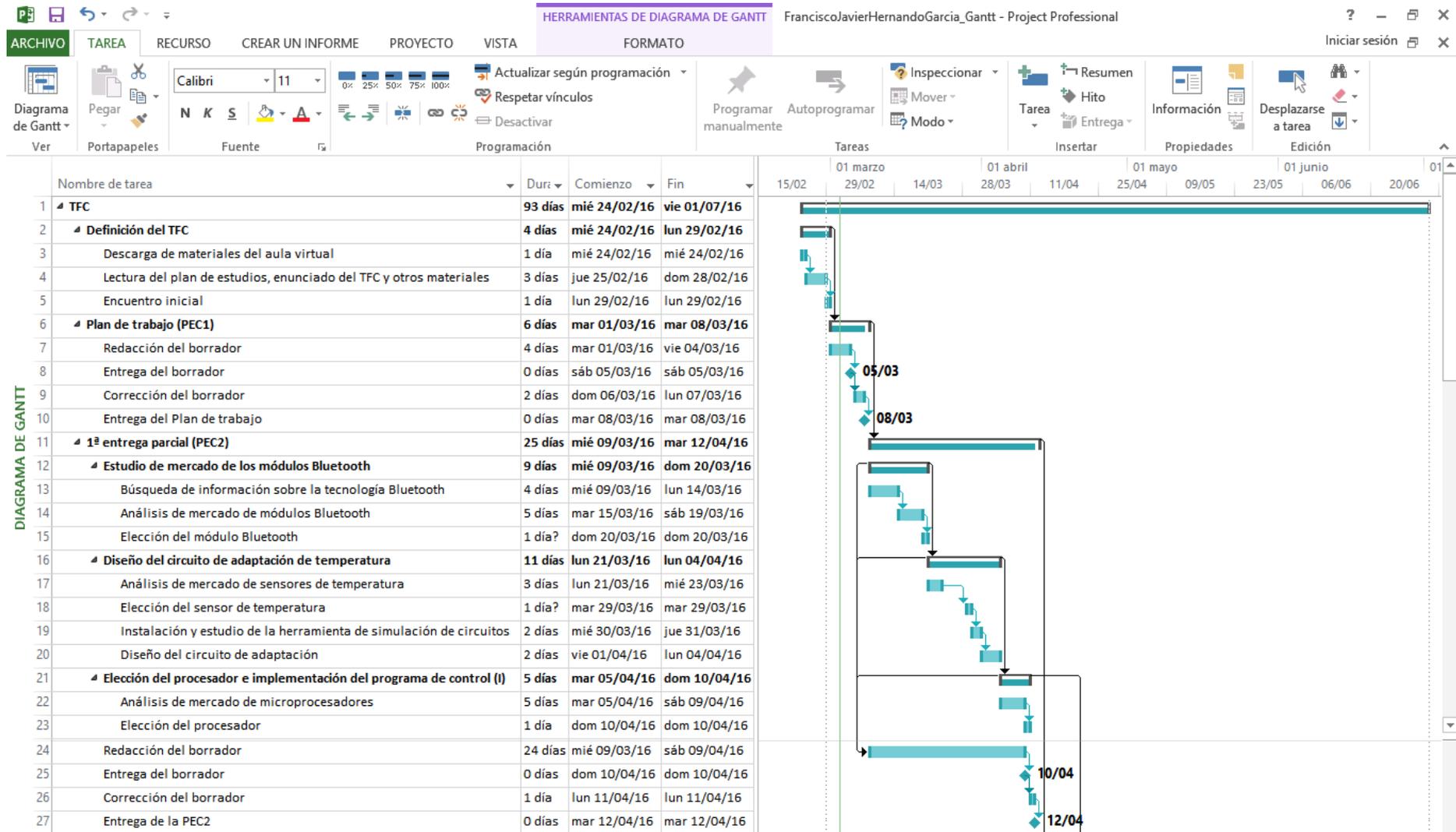


DIAGRAMA DE GANTT	28	2ª entrega parcial (PEC3)	25 días	mié 13/04/16	mar 17/05/16
	29	Diseño de la fuente de alimentación	7 días	mié 13/04/16	jue 21/04/16
	30	Estudio de la fuente de alimentación	3 días	mié 13/04/16	vie 15/04/16
	31	Análisis de consumos	2 días	sáb 16/04/16	lun 18/04/16
	32	Diseño de la fuente	3 días	mar 19/04/16	jue 21/04/16
	33	Elección del procesador y diseño del programa de control (y II)	12 días	vie 22/04/16	sáb 07/05/16
	34	Instalación y estudio de herramientas de programación de PIC	2 días	vie 22/04/16	dom 24/04/16
	35	Diseño y simulación del programa de control	11 días	lun 25/04/16	sáb 07/05/16
	36	Diseño del layout de la placa PCB	7 días	dom 08/05/16	dom 15/05/16
	37	Instalación y estudio de herramienta de diseño de placas PCB	4 días	dom 08/05/16	mié 11/05/16
DIAGRAMA DE GANTT	38	Diseño del layout	3 días	jue 12/05/16	dom 15/05/16
	39	Redacción del borrador	24 días	mié 13/04/16	sáb 14/05/16
	40	Entrega del borrador	0 días	dom 15/05/16	dom 15/05/16
	41	Corrección del borrador	1 día	lun 16/05/16	lun 16/05/16
	42	Entrega de la PEC3	0 días	mar 17/05/16	mar 17/05/16
	43	Entrega final	20 días	mié 18/05/16	mar 14/06/16
	44	Conclusiones y ampliaciones	2 días	mié 18/05/16	jue 19/05/16
	45	Reflexión sobre los objetivos del proyecto	1 día?	mié 18/05/16	mié 18/05/16
	46	Problemas surgidos	1 día?	mié 18/05/16	mié 18/05/16
	47	Lecciones aprendidas	1 día?	mié 18/05/16	mié 18/05/16
48	Soluciones aportadas	1 día?	jue 19/05/16	jue 19/05/16	
49	Posibles ampliaciones futuras	1 día?	jue 19/05/16	jue 19/05/16	
50	Memoria del TFC	20 días	mié 18/05/16	mar 14/06/16	
51	Redacción del borrador	19 días	mié 18/05/16	sáb 11/06/16	
52	Entrega del borrador	0 días	dom 12/06/16	dom 12/06/16	
53	Corrección del borrador	1 día?	lun 13/06/16	lun 13/06/16	
54	Entrega de la memoria	0 días	mar 14/06/16	mar 14/06/16	
55	Presentación del TFC	20 días	mié 18/05/16	mar 14/06/16	
56	Elaboración del borrador	19 días	mié 18/05/16	sáb 11/06/16	
57	Entrega del borrador	0 días	dom 12/06/16	dom 12/06/16	
58	Corrección del borrador	1 día?	lun 13/06/16	lun 13/06/16	
59	Entrega de la presentación	0 días	mar 14/06/16	mar 14/06/16	
60	Debate virtual	3 días	mié 29/06/16	vie 01/07/16	

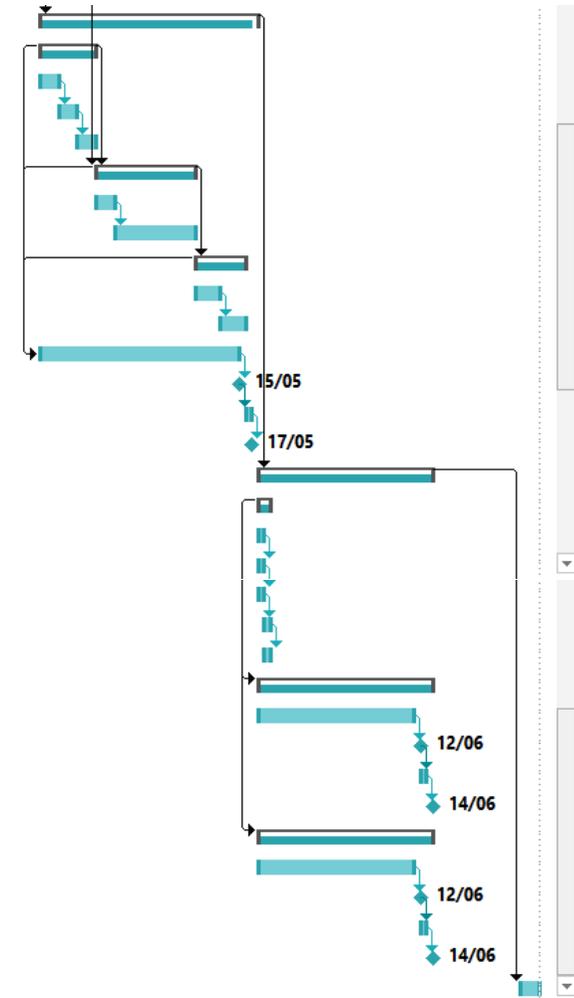


Ilustración 2: Diagrama de Gantt

1.5. Riesgos y plan de contingencia

Otra de las tareas a realizar durante la confección de nuestro Plan de trabajo, es la de identificar y analizar todas aquellas situaciones que puedan poner en riesgo nuestra planificación temporal y, por lo tanto, el cumplimiento de plazos y entregas.

1.5.1. Incidencias con el punto de trabajo

Nuestro punto de trabajo podría sufrir cualquier tipo avería que nos impediría su utilización hasta su reparación o reposición e incluso la pérdida parcial o total del trabajo realizado.

Plan de contingencia:

- Reposición inmediata del punto de trabajo.
- Realización diaria de copias de seguridad del trabajo realizado.

1.5.2. Dificultad del TFC

Algunos de los contenidos de este TFC son nuevos para el estudiante y otros se recibieron durante los primeros semestres de la carrera, por lo que la adquisición de esos nuevos conocimientos y el refresco de aquellos que están casi en el olvido pueden necesitar un tiempo extra.

Plan de contingencia:

- Aumentar las horas de dedicación mínima.
- Solicitar días libres o vacaciones en el trabajo.

1.5.3. Modificación de las condiciones laborales

Por necesidades de la producción puede ser necesario alargar las jornadas de trabajo o reducir las jornadas de descanso semanal e incluso desplazamientos al extranjero de varios días de duración.

Plan de contingencia:

- Aumentar las horas de dedicación mínima.
- Solicitar días libres por exceso de jornada o vacaciones en el trabajo.
- Tratar con el consultor la posibilidad de adelantar la fecha de entrega si vamos por delante de la planificación.

1.5.4. Indisposición médica

Una enfermedad o accidente de carácter grave o muy grave podría impedirnos durante unos días la dedicación a nuestro TFC.

Plan de contingencia:

- Aumentar las horas de dedicación mínima en cuanto sea posible.
- Solicitar en el trabajo días libres o vacaciones tras recibir el alta médica.

1.6. Productos obtenidos

Como resultado de la realización de este proyecto se van a obtener los siguientes entregables:

- Una memoria final de 60 páginas.
- Una presentación en vídeo de este TFC.
- Circuito de adaptación de señales de temperatura en TINA-TI.
- El código en lenguaje C del programa de control.
- Proyecto para la simulación del programa de control en Proteus.
- Esquemático del supervisor remoto de temperatura vía Bluetooth en Eagle.
- PCB del supervisor remoto de temperatura vía Bluetooth en Eagle.

1.7. Breve descripción de la memoria

En el siguiente capítulo trataremos el módulo Bluetooth. Empezaremos haciendo una pequeña introducción a esta tecnología y veremos qué características nos interesa que tenga el módulo Bluetooth de nuestro supervisor para poder elegir el más idóneo de todos los disponibles en el mercado.

En el capítulo 3 repasaremos los principales tipos de sensores de temperatura, elegiremos el más adecuado para nuestro supervisor y diseñaremos un circuito de adaptación para que la señal de temperatura que entregue el sensor alcance unos niveles de tensión adecuados a la entrada del convertidor analógico-digital de microcontrolador.

El capítulo 4 estará dedicado al microcontrolador. Haremos una breve introducción a estos componentes y analizaremos las características que deba reunir nuestro microcontrolador para poder seleccionar uno de ellos. Implementaremos el programa de control para el microcontrolador y realizaremos la simulación del mismo.

En el capítulo 5 diseñaremos la fuente de alimentación de nuestro supervisor en el que a partir de los 24 V que nos entrega una fuente externa obtendremos las tensiones que necesitan los componentes de nuestro diseño teniendo en cuenta el consumo de los mismos que habremos analizado previamente.

El capítulo 6 estará centrado en el diseño de los *layouts* de la PCB (*Printed Circuit Board* o placa de circuito impreso) a partir del esquemático de nuestro supervisor. También realizaremos una valoración económica o presupuesto de los que nos costaría la realización de nuestro prototipo.

El último capítulo queda reservado para las conclusiones en las que hablaremos del grado de cumplimiento de los objetivos propuestos, de los problemas encontrados, de las soluciones planteadas y de las posibles líneas de ampliación de nuestro proyecto.

2. El módulo Bluetooth

En este capítulo comenzaremos con una pequeña introducción teórica sobre la tecnología Bluetooth, comentaremos los criterios que tendremos en cuenta a la hora de elegir un módulo Bluetooth y cuáles son las principales características del módulo elegido.

2.1. Introducción a la tecnología Bluetooth^{vii viii ix x xi}

Bluetooth es una interfaz de radio que opera en la banda de los 2'4 GHz (2'402 a 2'480 GHz) dividida en 79 canales con un ancho de banda de 1 MHz, donde las comunicaciones cambian constantemente de frecuencia portadora. No es necesaria una trayectoria libre de obstáculos y está indicada para distancias cortas.

Los dispositivos pueden ser maestros o esclavos en función de si controlan al resto de dispositivos o si son controlados por alguno de ellos. Una *piconet* está formada por un maestro que asigna frecuencias y que controla el funcionamiento de la red, y el resto son esclavos del maestro.

Un maestro puede tener hasta 7 esclavos activos y otros 255 en estado de espera o *parked* (aparcado). Cada *piconet* sólo tiene un maestro y ese maestro, a su vez, puede ser esclavo de otra *piconet* formando una *scatternet*. El maestro dice a los esclavos de su *piconet* en que frecuencias emite y recibe, por ese motivo pueden coexistir hasta 10 *piconets* en un mismo recinto.

Nos encontramos con dos tipos de enlace:

- Enlaces síncronos de conexión orientada (SCO) que soportan conexiones simétricas, pensadas para voz, o asimétricas.
- Enlaces asíncronos de baja conexión (ACL) que soportan conexiones asimétricas, pensadas para datos, o simétricas.

Se definen tres clases de transmisores cuyo alcance varía en función de la potencia radiada:

Clase	Potencia máxima permitida	Alcance
I	100 mW (20 dBm)	100 m
II	2'5 mW (4 dBm)	15 – 20 m
III	1 mW (0 dBm)	10 m

Tabla 4: Clases de Bluetooth

Existen varias versiones de la tecnología Bluetooth que están diseñadas para la retro compatibilidad, lo que permite que el último estándar cubra todas las versiones anteriores. Además, los dispositivos también se pueden clasificar según su capacidad de canal:

Versión	Ancho de banda
1.2	1 Mbps
2.0 + EDR	3 Mbps
3.0 + HS	24 Mbps
4.0	32 Mbps

Tabla 5: Versiones de Bluetooth

Este estándar define varios perfiles de aplicación, denominados perfiles Bluetooth, en función de qué tipos de servicios puede ofrecer un dispositivo. A su vez, cada dispositivo puede admitir múltiples perfiles. Los principales perfiles son:

- Perfil de distribución de audio avanzado (A2DP).
- Perfil de control de audio y video (AVCTP).
- Perfil de control remoto de audio y vídeo (AVRCP).
- Perfil de distribución de audio y video (AVDTP).
- Perfil de distribución genérica de audio y video (GAVDP).
- Perfil básico de imagen (BIP).
- Perfil básico de impresión (BPP).
- Perfil de acceso RDSI común (CIP).
- Perfil de telefonía inalámbrica (CTP).
- Perfil de red de marcado (DUNP).
- Perfil de fax (FAX).
- Perfil de transferencia de archivos (FTP).
- Perfil de acceso genérico (GAP).
- Perfil genérico de intercambio de objetos (GOEP).
- Perfil de sustitución de cable de copia impresa (HCRP).
- Perfil manos libres (HFP).
- Perfil de dispositivo de interfaz humana (HID).
- Perfil de auricular (HSP).
- Perfil de intercomunicador (ICP).
- Perfil de acceso LAN (LAP).
- Perfil de objeto *push* (OPP).
- Perfil de redes de área personal (PAN).
- Perfil de acceso SIM (SAP).
- Perfil de aplicación de descubrimiento de servicio (SDAP).
- Perfil de sincronización (SP).
- Perfil de puerto de serie (SPP).

Ultra Wide Band (UWB)

Es una tecnología Bluetooth que permite velocidades desde 100 Mbps hasta 2 Gbps y su alcance está limitado a 10 metros. Utiliza un ancho de banda de 500 MHz en las bandas de 3 y 10 GHz, tiene un consumo muy reducido, una buena calidad de servicio y ha sido diseñado para distribuir contenidos multimedia en tiempo real.

ZigBee

Es un estándar que trabaja en la banda de 2'4 GHz, permite velocidades entre 20 Kbps y 250 Kbps y dispone de mecanismos de ahorro de potencia que reducen el consumo. Está orientado al control remoto por lo que es idóneo para aplicaciones domóticas debido a que los volúmenes de información que se transmiten son pequeños.

WiBree o Bluetooth Low Energy (BLE)

Trabaja en la banda de los 2'4 GHz y permite velocidades hasta 1 Mbps. Destaca su bajo consumo debido a la inactividad del bloque radio del dispositivo durante la mayor parte del tiempo, por lo que es adecuado para funcionar con pilas o con energía solar. Tiene un alcance de 200 metros sin obstáculos.

Campos de aplicación idóneos:

- Como control remoto de dispositivos del hogar.
- Sensores inalámbricos.
- Aplicaciones HID (*Human Interface Devices*).

Modos de trabajo:

- *Single mode*, de muy bajo consumo.
- *Dual mode*, para aplicaciones multimedia.

2.2. Análisis de mercado

A la hora de elegir entre las diversas opciones en módulos Bluetooth presentes en el mercado, acudimos a nuestro proveedor de componentes electrónicos (es.rs-online.com) y nos encontramos, entre otros, con los siguientes criterios de búsqueda:

- Clase de Bluetooth.
- Versión de Bluetooth.
- Interfaces admitidas.

Como hemos visto en el apartado 2.1, la clase nos va a determinar la potencia de transmisión y, por lo tanto, el alcance. Dado que nuestro supervisor va a recibir una alimentación de 24 voltios de una fuente externa, nos hace pensar que nuestro diseño no va destinado a ser un producto en movilidad sino un equipo que va a funcionar midiendo la temperatura de algún recinto cerrado.

Supondremos también que el equipo que reciba los datos de temperatura tampoco se encontrará muy lejos de nuestro supervisor y que una de las finalidades de nuestro proyecto es la eliminación de cableado. Por lo tanto, creemos que un módulo de clase 2 puede ser suficiente.

En cuanto a la versión, dado que nuestro supervisor no va a funcionar con pilas o baterías, sino con una fuente de alimentación externa, el consumo no nos preocupa y por ello consideramos que no es necesario un módulo de la versión 4.x correspondiente al *Bluetooth Low Energy* (BLE).

El tamaño de los datos a transmitir tampoco nos parece elevado, por lo que cualquiera de las versiones inferiores de Bluetooth nos parece, en un principio, válidas, ya que sus anchos de banda están comprendidos entre 1 y 3 Mbps.

En lo que se refiere a las interfaces y planeando que la comunicación entre el microcontrolador y el módulo Bluetooth va a ser serie, consideramos imprescindible que nuestro módulo tenga puerto serie o UART (*Universal Asynchronous Receiver-Transmitter* o Transmisor-Receptor Asíncrono Universal) y, por lo tanto, también es necesario que admita el perfil de puerto serie (SPP).

2.3. Nuestro módulo Bluetooth^{xii xiii}

Partiendo de los condicionantes que acabamos de comentar en el apartado 2.2 nos encontramos con el modelo RN42XVP-I/RM del fabricante Microchip:

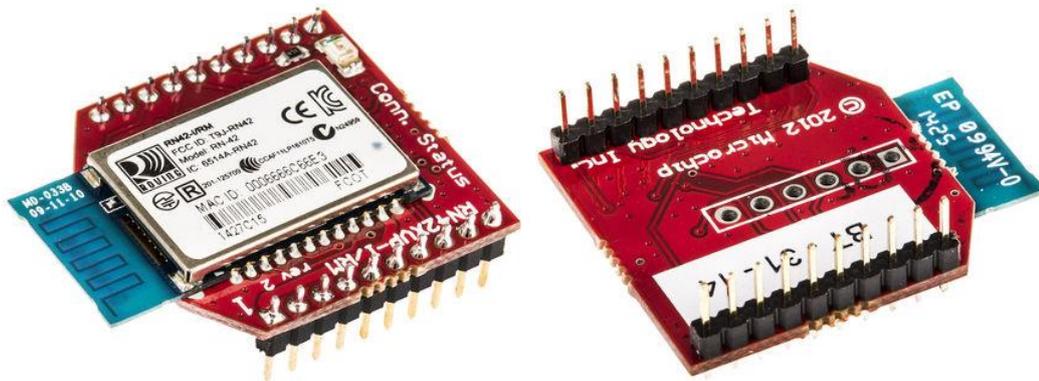


Ilustración 3: Módulo Bluetooth Microchip RN42XVP-I/RM

Es una solución de formato compacto y fácil de integrar para añadir comunicaciones Bluetooth a cualquier diseño. Está totalmente certificado, es compatible con varios protocolos de interfaz y presenta un bajo consumo de potencia.

Características:

- Versión 2.1 + velocidad de datos mejorada (EDR).
- Compatible con versiones anteriores de Bluetooth 2.0, 1.2 y 1.1.
- Consumo de potencia bajo: 26 μ A inactivo, 3 mA conectado, 30 mA transmisión.
- Interfaz serie UART con control de flujo RTS/CTS.
- Velocidad de transmisión de datos: 240 Kbps (modo esclavo o *slave*), 300 Kbps (modo maestro o *master*).
- Interfaces de conexión de datos UART (SPP o HCI) y USB (HCI).
- Soporta los perfiles Bluetooth SPP y HID, y los protocolos GAP, SDP, RFCOMM y L2CAP.
- Clase 2. Potencia de transmisión: + 4 dBm. Sensibilidad del receptor: - 80 dBm.
- Alcance hasta 20 metros.
- Rango de frecuencia RF: 2,402 GHz a 2,480 GHz con 79 canales de 1 MHz de ancho de banda.
- Alimentación entre + 3,0 V y + 3,6 V DC
- Compatible con las familias de microcontrolador de Microchip: PIC16F, PIC18F, PIC24F/H, dsPIC33 y PIC32.

3. El sensor de temperatura y circuito de adaptación

En este capítulo repasaremos los diferentes sensores de temperatura, elegiremos uno de ellos en función de unos criterios de búsqueda y diseñaremos un circuito que adapte la señal que proporciona nuestra sonda a la entrada del convertor analógico-digital.

3.1. Introducción a los sensores de temperatura^{xiv}

Son dispositivos que transforman la temperatura como variable física en una variable electrónica.

Tipos de sensores:

- Los bimetales, que se basan en el cambio de volumen de los materiales cuando varía la temperatura. Consisten en la unión de dos láminas de metales con coeficientes de dilatación con la temperatura muy distintos. Se utilizan como termostatos en aparatos de aire acondicionado y calefacción, por lo que serían poco indicados para nuestro proyecto.
- Las RTD (*resistor thermal detector* o resistencia detector térmico), que se basan en el aumento de la resistencia eléctrica de los metales al aumentar la temperatura. Pueden conectarse en un puente resistivo o puente de Wheatstone donde podremos obtener una tensión que variará con la temperatura. El platino, metal típico en estos sensores, ofrece un margen lineal en un amplio margen de medida.



Ilustración 5: Termostato bimetálico

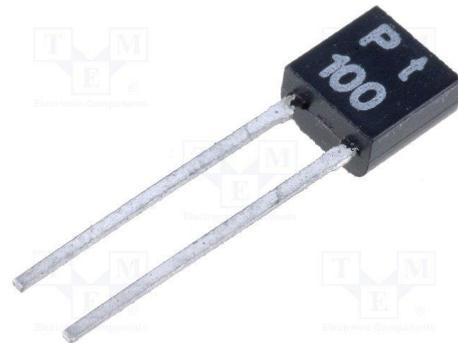


Ilustración 4: RTD PT100

- Las resistencias semiconductoras o termistores como, por ejemplo, las NTC (*negative thermal coefficient* o de coeficiente negativo de temperatura), cuya resistencia disminuye al aumentar la temperatura; y las PTC (*positive thermal coefficient* o de coeficiente positivo de temperatura), cuya resistencia aumenta al aumentar la temperatura. El inconveniente de estos sensores es que su linealidad depende de la curva característica corriente-tensión en la que estén trabajando.
- Las basadas en uniones de materiales semiconductores, cuyo comportamiento varía con la temperatura. Un ejemplo de este tipo de sensor de temperatura es el LM35 que ofrece a su salida una tensión de 10 mV/°C.



Ilustración 6: Termistores PTC y NTC

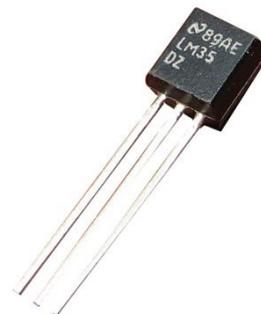


Ilustración 7: LM35

- Los termopares, que se basan en el hecho de que cuando la unión de dos metales diferentes se somete a una variación de temperatura, aparece en sus terminales una tensión termoeléctrica. Uno de sus inconvenientes es que necesitan una temperatura de referencia.
- Los cristales de cuarzo, que sufren oscilaciones al someterlos a temperaturas extremas.
- Los sensores por radiación electromagnética, que detectan la temperatura de los cuerpos a partir de las radiaciones electromagnéticas que emiten.



Ilustración 8: Sonda de temperatura basada en termopar

3.2. Análisis de mercado

Si tenemos en cuenta el diagrama de bloques de nuestro supervisor, podemos apreciar que el sensor de temperatura es un elemento externo, es decir, no se trata de un componente soldado a nuestra placa sino de un sensor tipo sonda conectado de alguna manera a la placa de nuestro supervisor. Así pues, descartamos sensores del tipo del LM35.

Volvemos a nuestro proveedor de componentes electrónicos y analizamos las diferentes sondas de temperatura que nos ofrecen y nos encontramos con que algunas utilizan tecnología NTC, otras PTC, los termopares y las RTD.

Hemos visto que trabajar con NTC y PTC es algo complicado porque hay que hacerlas trabajar en la curva característica corriente-tensión correcta para obtener de ellas una respuesta lineal y las basadas en termopar necesitan una temperatura de referencia.

Sin embargo, las sondas basadas en RTD, como las de platino, tienen una respuesta lineal en casi todo el rango de medida. El rango que debemos poder medir con nuestro supervisor está comprendido entre los 20 y los 120°C, pero deberemos tener en cuenta un rango ligeramente mayor a la hora de elegir nuestro sensor.

3.3. Nuestro sensor de temperatura^{xv xvi}

El sensor elegido está basado en un PT1000 y según nuestro proveedor sus principales características son:

- Sensor de película fina de platino PT1000 (clase B).
- Longitud de sonda: 30 mm
- Diámetro de sonda: 15 mm
- Mínima temperatura detectada: - 50 °C
- Máxima temperatura detectada: + 150 °C
- Tipo de terminación: extremo de dos cables.
- Cable de 1 metro trenzado doble con aislamiento de Teflón de 7 / 0,2 mm



Ilustración 9: Sensor de temperatura

3.4. Diseño del circuito de adaptación de señales^{xvii xviii xix}

A la hora de elegir nuestro módulo Bluetooth, vimos que prácticamente la totalidad de ellos trabajan con una tensión de alimentación comprendida entre los 3 y los 3,6 V. A la hora de comunicar el módulo Bluetooth con nuestro microcontrolador a través del puerto serie y con la finalidad de no complicar en exceso nuestro diseño, hemos visto necesario que los niveles lógicos de ambos puertos tengan el mismo nivel de tensión.

Esto implica que:

- La tensión de alimentación del microcontrolador deberá ser la misma que la del módulo Bluetooth, que está comprendida entre 3 y 3,6 V por lo que estableceremos la tensión de alimentación en 3,3 V.
- La señal analógica que genere nuestro circuito de adaptación de señales como paso intermedio entre el sensor de temperatura y el conversor analógico-digital deberá entregar una tensión mínima de 0 V, correspondiente a la detección de la temperatura mínima de 20 °C, y una tensión máxima de 3,3 V, correspondiente a la detección de la temperatura máxima de 120 °C.
- Para que nuestro sensor de temperatura entregue una tensión de 0 V al detectar la temperatura mínima a considerar, deberemos montarla en un puente de Wheatstone, por lo que descartaremos la opción de montar un divisor de tensión donde no tendríamos ninguna opción de obtener una tensión nula.

En cuanto al rango de detección de temperatura y con el fin de asegurarnos por completo de que el conversor analógico-digital (A/D) va a muestrear y codificar las temperaturas comprendidas entre los 20 y los 120 °C, creemos oportuno ampliar ese rango en ± 10 °C, de modo que el rango a considerar ahora estará comprendido entre los 10 y los 130 °C.

Como hemos visto en la introducción a los sensores de temperatura, el valor de la resistencia eléctrica de las RTD varía en función de la temperatura y viene dada por la siguiente expresión:

$$R(T) = R_0(1 + \alpha(T - T_0))$$

Donde:

- $R(T)$ es el valor de la resistencia eléctrica de nuestro sensor a la temperatura a considerar.
- R_0 es la resistencia eléctrica de nuestro sensor a la temperatura de referencia, en el caso de nuestra PT1000 es de 1000 Ω a la temperatura de 0 °C.
- α es el coeficiente de temperatura del metal que indica la variación de su resistencia eléctrica al variar la temperatura, en nuestro caso, el coeficiente del platino es de 0,00385.

Calculamos el valor de la resistencia eléctrica a la temperatura mínima a considerar:

$$R(10) = 1000(1 + 0,00385(10 - 0)) = 1038,5 \Omega$$

Y calculamos el valor de la resistencia eléctrica a la temperatura máxima a considerar:

$$R(130) = 1000(1 + 0,00385(130 - 0)) = 1500,5 \Omega$$

Montamos un puente de Wheatstone ajustando el valor del resto de resistencias al valor de la resistencia de la PT1000 a la temperatura de 10 °C para que la tensión de salida sea de 0 V. Para el valor teórico de 1038,5 Ω tomamos los siguientes valores reales de resistencia en el mercado: 1,02 kΩ y 18,2 Ω y realizamos la simulación:

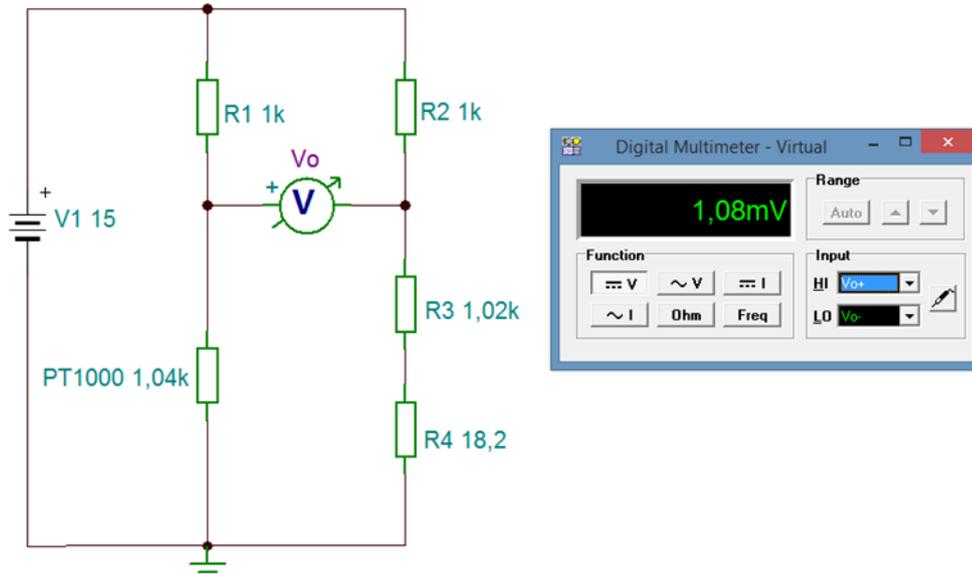


Ilustración 10: Simulación del puente de Wheatstone a la temperatura mínima

Vemos que para la temperatura mínima de 10 °C, la salida del puente de Wheatstone es de 1,08 mV que consideraremos nula para que sea detectada como nivel lógico bajo por el conversor A/D.

Hagamos ahora la simulación para el valor máximo de temperatura de 130 °C, en el que el valor de resistencia de nuestra PT1000 es de 1500,5 Ω:

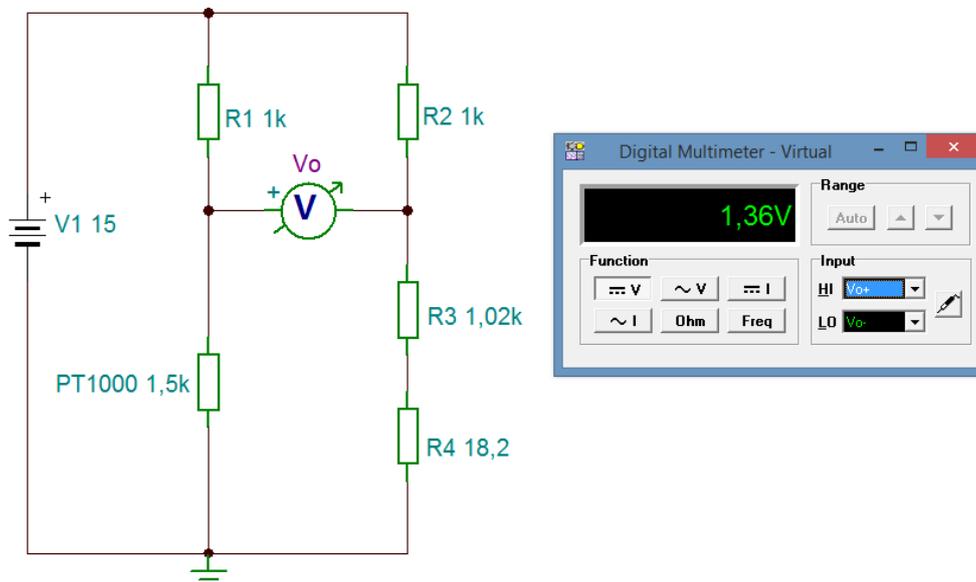


Ilustración 11: Simulación del puente de Wheatstone a la temperatura máxima

Y vemos que para la temperatura máxima de 130 °C, la salida del puente de Wheatstone es de 1,36 V, tensión que deberemos amplificar para que alcance los 3,3 V a la entrada del conversor analógico-digital y sea detectada como nivel lógico alto.

Para amplificar esta tensión utilizaremos un amplificador operacional como el $\mu A741$, muy utilizado en montajes electrónicos básicos, configurado en modo diferencial como el de la siguiente ilustración:

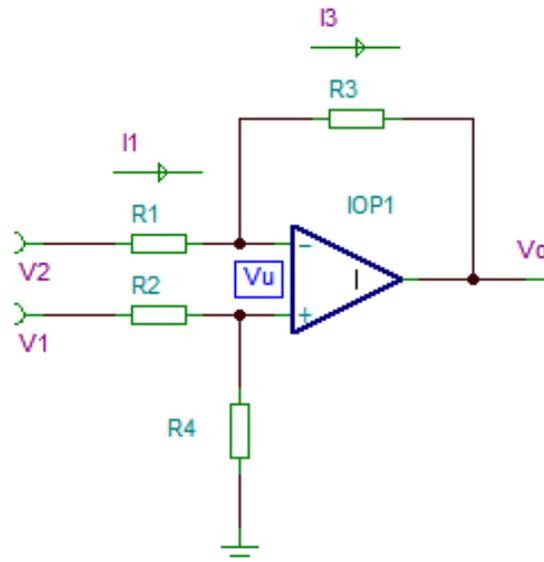


Ilustración 12: Amplificador operacional en configuración de ganancia diferencial

Necesitamos amplificar la tensión de 1,36 V para que alcance los 3,3 V correspondientes al nivel lógico alto que deseamos a la entrada del convertor A/D, por lo que la ganancia de nuestro amplificador deberá ser de:

$$\Delta_V = \frac{V_o}{V_i} = \frac{3,3}{1,36} = 2,4265$$

La ganancia dependerá de los valores de R1, R2, R3 y R4, así que comenzamos los cálculos partiendo de que las corrientes que atraviesan R1 y R3 son iguales:

$$I_1 = I_3$$

Estos valores de corriente son iguales a:

$$\frac{V_2 - V_u}{R_1} = \frac{V_u - V_o}{R_3}$$

$$\frac{V_2}{R_1} - \frac{V_u}{R_1} = \frac{V_u}{R_3} - \frac{V_o}{R_3}$$

Sacamos factor común V_u :

$$\frac{V_2}{R_1} - \frac{V_u}{R_1} - \frac{V_u}{R_3} = -\frac{V_o}{R_3}$$

$$\frac{V_2}{R_1} - V_u \left(\frac{1}{R_1} + \frac{1}{R_3} \right) = -\frac{V_o}{R_3}$$

$$\frac{V_2}{R_1} - V_u \left(\frac{R_1 + R_3}{R_1 \cdot R_3} \right) = -\frac{V_o}{R_3}$$

Calculamos el valor de V_u en la entrada no inversora:

$$V_u = \frac{V_1}{R_2 + R_4} R_4$$

Para simplificar las dos últimas expresiones igualaremos los valores de R1 y R2, y de R3 y R4 de modo que:

$$V_u = \frac{V_1}{R_1 + R_3} R_3$$

Y sustituimos en la expresión anterior:

$$\frac{V_2}{R_1} - \left(\frac{V_1}{R_1 + R_3} R_3 \right) \cdot \left(\frac{R_1 + R_3}{R_1 \cdot R_3} \right) = -\frac{V_o}{R_3}$$

Simplificamos eliminando R1+R3:

$$\frac{V_2}{R_1} - \frac{V_1 \cdot R_3}{R_1 \cdot R_3} = -\frac{V_o}{R_3}$$

Desarrollamos la resta:

$$\frac{V_2 \cdot R_3}{R_1 \cdot R_3} - \frac{V_1 \cdot R_3}{R_1 \cdot R_3} = -\frac{V_o}{R_3}$$

$$\frac{V_2 \cdot R_3 - V_1 \cdot R_3}{R_1 \cdot R_3} = -\frac{V_o}{R_3}$$

Volvemos a simplificar eliminando R3:

$$\frac{(V_2 - V_1) \cdot R_3}{R_1} = -V_o$$

$$V_o = -(V_2 - V_1) \frac{R_3}{R_1}$$

Cambiamos el orden de la resta:

$$V_o = (V_1 - V_2) \frac{R_3}{R_1}$$

Como Vi es la diferencia entre V1 y V2:

$$V_o = V_i \frac{R_3}{R_1}$$

$$\frac{V_o}{V_i} = \frac{R_3}{R_1}$$

La ganancia en tensión viene dada por la siguiente expresión:

$$\Delta_V = \frac{R_3}{R_1}$$

Y la relación entre R3 y R1 vienen dada por:

$$R_3 = \Delta_V \cdot R_1$$

Entonces, si tomamos para R1 y R2 un valor de 1 kΩ, los valores de R3 y R4 serán de:

$$R_3 = R_4 = \Delta_V \cdot R_1 = 2,4265 \cdot 1000 = 2462,5 \Omega$$

Si tomamos valores reales de resistencias en el mercado, las elegiríamos de 2,43 k Ω . Realizamos la simulación para nuestro valor máximo de temperatura, 130 °C, donde a la salida del puente de Wheatstone teníamos 1,36 V y a la salida del circuito de adaptación queremos tener 3,3 V:

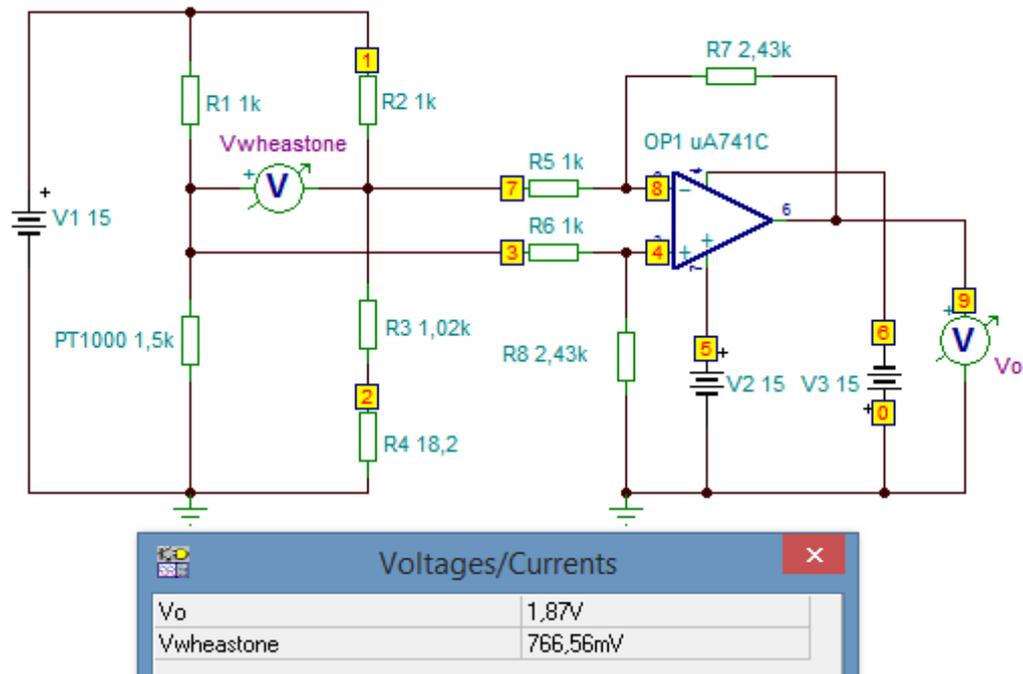


Ilustración 13: Simulación de nuestro primer circuito de adaptación de señal de temperatura

Y vemos que los resultados no son los esperados: a la salida del puente de Wheatstone tenemos 766,56 mV, en lugar de los 1,36 V, y a la salida del circuito de adaptación tenemos 1,87 V, en lugar de los 3,3 V.

Estas enormes diferencias se deben a una desadaptación de impedancias entre el puente de Wheatstone y el circuito de adaptación que eliminaremos interponiendo entre los dos circuitos otros dos amplificadores operacionales pero en modo seguidor de tensión como el de la figura siguiente:

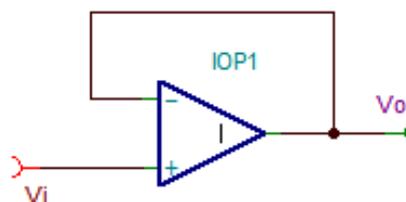


Ilustración 14: Amplificador operacional en modo seguidor de tensión

De este modo evitaremos que las resistencias R5, R6, R7 y R8 interfieran en el puente de Wheatstone y, al mismo tiempo, que las resistencias R1, R2, R3 y R4 no interfieran en el valor de ganancia de tensión del amplificador operacional.

A continuación repetiremos de nuevo la simulación con los dos operacionales seguidores de tensión, donde obtendremos los resultados previstos en la medición del valor máximo de temperatura, 130 °C:

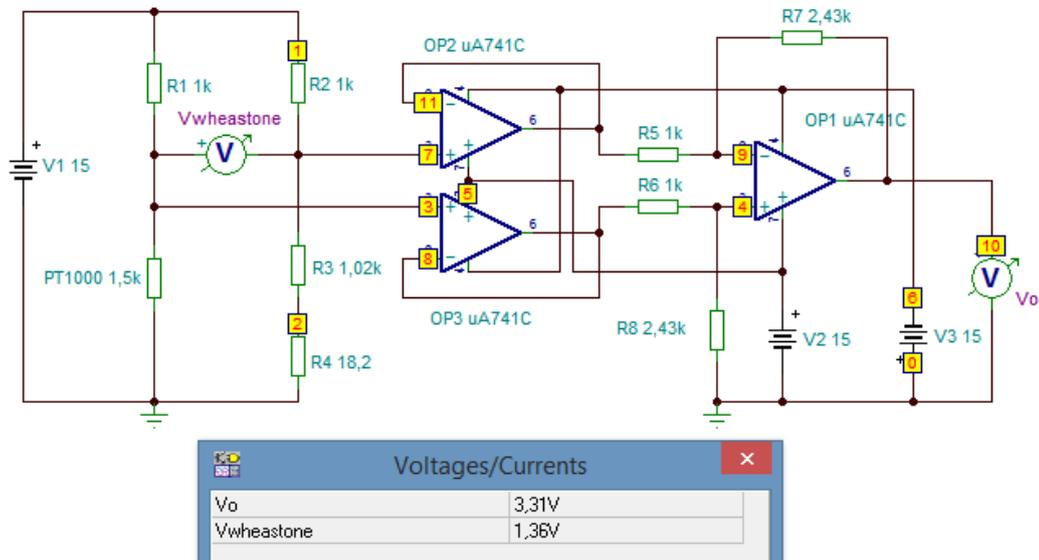


Ilustración 15: Simulación del circuito de adaptación de señales a la temperatura máxima

Y repetimos la simulación para el valor mínimo de temperatura que queremos medir, esto es, a 10 °C:

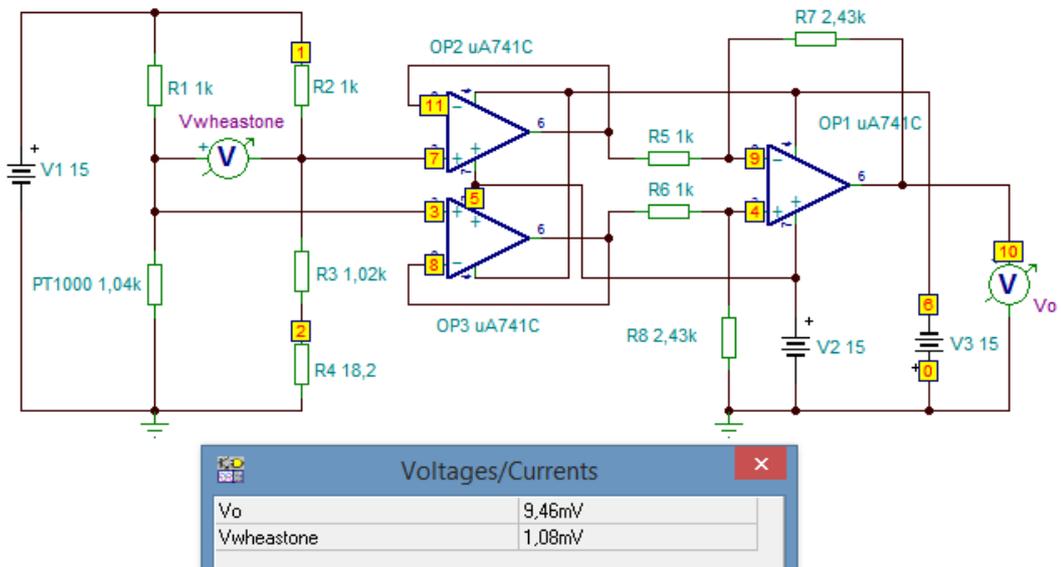


Ilustración 16: Simulación del circuito de adaptación de señales a la temperatura mínima

4. El microcontrolador

En este capítulo comenzamos con una muy pequeña introducción a los microcontroladores, continuaremos con la elección de uno de ellos según unos criterios de búsqueda que comentaremos más adelante, y finalizaremos con la implementación y simulación del programa de control.

4.1. Introducción a los microcontroladores^{xx}

Un microcontrolador es un dispositivo diseñado para la gestión del funcionamiento de un determinado proceso para el que ha sido adaptado. Sus elementos principales son:

- Procesador, que se encarga del proceso de lectura y ejecución de instrucciones.
- Memoria, que almacena las instrucciones del proceso y los datos que manipula el mismo.
- Puertos de entrada y salida, que permiten la comunicación con otros elementos del sistema, como por ejemplo:
 - Conversores analógico-digital (A/D) y/o digital-analógico (D/A).
 - Bus de comunicaciones, que pueden seguir los protocolos UART, I2C, USB...
- Reloj, que establece la frecuencia de trabajo y permite la sincronización con otros elementos del sistema.
- *Timers*, que controlan intervalos de tiempo.
- *Watchdogs*, que reinician el proceso si el sistema se ha quedado bloqueado.

Nos podemos encontrar con dos tipos de arquitecturas en función de la manera de implementar la memoria y los buses de comunicación:

- Von Neumann, cuando las instrucciones y los datos se encuentran almacenados en la misma memoria y los buses para acceder a ambos son los mismos.
- Harvard, cuando las instrucciones y los datos se encuentran almacenados en memorias distintas, los buses están segregados para cada tipo de memoria y el tamaño de cada bus se adapta al tamaño de los datos que debe procesar.

En función de los repertorios de instrucciones, los procesadores pueden ser de dos tipos:

- RISC (*Reduced Instruction Set Computing*), que tienen un número muy pequeño de instrucciones simples y de tamaño fijo que se ejecutan en un ciclo de reloj.
- CISC (*Complex Instruction Set Computing*), que tienen un mayor número de instrucciones que, además, permiten realizar operaciones más complejas y que, por lo tanto, requieren más de un ciclo de reloj.

4.2. Estudio de mercado

Ha llegado el momento de elegir un microcontrolador y acudimos de nuevo a nuestro proveedor de componentes electrónicos donde nos encontramos con un gran número de criterios de búsqueda que nos ayudarán en la selección del dispositivo más apropiado.

Comenzaremos restringiendo la búsqueda a dispositivos de la marca Microchip por tres motivos: podemos utilizar su software de programación, además ofrecen una amplia gama de microcontroladores entre los que fácilmente encontraremos uno que se adapte a nuestras necesidades y, por si fuera poco, nuestro módulo Bluetooth ofrecía compatibilidad con algunas de sus familias de microcontroladores.

Otros de los condicionantes será que nuestro microcontrolador incluya al menos un conversor analógico-digital para digitalizar la señal analógica que ofrece el sensor de temperatura a través del circuito de adaptación.

Otro dato a tener en cuenta es la precisión del conversor A/D. Como hemos comentado en el apartado anterior, vamos a medir temperaturas comprendidas entre 10 y 130 °C con una precisión mínima requerida de 1 °C. El conversor A/D tendrá que tener la capacidad de codificar 120 muestras o valores de tensión analógica distintos, por lo que con una precisión de 8 bits debería ser más que suficiente ya que $2^8=256$ niveles.

Esta información de temperatura será transmitida mediante un módulo Bluetooth que comunicaremos con el microcontrolador a través de un puerto serie, por lo que otro de nuestros criterios de búsqueda será la inclusión de al menos un puerto UART.

Sólo con estos criterios conseguimos reducir a 59 el número de modelos disponibles. Para restringir un poco más la búsqueda elegimos a aquellos que pueden montarse sobre orificio pasante (como nuestro módulo Bluetooth) en lugar de los de montaje superficial, de modo que ahora podemos elegir entre 8 modelos.

Según las características de nuestro módulo Bluetooth, éste ofrece compatibilidad con la familia PIC16F, de modo que ahora ya podemos elegir entre 3 modelos, de los que elegiremos el que tenga mayor tamaño de memoria y menor tamaño físico.

4.3. Nuestro microcontrolador^{xxi xxii xxiii}

El modelo elegido es el PIC16F723-I/SP, de la familia PIC16F722/723/724/726/727 de microcontroladores, que se basa en el núcleo de arquitectura media de Microchip con una pila de hardware de 8 niveles de profundidad y 35 instrucciones. Estos microcontroladores ofrecen hasta 5 MIPS (millones de instrucciones por segundo), hasta 14 Kbytes de memoria de programa y hasta 368 bytes de RAM. Incorpora un oscilador con una precisión de $\pm 1\%$.

Sus principales características son:

- Encapsulado SPDIP, 28 pines, montaje en orificio pasante.
- Velocidad de CPU máxima de 20 MHz
- Arquitectura RISC.
- 35 instrucciones.
- Pila de hardware de 8 niveles.
- Ancho del bus de datos: 8 bits.
- Tamaño de la memoria RAM: 192 Bytes.
- Memoria de programa: tipo Flash de 7 kB.
- Tensión de alimentación de funcionamiento típica: 2'5, 3'3 o 5 V.
- Tecnología nanoWatt XLP.
- Reset de encendido (POR).
- Temporizador de arranque (PWRT).
- Temporizador de arranque de oscilador (OST).
- Reset de caída de tensión (BOR).
- Temporizador Watchdog (WDT).

- Programación serie en circuito (ICSP).
- Depuración en circuito (ICD).

Periféricos:

- Un conversor analógico-digital (ADC) de 11 canales con una resolución de 8 bits.
- Módulo de oscilador de detección táctil capacitivo mTouch de 8 canales.
- Dos módulos de captura, comparación y PWM (CCP) con una resolución de 10 bits.
- Dos temporizadores de 8 bits y un temporizador de 16 bits.
- Un transmisor-receptor síncrono-asíncrono universal direccionable (AUSART).
- Un puerto serie síncrono (MSSP) con SPI e I2C.



Ilustración 17: Microchip PIC16F723

4.4. Diseño del programa de control^{xxiv xxv}

Para el diseño del programa de control utilizaremos la aplicación PIC C Compiler, debido a la gran cantidad de información y ejemplos que ofrece una multitud de usuarios en Internet.

Nuestro programa deberá configurar una serie de elementos del microcontrolador como: el oscilador interno, las entradas y salidas, el conversor analógico-digital y el puerto serie. Al mismo tiempo deberá ocuparse de configurar el módulo Bluetooth cada vez que arranque nuestro supervisor y de controlar su estado cuando esté funcionando.

Será necesario consultar la documentación del módulo Bluetooth para conocer cuáles son los comandos de configuración y cómo responde el módulo a cada comando que se le envía, con la finalidad de que se lleve a cabo correctamente la configuración inicial y la comprobación de su funcionamiento.

Será muy importante conocer cuáles son los parámetros que trae configurados por defecto, comprobar si algunos de ellos cubren nuestras necesidades o si, por el contrario, es necesario configurar otros parámetros a nuestra conveniencia.

Y en cuanto al tratamiento de la información de la temperatura, deberá indicar a través de qué entrada analógica se está introduciendo esa señal, cuáles son los niveles de la tensión de referencia y cuál es la resolución del conversor para que la conversión sea la correcta.

Para realizar la simulación utilizaremos la aplicación Proteus en cuya librería no aparece nuestro PIC16F723 pero sí el PIC16F723A, por lo que a efectos de programación y simulación utilizaremos este segundo modelo que tan sólo se diferencia del nuestro en el tamaño de la memoria de programa (ofrece 4096 palabras frente a los 7 kB de nuestro PIC16F723).


```

66 //          conectado y 0,0,0 si no está conectado) //
67 // //
68 //      R,1      Fuerza un reboot completo del módulo (tras un booting el //
69 //          módulo no puede entrar en modo command hasta pasados 500 ms) //
70 // //
71 // Sensor de temperatura PT1000 //
72 //      10°C -> 1038,5ohm (5,73mV)          130°C -> 1500,5ohm (3,3V) //
73 // //
74 // Para realizar la simulación deberán omitirse aquellas líneas del código con //
75 // doble asterisco (**), de lo contrario el programa entrará en bucle esperando //
76 // la respuesta del módulo Bluetooth y no avanzará //
77 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
78
79 #include <16f723A.h>
80
81 //Bits de configuración
82 #fuses HS,INTRC_IO,NOWDT,PUT,NOMCLR,NOPROTECT,NOBROWNOUT,NOPLLEN,NODEBUG,NOVCAP
83 //HS -> Frecuencia del oscilador > 4 MHz
84 //INTRC_IO -> Se utilizará el oscilador interno
85 //NOWDT -> No se utilizará Watch Dog
86 //NOMCLR -> Master Clean deshabilitado
87 //NOPROTECT -> Código no protegido contra lectura
88 //NOBROWNOUT -> No resetea la MCU aunque haya picos de tensión
89 //NOPLLEN -> Divisor de frecuencia deshabilitado por hardware
90 //NODEBUG -> Modo debug deshabilitado
91 //NOVCAP -> Pin VCAP deshabilitado
92
93 #use delay(clock=16M)
94
95 //Definición del puerto serie
96 //115200 baudios, 8 bits, sin paridad, TX por Pin_C6, RX por Pin_C7
97 #use RS232(FORCE_SW, XMIT=PIN_C6, RCV=PIN_C7, BAUD=115200, BITS=8, PARITY=N)
98
99 #define MAX_RESPUESTA 16 //Limita el tamaño máximo del buffer de entrada
100
101 #define adc=8; //Resolución del convertor A/D = 8 bits
102
103 //Función que devuelve el primer carácter de la respuesta del módulo Bluetooth
104 //tras enviarle un comando
105 char respuesta(){
106     char dato[MAX_RESPUESTA]; //Crea un array del tipo char
107     int8 c; //Crea un índice para el array
108     if(kbhit()){ //kbhit() devuelve TRUE si hay un dato para ser
109         //leído en el buffer de entrada
110         c=0; //Reinicia el índice del array a 0
111         while(kbhit()){ //Bucle mientras termina la transmisión
112             dato[c] = getc(); //Leemos el dato introduciéndolo en un array
113             c++; //Incrementa el índice del array
114             if(c>=(MAX_RESPUESTA-1)){
115                 break; //Si se llena el buffer de entrada se finaliza la
116             } //lectura para evitar perder datos recibidos
117         }
118     }
119     return(dato[0]); //Devuelve el carácter en la primera posición del
120 //array
121 }
122
123 //Función que configura el módulo Bluetooth, mantiene los valores por defecto y
124 //cambia sólo aquellos que nos interesan
125 void configBT(){
126     delay_ms(1000); //Retardo antes de entrar en modo command
127     while(respuesta()!='C'){ //**Hasta que el modulo no devuelva CMD
128         printf("$$$"); //Entra en modo command
129         delay_ms(1000); //Retardo antes de enviar comandos
130     } //**

```

```

131 while(respuesta()!='A'){ /**Hasta que el módulo no devuelva AOK
132     printf("SF,1"); //Recupera los valores por defecto
133     delay_ms(100); //Retardo antes de enviar comandos
134 } /**
135 while(respuesta()!='A'){ /**Hasta que el módulo no devuelva AOK
136     printf("SA,0"); //Establece modo abierto de autenticación
137     delay_ms(100); //Retardo antes de enviar comandos
138 } /**
139 while(respuesta()!='A'){ /**Hasta que el módulo no devuelva AOK
140     printf("SM,0"); //Establece modo esclavo
141     delay_ms(100); //Retardo antes de enviar comandos
142 } /**
143 while(respuesta()!='A'){ /**Hasta que el módulo no devuelva AOK
144     printf("SN,Supervisor de temperatura"); //Nombre asignado como
145                                             //"Supervisor de temperatura"
146     delay_ms(100); //Retardo antes de enviar comandos
147 } /**
148 while(respuesta()!='A'){ /**Hasta que el módulo no devuelva AOK
149     printf("SQ,16"); //Modo optimizado para transferencia de datos
150     delay_ms(100); //Retardo antes de enviar comandos
151 } /**
152 while(respuesta()!='E'){ /**Hasta que el módulo no devuelva END
153     printf("---"); //Salir del modo command
154     delay_ms(100); //Retardo antes de enviar comandos
155 } /**
156 }
157
158 //Función que comprueba si no hay conexión en el módulo Bluetooth para reiniciarlo
159 void checkBT(){
160     delay_ms(1000); //Retardo antes de entrar en modo command
161     while(respuesta()!='C'){ /**Mientras el módulo no responda CMD
162         printf("$$$"); //Entra en modo command
163         delay_ms(1000); //Retardo antes de enviar comandos
164     } /**
165     while(respuesta()!='0' && respuesta()!='1'){ /**Hasta que el módulo no
166                                             //devuelva 0,0,0 o 1,0,0
167         printf("GK"); //Pregunta al módulo su estado de conexión
168         delay_ms(100); //Retardo antes de enviar comandos
169         if(respuesta()=='0'){ //Si no hay conexión
170             printf("R,1"); //Envía comando reboot
171             delay_ms(1000); //Espera un segundo
172             configBT(); //Configura el módulo Bluetooth
173         }
174         if(respuesta()=='1'){ //Si hay conexión
175             while(respuesta()!='E'){ /**Mientras el módulo no responda END
176                 printf("---"); //Finaliza el modo command
177                 delay_ms(100); //Retardo antes de enviar comandos
178             } /**
179         }
180     } /**
181 }
182
183 //Función principal
184 void main(){
185     int8 valorADC; //Almacena el valor del canal analógico
186     float voltaje, temp; //Almacena valores de tensión y temperatura
187
188     setup_oscillator(OSC_16MHZ); //Configura la frecuencia del oscilador
189                                 //interno a 16 MHz
190     set_tris_C(0B10000000); //Configuración de puertos:
191                             //PIN_C7 como entrada RX,
192                             //PIN_C6 como salida TX
193     setup_adc(ADC_CLOCK_INTERNAL); //Conversor AD funciona con reloj interno
194
195     setup_adc_ports(sAN0 | VSS_VDD); //Configura AN0 como entrada analógica y
196                                     //voltaje de referencia entre VDD y GND

```

```

197     set_adc_channel(0);           //Se tomará lectura del canal analógico 0
198
199     configBT();                 //Configura el módulo Bluetooth
200
201     while(TRUE){               //Bucle para envío constante de datos
202         valorADC = read_adc();   //Leer valor analógico
203         voltaje = ((valorADC*3.3)/255.0); //Convertir en voltaje
204         temp = ((voltaje*120.0)/3.3)+10.0; //Convertir en temperatura
205         printf("\n Temperatura = %f grados centigrados",temp); //Envía temperatura
206         checkBT();             //Comprueba estado de conexión del módulo BT
207         delay_ms(60000);       //Envía la temperatura cada 1 minuto
208     }
209 }
    
```

Ilustración 18: Programa de control

A parte de los comentarios añadidos en el código del programa de control que hemos adjuntado, cabe justificar aquí el porqué de algunos comandos de configuración del módulo Bluetooth. Hemos elegido el modo de autenticación abierto para que el dispositivo que se conecte al módulo no tenga que introducir ningún PIN (línea 136 del código). Y configuramos el módulo en modo esclavo o *slave* para que siempre sea visible y pueda establecer una conexión sin que tengamos que enviarle una orden explícita para ello (línea 140 del código).

El valor numérico de la variable “voltaje” (línea 203 del código) se calcula a partir de la variable “valorADC”, que es el valor decimal que ofrece el conversor A/D a su salida, al que se le multiplica el nivel de tensión del voltaje de referencia, 3,3 V, y se divide entre 255 que son los niveles de cuantificación, niveles que vienen de la expresión $2^8 - 1$, siendo 8 el número de bits de resolución del conversor.

El valor numérico de la variable “temp” (línea 204 del código) se calcula a partir de la variable “voltaje”, cuyo valor se multiplica por 120, que es nuestro rango de medida de temperatura (de 10 a 130 °C), se divide entre el valor de la tensión del voltaje de referencia, 3,3 V, y se le suma 10 que es donde empieza nuestro rango de medida.

El envío de la información de temperatura se realiza cada minuto, pero podría modificarse fácilmente cambiando el valor del *delay* de la línea 207 del código.

4.5. Simulación del programa de control^{xxvi}

Dibujamos en Proteus nuestro puente de Wheatstone y el circuito de adaptación de señal de temperatura al que vamos a añadir el microcontrolador. Como hemos comentado en el apartado 4.4, nuestro PIC16F723 no está incluido en la librería de componentes de Proteus por lo que para la simulación hemos elegido el PIC16F723A.

Nuestro módulo Bluetooth tampoco está disponible en la librería, por lo que utilizaremos la herramienta “Virtual Terminal” para, al menos, ver los comandos que el microcontrolador enviará al módulo Bluetooth a través del puerto serie, ya que tampoco nos resulta posible emular su comportamiento de ninguna manera.

Por ese motivo, para realizar la simulación, es necesario omitir algunas de las líneas del código que al ejecutarse esperan la respuesta del módulo Bluetooth para poder continuar o para tomar una u otra decisión, ya que de lo contrario la ejecución del código entraría en un bucle sin salida.

Dentro de las propiedades del símbolo que representa nuestro PIC hay una serie de parámetros que es imprescindible indicar, como la frecuencia del oscilador, que deberá coincidir con la frecuencia establecida en el programa de control, y la ruta donde se aloja el programa de control.

Del mismo modo, en las propiedades del "Virtual Terminal" indicaremos los mismos datos de velocidad, número de bits de datos y parada, y paridad que la del puerto serie del microcontrolador que aparecen en las líneas de configuración del programa de control.

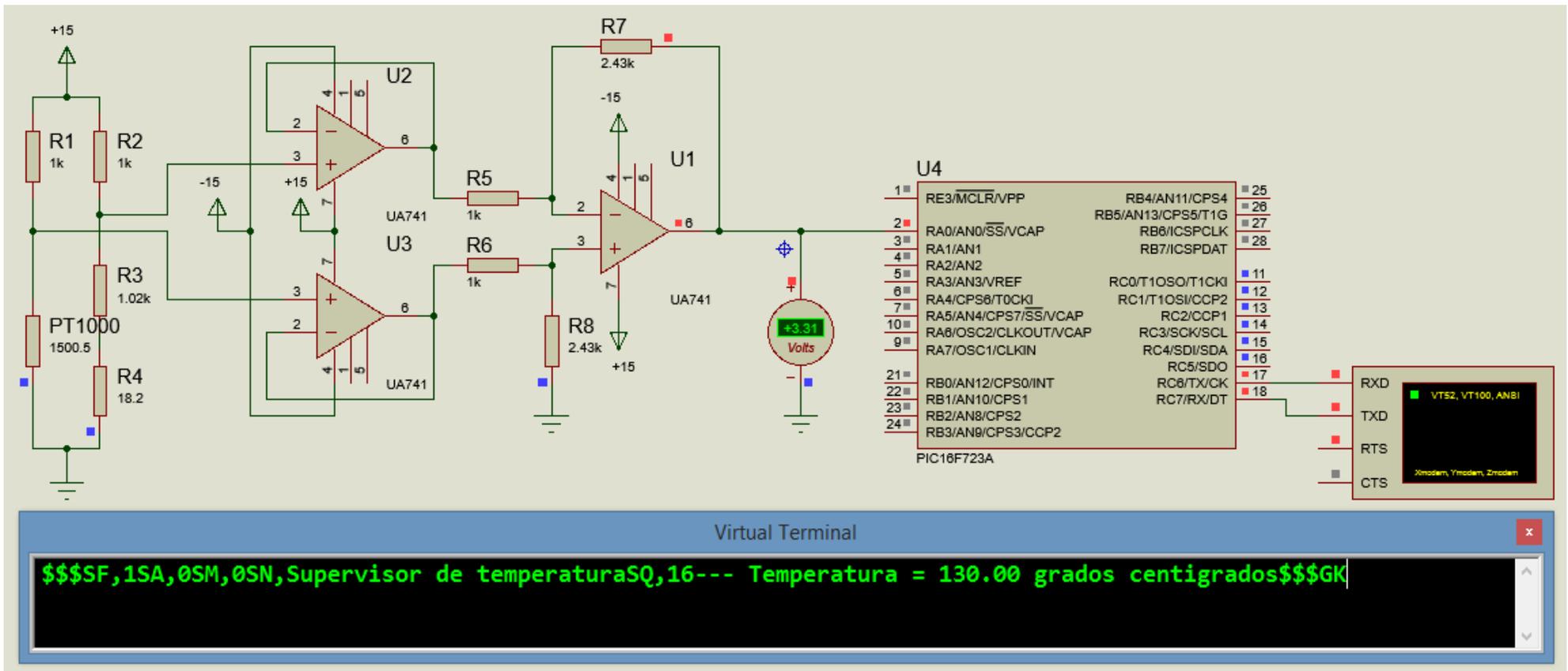


Ilustración 19: Simulación en Proteus del programa de control

Como podemos ver en la captura de pantalla de la simulación del programa de control, la secuencia de comandos enviados al módulo Bluetooth a través del puerto serie es la siguiente:

- “\$\$\$” hace que el módulo Bluetooth entre en modo *command*. El módulo respondería “CMD” para que el programa pudiera continuar su ejecución (líneas 127 y 128 del código).
- “SF,1” llama a los valores de configuración por defecto. El módulo respondería “AOK” y el programa enviaría el siguiente comando de configuración (líneas 131 y 132 del código).
- “SA,0” establece el modo abierto de autenticación para que el dispositivo que desee conectarse al supervisor no necesite introducir ningún PIN. El módulo respondería “AOK” y continuaría la configuración (líneas 135 y 136 del código).
- “SM,0” establece el modo esclavo o *slave* de operación. El módulo respondería “AOK” y se enviaría el siguiente comando (líneas 139 y 140 del código). Si el supervisor trabajase en modo maestro o *master* tendríamos que enviarle una orden cada vez que quisiéramos que estableciese una conexión con otro dispositivo.
- “SN,Supervisor de temperatura” asigna este nombre a nuestro dispositivo. El módulo respondería “AOK” y el programa enviaría el siguiente comando de configuración (líneas 143 y 144 del código).
- “SQ,16” activa el modo optimizado para la transferencia de datos. El módulo respondería “AOK” y el programa enviaría el siguiente comando para finalizar la configuración (líneas 148 y 149 del código).
- “---” para que el módulo Bluetooth salga del modo *command*. El módulo respondería “END” para que el programa continuase su ejecución (líneas 152 y 153 del código).
- Se envía la información de temperatura (líneas desde la 202 hasta la 205 del código).
- “\$\$\$” hace que el módulo Bluetooth entre en modo *command*. El módulo respondería “CMD” y el programa enviaría el comando siguiente (líneas 161 y 162 del código).
- “GK” pregunta al módulo Bluetooth el estado de conexión (línea 167 del código). El módulo respondería de dos maneras distintas:
 - Si hubiera una conexión respondería “1,0,0” y en ese caso el programa enviaría “---” para que el módulo Bluetooth saliese del modo *command*. El módulo respondería “END” (líneas desde la 174 hasta la 179 del código) y el bucle volvería al principio (línea 202 del código) para enviar de nuevo la información de temperatura.
 - Si no hubiera una conexión respondería “0,0,0” y en ese caso el programa enviaría “R,1” para que el módulo hiciera un *reboot*, después configuraría el módulo (líneas desde la 169 hasta la 173 del código) y el bucle volvería al principio (línea 202 del código) para enviar de nuevo la información de temperatura.

5. La fuente de alimentación

En este capítulo diseñaremos la fuente de alimentación de nuestro supervisor. Primero veremos brevemente los tipos de fuente existentes y después analizaremos los consumos de nuestro supervisor, datos imprescindibles para el posterior diseño de su fuente de alimentación.

5.1. Introducción a las fuentes de alimentación^{xxvii}

Sin entrar en detalles sobre las fuentes de alimentación y los bloques o módulos que las componen, dentro del alcance de nuestro proyecto nos interesa saber de la existencia de dos tipos básicos de fuentes de alimentación: las lineales y las conmutadas; que presentan una serie de ventajas e inconvenientes.

Por un lado, las fuentes lineales presentan un diseño muy simple y, por lo tanto, económico, sin embargo son poco eficientes en cuanto a la regulación de tensión, estando esta eficiencia comprendida entre el 30 y el 60 %, y sólo funcionan como reductores de tensión ofreciendo una única salida.

Por otro lado, las fuentes conmutadas presentan un diseño mucho más complejo, pese a que son de menor tamaño, y son mucho más eficientes que las fuentes lineales, estando esta eficiencia comprendida entre el 68 y el 90 %. Además, pueden funcionar como reductores (*buck*), elevadores (*boost*) o inversores de tensión (*charge pump*), con múltiples salidas.

5.2. Análisis de consumo

En el diseño de una fuente de alimentación es importante calcular previamente el consumo de nuestro circuito con la finalidad de dimensionar nuestra fuente correctamente para que proporcione la corriente suficiente que va a requerir nuestro diseño para su correcto funcionamiento.

Hemos visto que nuestro circuito consta de componentes activos que necesitan una alimentación simétrica o dual de ± 15 V y otros que necesitan una alimentación de 3,3 V. A través de las hojas de especificaciones de cada componente o *datasheets* podemos obtener información acerca de sus consumos.

Tomaremos los valores máximos en lugar de los típicos para ponernos en el peor de los casos y diseñar la fuente sin quedarnos cortos. En el caso del módulo Bluetooth, el valor máximo de consumo tiene lugar cuando está transmitiendo, y en el caso del microcontrolador, el valor máximo de consumo tiene lugar cuando tenemos corriente en todos sus pines:

Componente	Tensión de alimentación	Consumo
Amplificadores operacionales UA741CP	± 15 V	3,3 mA
Total fuente ± 15 V		9,9 mA
Microcontrolador PIC16F723	1,8-5,5 V	70 mA
Módulo Bluetooth RN42XV	3,3 V	30 mA
Total fuente 3,3 V		100 mA

Tabla 6: Análisis de consumo

5.3. Diseño de la fuente de alimentación^{xxviii xxix}

Partimos de la idea de que será más sencillo diseñar tres fuentes de alimentación, en las que a partir de la tensión que nos proporciona una fuente externa a nuestro supervisor nos genere las tres tensiones necesarias para el funcionamiento de los bloques internos, que diseñar una o dos fuentes en las que a partir de la tensión de entrada nos genere las tensiones necesarias en diferentes pasos o etapas.

Para el diseño utilizaremos la aplicación *online* denominada “Webench Design Center” que ofrece la página web de Texas Instruments, en la que introduciendo los datos de la tensión de entrada y la tensión y corriente de salida requeridos, nos facilitará varios diseños basados en la eficiencia, el coste y el tamaño (*footprint*).

Como se nos pide una fuente de alimentación eficiente y nuestro prototipo, en el caso de ser aprobado, podría ser producido a gran escala, será necesario que su coste sea el menor posible. Así pues, elegiremos un diseño que aúne ciertos compromisos de eficiencia, coste y tamaño.

5.3.1. Diseño de la fuente de + 15 V^{xxx}

Deseamos obtener 15 V a partir de los 24 V que nos entrega una fuente de alimentación externa a nuestro supervisor. Tenemos en cuenta que el consumo en corriente de los componentes que trabajan con esta tensión de alimentación es de 10 mA y para que nuestro diseño tenga un margen de operación suficiente duplicaremos ese valor de consumo.

Con estos datos, la aplicación Webench nos ofrece el siguiente diseño con menor *footprint*:

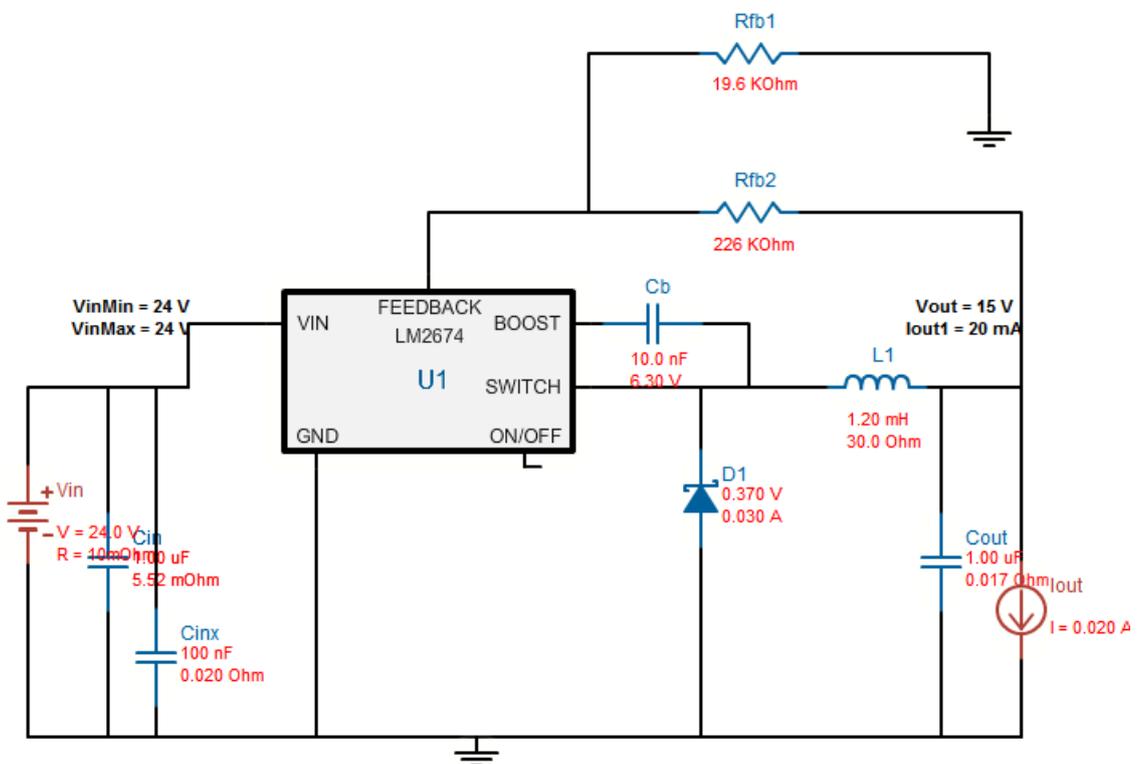


Ilustración 20: Esquema de la fuente de alimentación de + 15 V

Y una eficiencia del 74 %:

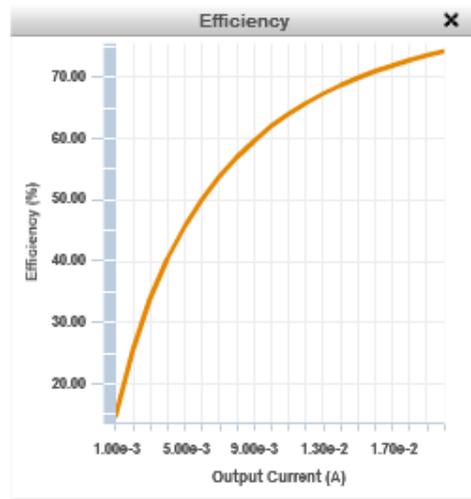


Ilustración 21: Eficiencia de la fuente de + 15 V

En cuanto a los componentes que acompañan al circuito integrado, hay que decir que los condensadores C_{in} y C_{inX} disminuyen el rizado de la tensión de entrada y el condensador C_{out} disminuye el rizado de la tensión de salida. Las resistencias R_{fb1} y R_{fb2} forman un divisor de tensión que fija la tensión de salida al valor deseado, en este caso 15 V, mediante el terminal *Feedback*.

Estos convertidores DC-DC incorporan en su interior un transistor MOSFET, entre sus elementos básicos, de modo que en el estado “ON”, que tiene lugar cuando este transistor conduce, la corriente va desde la entrada hasta la salida cargando la bobina L1 y manteniendo estable la tensión de salida a través del condensador C_{out} . El diodo D1 no conduce en este momento ya que está polarizado en inversa.

En el estado “OFF” que tiene lugar cuando el transistor está en estado de corte, éste no conduce y la energía almacenada en la bobina L1 hace circular una corriente por el circuito gracias al diodo D1 que entra en conducción al estar polarizado ahora en directa.

En este estado, el condensador C_b se carga durante la recirculación de la bobina L1 que permite el diodo D1 para, posteriormente, facilitar la conmutación del transistor al estado de conducción al suministrar la tensión necesaria a través del terminal *Boost*.

Los valores de estos componentes vienen dados por la aplicación Webench y no son para nada aleatorios, sino que están relacionados con la tensión y corriente de salida requeridos y varían en función del regulador elegido. El cálculo de dichos componentes también puede realizarse según los detalles que figuran en los *datasheets* de este tipo de reguladores.

5.3.2. Diseño de la fuente de - 15 V^{xxx}

Ahora deseamos obtener - 15 V a partir de los 24 V que nos ofrece la fuente de alimentación externa. Al igual que en el apartado 5.3.1, tendremos en cuenta el consumo en corriente de los componentes activos que trabajan con esta tensión de alimentación, que es de 10 mA, y lo duplicaremos para que nuestro diseño tenga un margen de trabajo más que suficiente.

La aplicación Webench nos ofrece el siguiente diseño con el menor *footprint*:

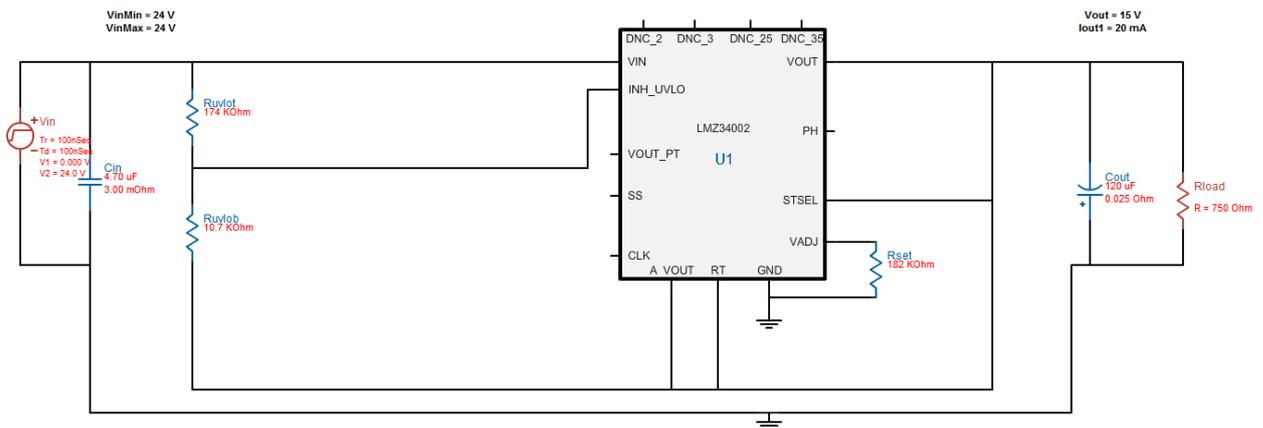


Ilustración 22: Esquema de la fuente de alimentación de - 15 V

Y con una eficiencia del 74 %:

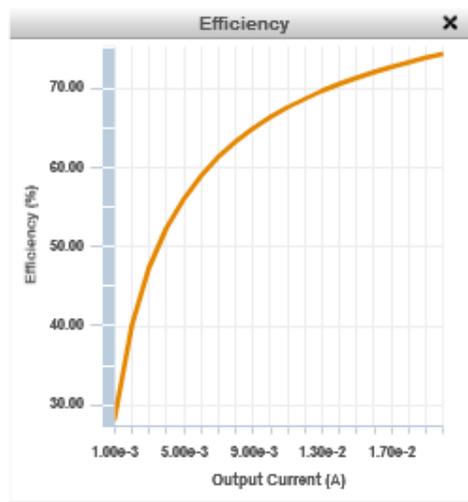


Ilustración 23: Eficiencia de la fuente de - 15 V

La función de los componentes que aparecen junto a este circuito integrado es:

- El condensador C_{in} disminuye el rizado de la tensión de entrada y el condensador C_{out} hace lo mismo con el rizado de la tensión de salida.
- La resistencia R_{set} es la que ajusta el valor de la tensión de salida, en este caso a 15 V, a través del terminal "Vadj".
- Las resistencias R_{vlot} y R_{vlob} forman un divisor de tensión que determina el nivel de la tensión o umbral en el que el transistor MOSFET interno cambia de estado entre "ON" y "OFF" a través del terminal "INH_UVLO".

5.3.3. Diseño de la fuente de 3,3 V^{xxxii}

Por último tenemos que obtener 3,3 V a partir los 24 V que nos ofrece la fuente externa de alimentación. De nuevo, como en los apartados anteriores, tenemos en cuenta que el consumo en corriente de los componentes activos que trabajan con esta tensión de alimentación es de 100 mA y para que nuestro diseño tenga un margen de funcionamiento suficiente tendremos en cuenta el doble de ese valor de consumo.

Con estos datos, el diseño con menor *footprint* que nos ofrece la aplicación *online* Webench es el siguiente:

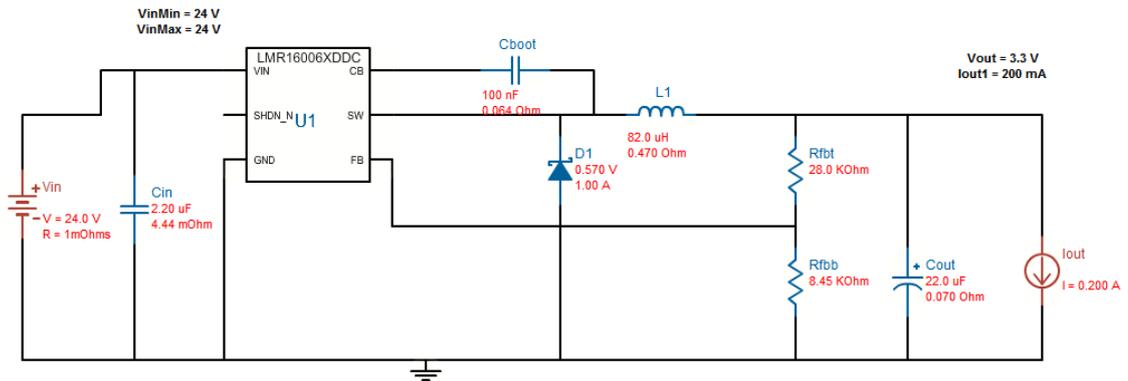


Ilustración 24: Esquema de la fuente de alimentación de 3,3 V

Con una eficiencia del 83 %:

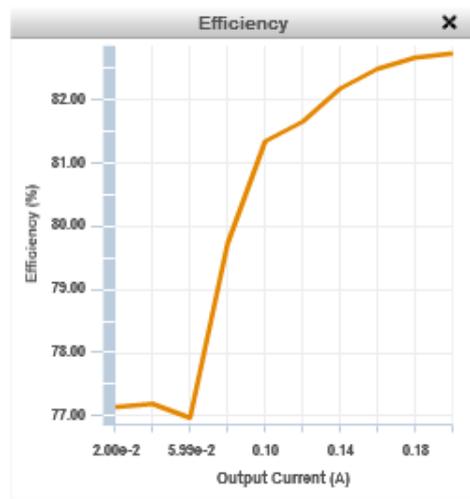


Ilustración 25: Eficiencia de la fuente de 3,3 V

El funcionamiento es prácticamente idéntico al comentado en el apartado 5.3.1, esto es:

- El condensador C_{in} disminuye el rizado de la tensión de entrada y el condensador C_{out} disminuye el rizado de la tensión de salida.
- Las resistencias R_{fbt} y R_{fbb} forman un divisor de tensión que fija la tensión de salida al valor deseado, en este caso 3,3 V, a través del terminal “FB”.
- En el estado “ON”, que tiene lugar cuando el transistor MOSFET conduce, la corriente va desde la entrada hasta la salida cargando la bobina L1 y manteniendo estable la tensión de salida a través de C_{out} . El diodo D1 no conduce en este momento ya que está polarizado en inversa.
- En el estado “OFF” que tiene lugar cuando el transistor se encuentra en estado de corte, éste no conduce y la energía almacenada en la bobina L1 hace circular una corriente por el circuito gracias al diodo D1 que entra en conducción al estar polarizado ahora en directa.
- En este último estado, el condensador C_{boot} se carga durante la recirculación de la bobina L1 que permite al diodo D1, para facilitar posteriormente la conmutación al estado de conducción del transistor al suministrar la tensión necesaria mediante el terminal “CB”.

6. Diseño del *layout* de la PCB^{xxxiii xxxiv}

En este capítulo vamos a diseñar la placa de circuito impreso o PCB de nuestro supervisor remoto de temperatura. En el diseño deberemos de tener en cuenta una serie de condicionantes que detallamos a continuación:

- Para el diseño del trazado de las pistas correspondientes a la fuente de alimentación, tomaremos como referencia la lista de componentes ofrecida por la aplicación *online* Webench de Texas Instruments donde todos ellos son componentes de montaje superficial o SMD (*Surface Mounted Device*).
- A pesar de que la aplicación Webench ofrece la opción de descargar el esquemático y/o el *layout* de la PCB en un formato de archivo reconocido por Eagle, no es posible realizar posteriormente la importación de cualquiera de esos dos archivos, por lo que nos veremos obligados a dibujar de nuevo el esquemático de las tres fuentes en Eagle para diseñar posteriormente el trazado de estas pistas.
- El módulo Bluetooth sólo puede montarse mediante orificios pasantes o *through hole* por lo que elegiremos el mismo tipo de montaje para el resto de circuitos integrados, como el microcontrolador y los amplificadores operacionales, a pesar de tener también disponible la versión de montaje superficial o SMD. De esta manera intentaremos diseñar las pistas de los bloques del puente de Wheatstone, adaptación de señal de temperatura, microcontrolador y módulo Bluetooth en una sola cara.
- Como estamos contemplando el diseño de una parte de la placa con componentes SMD y otra con componentes *through hole*, nos vemos obligados a utilizar una placa de doble cara. Además, es posible que en algunas circunstancias nos resulte inevitable o, en otras, de gran ayuda poder trazar alguna pista en la cara opuesta a la de nuestro interés.
- Acudimos de nuevo a nuestro proveedor de componentes electrónicos para consultar el tamaño de las placas de circuito impreso de doble cara y nos encontramos con que la de menor tamaño es de 160 x 100 mm, que creemos que es de un tamaño más que suficiente para alojar todo nuestro diseño.

Con la ayuda del software Eagle de CadSoft, dibujamos el esquemático completo de nuestro supervisor de temperatura para, posteriormente, poder colocar los componentes electrónicos sobre la PCB de nuestro supervisor siguiendo una disposición lógica similar a la del esquemático y diseñar el trazado de las pistas.

Para simplificar este diseño utilizaremos un plano de masa que permita conectar todos aquellos terminales de los componentes de nuestro circuito que van a masa en una gran y única pista. De este modo nos ahorramos una multitud de pistas que podrían complicar en exceso nuestro diseño.

Utilizaremos la herramienta "Autoroute" que diseña automáticamente el trazado de las pistas teniendo en cuenta el conexionado de los componentes según el esquemático y la disposición espacial de los mismos sobre la PCB. A pesar de la gran utilidad de esta herramienta, en algunas ocasiones será necesario retocar algún trazado e incluso trazarlo manualmente porque esta herramienta no haya podido completar el trazado al 100 %.

A continuación mostraremos el esquemático de nuestro diseño, la disposición espacial de sus componentes electrónicos y el trazado de las pistas en la capa superior e inferior.

6.1. Esquemático

En la siguiente imagen podemos apreciar el esquema elaborado con la aplicación Eagle de CadSoft, a partir del cual se realizará el diseño de la PCB.

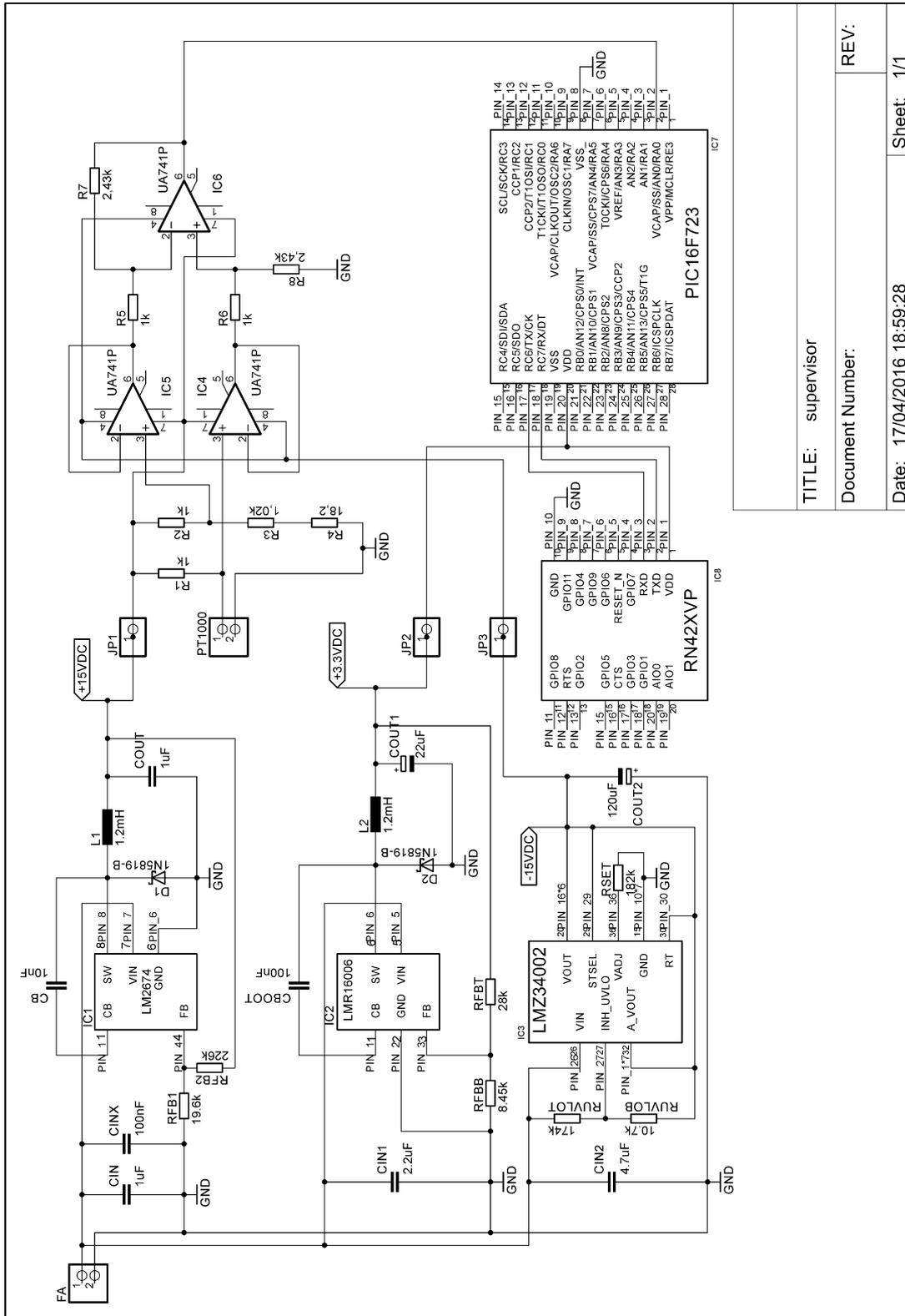


Ilustración 26: Esquemático de nuestro supervisor remoto de temperatura

TITLE: supervisor	REV:
Document Number:	
Date: 17/04/2016 18:59:28	Sheet: 1/1

6.2. Layouts de la PCB

Para el diseño del trazado de las pistas es imprescindible conocer el tamaño de los componentes, el número de sus pines, la disposición y función de los mismos, con la finalidad de que a la hora de soldar cada uno de los componentes a la PCB coincidan cada uno de los pines con las pistas trazadas.

La librería de componentes de Eagle incorpora una gran variedad de los componentes más usuales en los diseños electrónicos, pero no es difícil encontrarnos con la problemática de que los componentes que tengamos que utilizar en nuestro diseño no aparezcan en su librería.

En estos casos Eagle ofrece la posibilidad de añadir nuevos componentes en su librería, bien descargándolos de Internet, si es que estos están disponibles en alguna web, o bien creándolos como si fueran nuevos componentes mediante el dibujado de su huella o *footprint*, especificando su tamaño y la posición de sus pines, y el dibujado de su esquemático donde indicamos la función de cada uno de esos pines.

Una vez que hemos dibujado el componente o *device* y su esquemático, los asociaremos formando un paquete o *package*, el cual se incorpora a la librería para que pueda ser añadido al esquemático de nuestro supervisor.

Toda esta información está recogida en los *datasheets* de los componentes, de la que hacemos una pequeña recopilación a continuación. Comenzamos con el convertor DC-DC elegido para obtener la tensión de + 15 V.

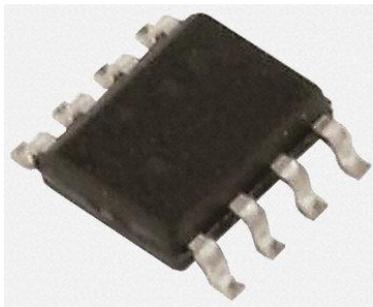


Ilustración 29: LM2674

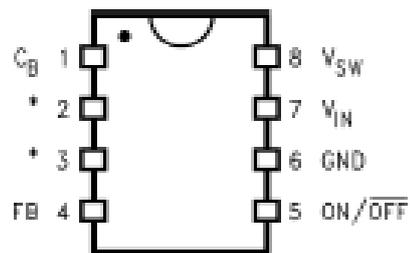


Ilustración 28: Pineado del LM2674

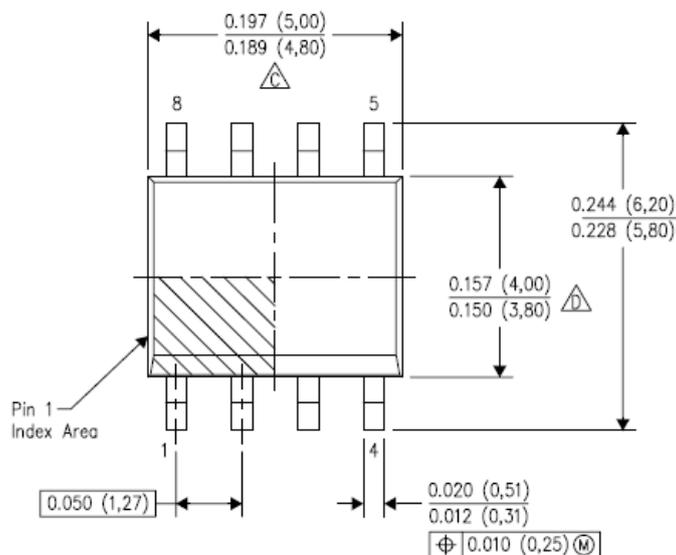


Ilustración 27: Dimensiones en pulgadas (y en milímetros) del LM2674

En cuanto al convertor DC-DC utilizado en la fuente de 3,3 V, estas son las características que hemos utilizado para añadirlo en la librería de Eagle:

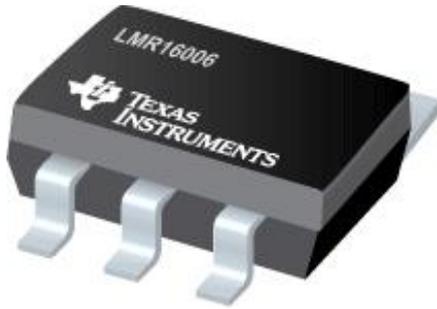


Ilustración 32: LMR16006

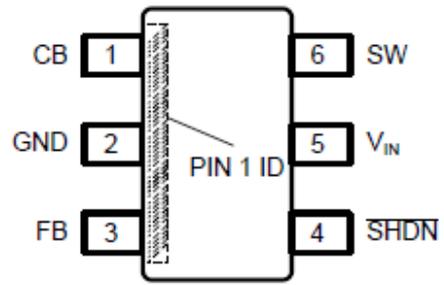


Ilustración 31: Pineado del LMR16006

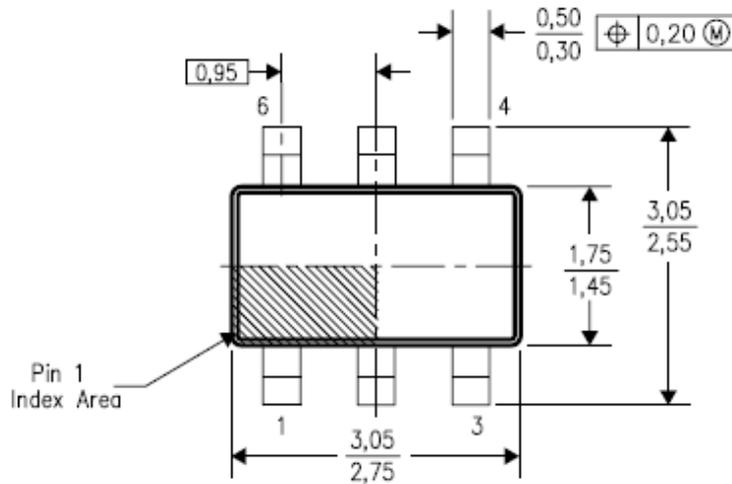


Ilustración 30: Dimensiones en milímetros del LMR16006

De la misma manera que lo hemos hecho con el convertor que nos entrega los - 15 V, donde destacamos el gran número de pines comunes, característica que puede facilitarnos la tarea del trazado de pistas:



Ilustración 33: LMZ34002

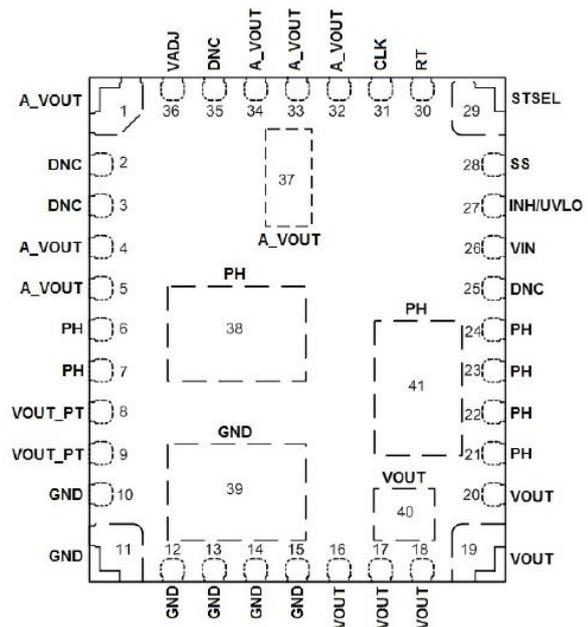


Ilustración 34: Pineado del LMZ34002

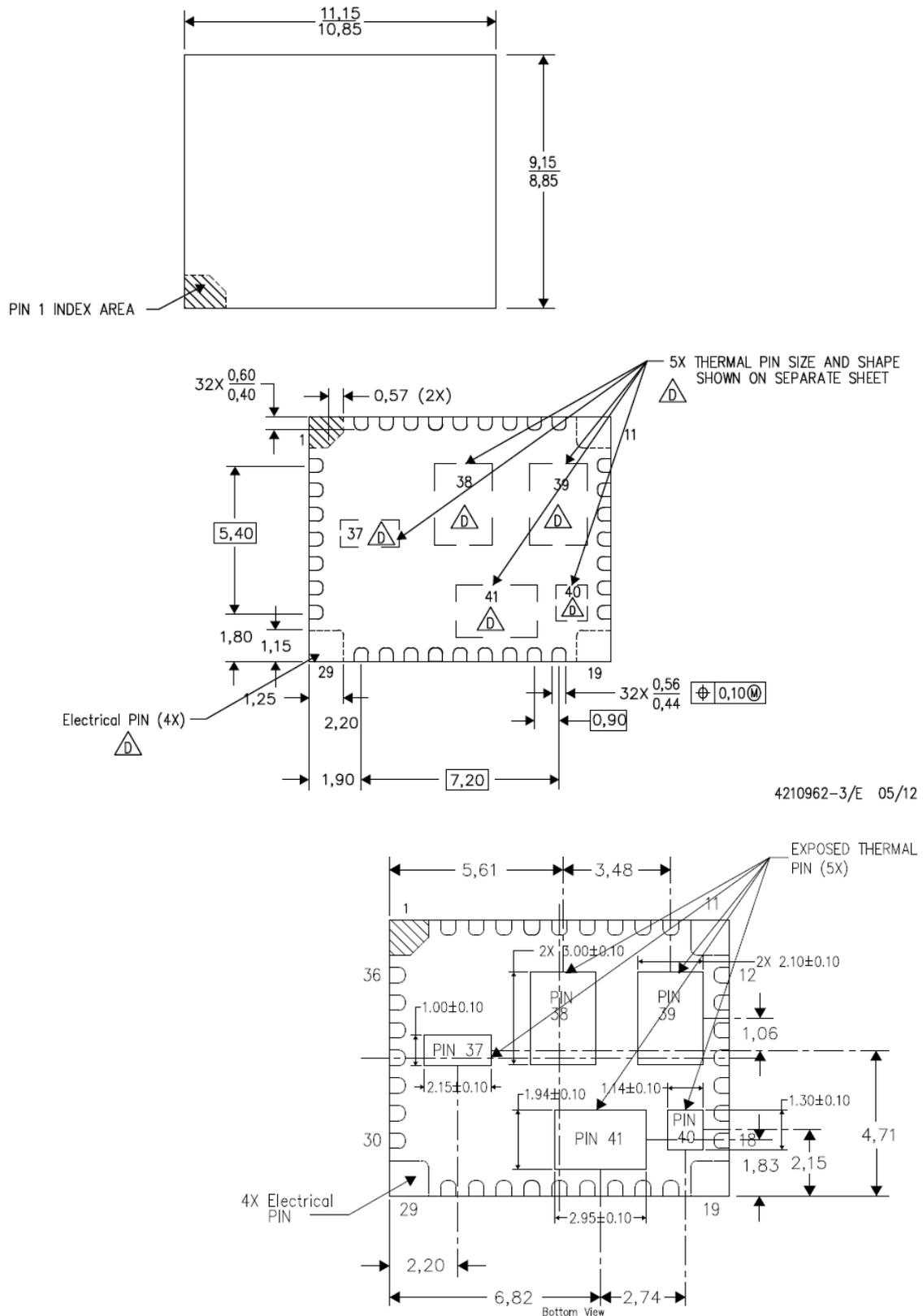


Ilustración 35: Dimensiones en milímetros del LMZ34002

La información relativa al conexionado del microcontrolador y sus dimensiones es la siguiente:

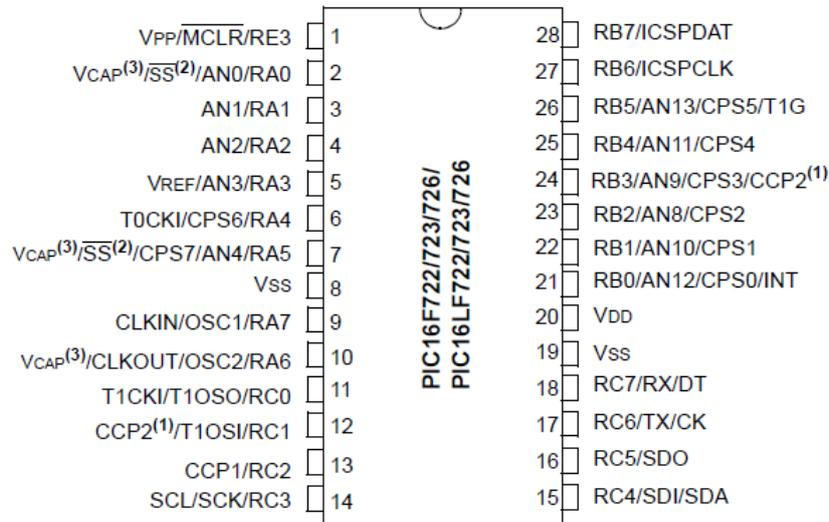
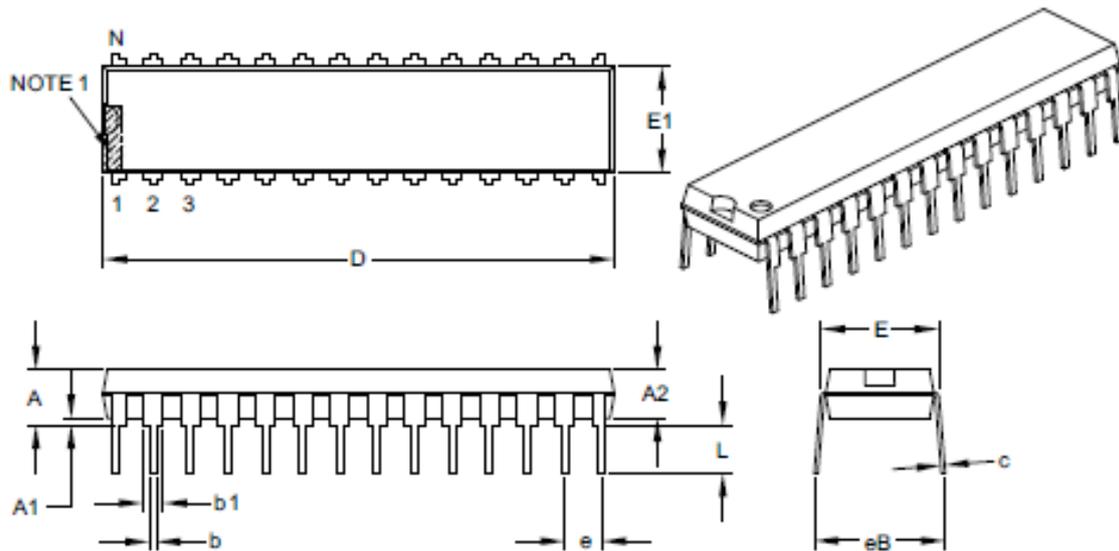


Ilustración 37: Pineado del PIC16F723



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	.100 BSC		
Top to Seating Plane	A	—	—	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	—	—
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	—	—	.430

Ilustración 36: Dimensiones en pulgadas del PIC16F723

Y la del módulo Bluetooth es la que aparece a continuación:

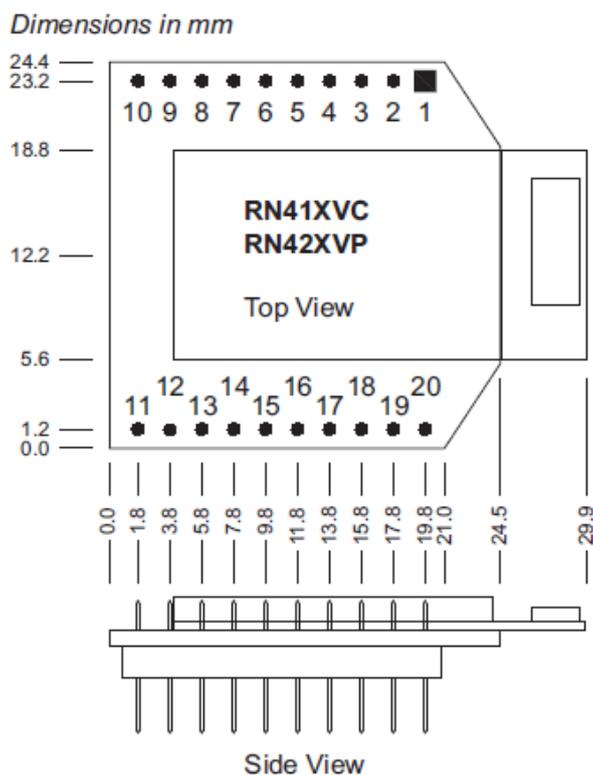


Ilustración 38: Dimensiones en milímetros del RN42XVP

Número de pin	Nombre de señal
1	VDD
2	TXD
3	RXD
4	GPIO7
5	RESET_N
6	GPIO6
7	GPIO9
8	GPIO4
9	GPIO11
10	GND
11	GPIO8
12	RTS
13	GPIO2
14	No asignada
15	GPIO5
16	CTS
17	GPIO3
18	GPIO7
19	AIO0
20	AIO1

Tabla 7: Pinedo del RN42XVP

En la siguiente imagen podemos apreciar cual será el acabado de nuestra placa, con la posición que ocuparán sus componentes y el trazado de las diferentes pistas y planos de masa en ambas caras de la placa.

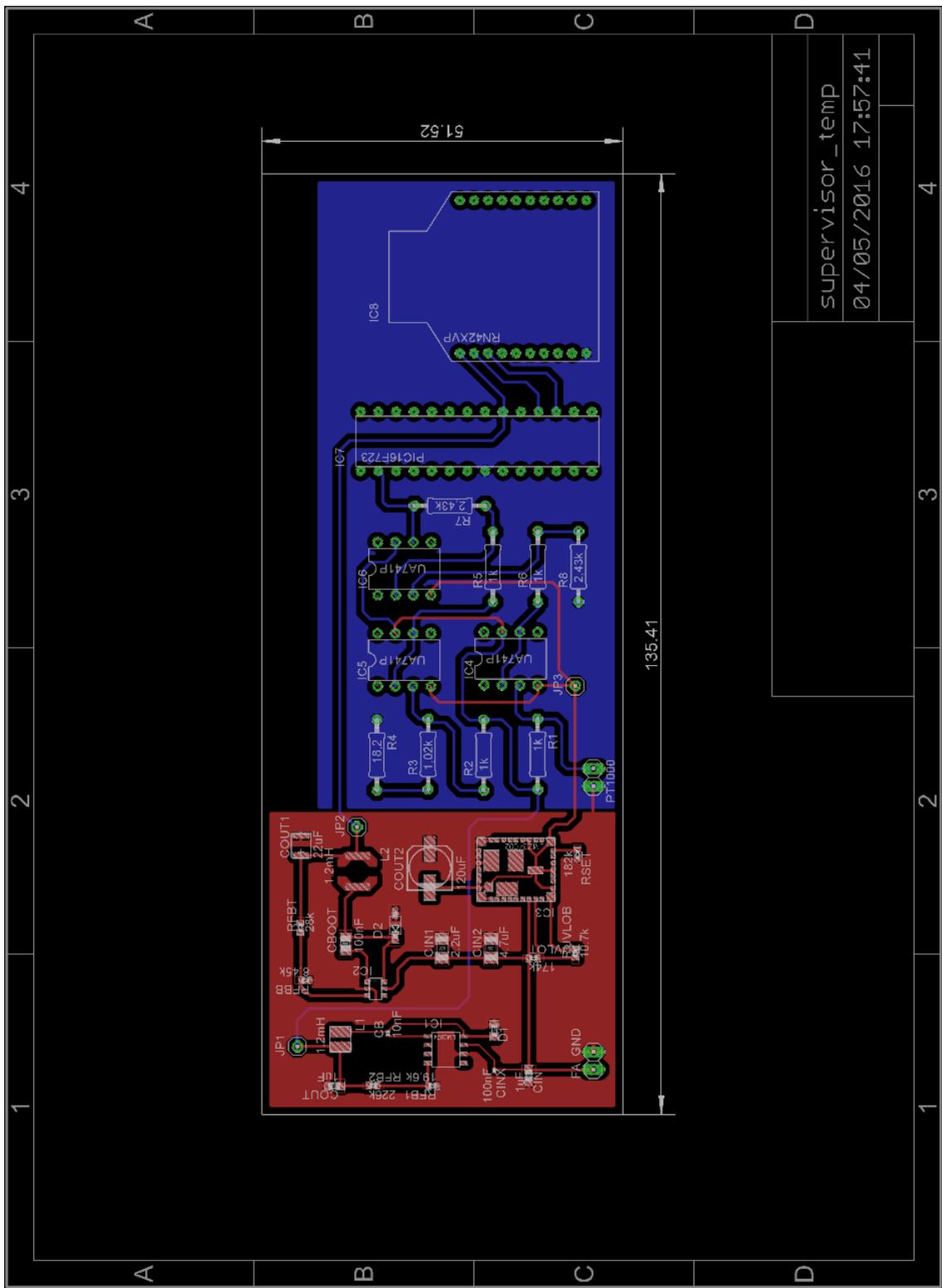


Ilustración 39: Diseño de la PCB de nuestro supervisor de temperatura

6.2.1. Disposición de los componentes en la PCB

Para mayor detalle adjuntamos esta imagen donde podemos apreciar la posición que ocuparán en la PCB los componentes de nuestro diseño. Todos ellos se alojarán sobre la capa superior o *top* de la placa.

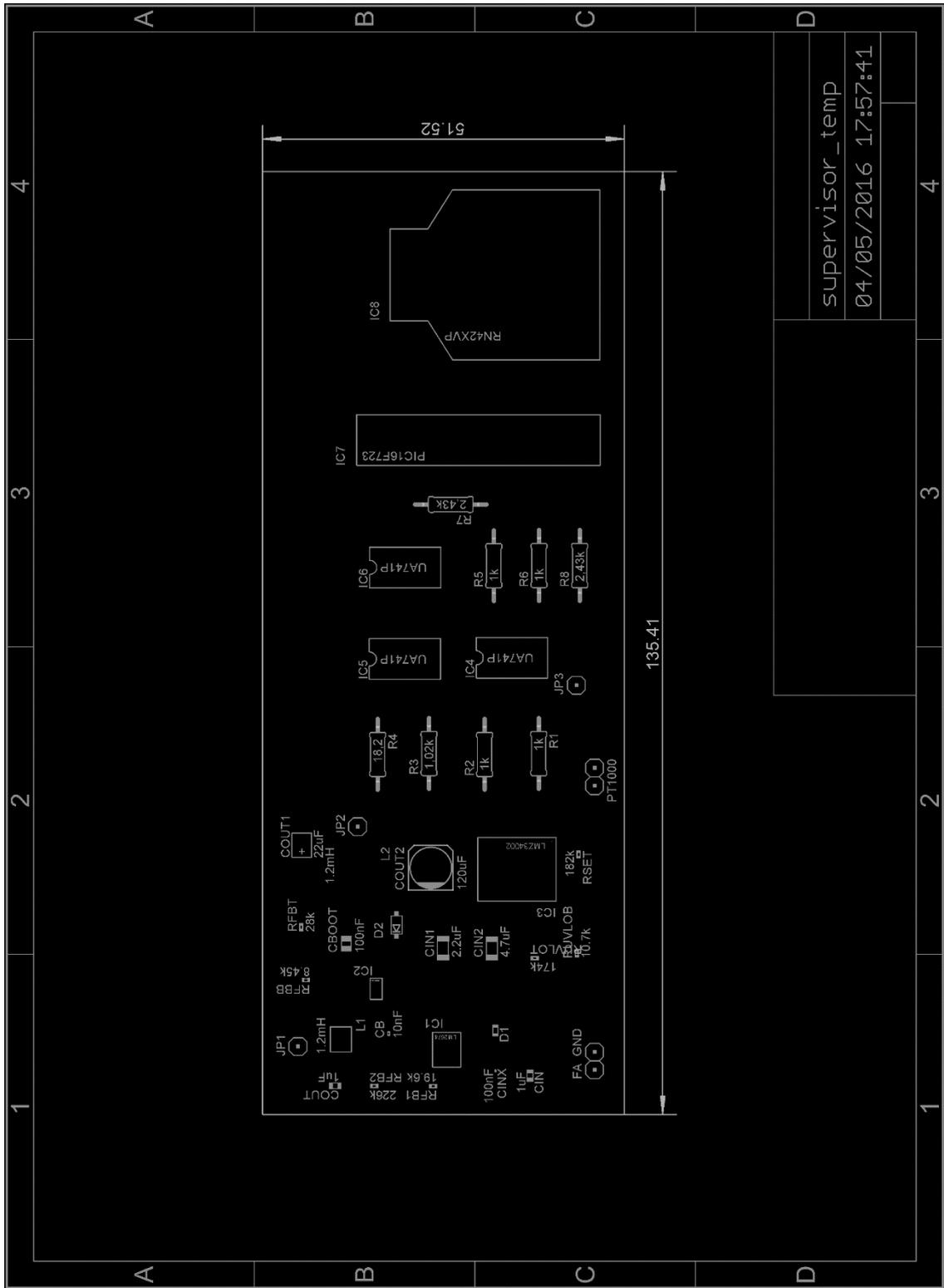


Ilustración 40: Disposición de los componentes en la PCB de nuestro supervisor remoto de temperatura

6.2.2. Trazado de las pistas

A continuación mostramos el trazado de las pistas y planos de masa de la cara superior o *top*, en color rojo, y de las pistas y plano de masa de la capa inferior o *bottom*, en color azul. Para facilitar la conexión entre pistas que discurren por una de las caras y que en un momento dado cambian de cara, utilizamos conectores macho de 1 pin.

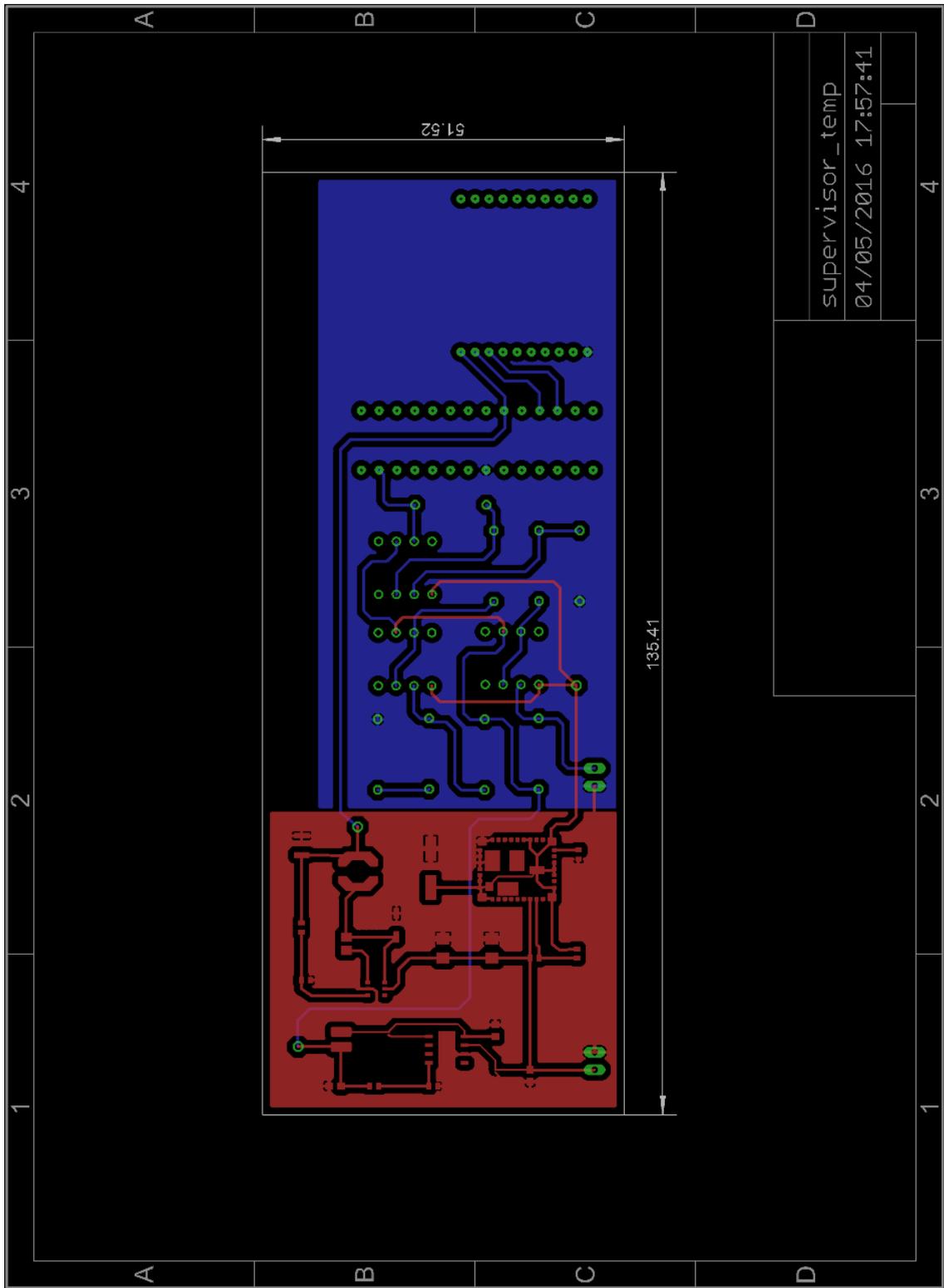


Ilustración 41: Trazado de las pistas en la PCB de nuestro supervisor remoto de temperatura

6.3. Valoración económica

A continuación analizaremos el coste económico de nuestro prototipo a partir de los precios de mercado de todos los componentes que son necesarios en este diseño. Para la realización de este presupuesto acudiremos una vez más a nuestro proveedor de componentes para consultar sus precios.

Componente	Descripción	Precio
Convertor 24V a 15V		
IC1	LM2674M-ADJ Regulador de conmutación, controlador Buck	3,28
CIN	Condensador cerámico multicapa 1 μ F 35 V (2 x 0,08 €)	0,16
CINX	Condensador cerámico multicapa 100 nF 10 % 50 V	0,06
COUT	Condensador cerámico multicapa 1 μ F 10 % 25 V	0,03
CB	Condensador cerámico multicapa 10 nF 10 % 6,3 V	0,01
D1	Diodo Schottky 30 mA 45 V	0,12
L1	Bobina 1,2 mH	0,18
RFB1	Resistencia 19,6 k Ω 1 % 0,063 W	0,01
RFB2	Resistencia 226 k Ω 1 % 0,063W	0,03
Convertor 24V a 3,3V		
IC2	LMR16006 Regulador de conmutación	2,78
CIN1	Condensador cerámico multicapa 2,2 μ F 10 % 50 V	0,19
COUT1	Condensador de tántalo 22 μ F 20 % 8 V	0,88
CBOOT	Condensador cerámico multicapa 0,1 μ F 10 % 50 V	0,02
D2	Diodo Schottky 1 A 60 V	0,25
L2	Bobina 82 μ H	0,43
RFBB	Resistencia 8,45 k Ω 1 % 0,063 W	0,01
RFBT	Resistencia 28 k Ω 1 % 0,063 W	0,02
Convertor 24 V a - 15V		
IC3	LMZ34002 Convertidor DC-DC	13,90
CIN2	Condensador cerámico multicapa 4,7 μ F 10 % 50 V	0,53
COUT2	Condensador electrolítico de aluminio 120 μ F 20 % 20 V	1,10
RSET	Resistencia 182 k Ω 1 % 0,063 W	0,01
RUVLOB	Resistencia 10 k Ω 1 % 0,063 W	0,04
RUVLOT	Resistencia 178 k Ω 1 % 0,063 W	0,05
Circuito de adaptación de señal de temperatura		
R1, R2, R5, R6	Resistencia 1 k Ω 1 % 0,05 W (4 x 0,18 €)	0,72
R3	Resistencia 1,02 k Ω 0,1 % 0,25 W	0,41
R4	Resistencia 18,2 Ω 0,1 % 0,25 W	0,37
R7, R8	Resistencia 2,43 k Ω 1 % 0,125 W (2 x 0,10 €)	0,20
PT1000	Sensor de temperatura de resistencia de platino	44,03
IC4, IC5, IC6	LM741CN Amplificador operacional (3 x 0,63 €)	1,89
Microcontrolador		
IC7	PIC16F723-I/SP Microcontrolador	1,79
Módulo Bluetooth		
IC8	RN42XVP-I/RM Módulo Bluetooth	21,94
PCB		
	Placa foto resistente doble cara 160x100x1,6 mm grosor 70 μ m	10,94
Conectores		
FA, PT1000	Terminal para PCB con tornillo para 2 vías (2 x 0,50 €)	1,00
JP1, JP2, JP3	Conector macho recto 1 pin (3 x 0,03 €)	0,09
TOTAL		107,47 €

Tabla 8: Valoración económica del supervisor remoto de temperatura vía Bluetooth

7. Conclusiones y ampliaciones

En este capítulo vamos a hablar sobre: el cumplimiento de los objetivos que nos marcamos al inicio del proyecto, los problemas que nos hemos encontrado durante la ejecución del proyecto, las lecciones que hemos aprendido con nuestros errores y con los contratiempos encontrados, y las soluciones que hemos aportado para minimizar el impacto de esos problemas. También aportaremos ideas que puedan constituir líneas de ampliación futuras.

7.1. Reflexión sobre los objetivos

Damos por alcanzados los objetivos de esta asignatura ya que hemos puesto en práctica los conocimientos teóricos adquiridos en algunas de las asignaturas que hemos cursado a lo largo de los estudios de esta Ingeniería, sobre todo aquellas en las que se impartían contenidos que hemos utilizado en esta área de TFC, como son electricidad, electrónica y programación.

Hemos puesto en práctica las habilidades de comunicación oral y escrita mediante los documentos escritos y la presentación virtual que hemos entregado. Estas habilidades las empezamos trabajando con la asignatura de “Competencias comunicativas para profesionales de la TIC” y las hemos ido desarrollando a lo largo de toda la carrera.

Hemos llevado a cabo un proyecto en el que hemos vivido en primera persona sus cinco fases del ciclo de vida (iniciación, planificación, ejecución, seguimiento y control, y cierre) y hemos tenido la oportunidad de poner en práctica los conocimientos adquiridos en la asignatura de “Gestión de proyectos”.

También damos por conseguidos los objetivos marcados para este proyecto en concreto. Hemos diseñado un supervisor remoto de temperatura vía Bluetooth así como los diferentes bloques que lo componen, esto es, el bloque de adaptación de señal de temperatura y el bloque de la fuente de alimentación con la ayuda de una aplicación específica.

Hemos implementado un programa en lenguaje C para el microcontrolador y hemos simulado su funcionamiento y el del resto del circuito con herramientas de simulación de circuitos electrónicos. Y hemos diseñado una PCB para nuestro supervisor con la herramienta correspondiente.

7.2. Problemas surgidos

Una de las tareas que nos ha llevado más tiempo del previsto ha sido el aprendizaje en la utilización de algunas de las diferentes aplicaciones que teníamos que utilizar en la ejecución del proyecto. Herramientas como TINA, Proteus o PIC C Compiler son bastante intuitivas y no llevan mucho tiempo hasta que el usuario se acostumbra a tener localizadas las diferentes herramientas y menús.

Sin embargo, Eagle ha sido la aplicación que nos ha requerido mayor cantidad de tiempo porque hemos tenido que aprender a añadir componentes electrónicos en su librería para poder realizar el diseño de la PCB. Aunque existe una amplia variedad de video tutoriales en Internet, hay que emplear bastante tiempo en seleccionar videos que realmente merezcan la pena y después visionarlos.

Durante el diseño de la fuente de alimentación nos surgió otro problema relacionado con el software. Nuestra intención fue la de realizar una simulación con los diseños que obtuvimos con la aplicación *online* Webench, pero nos resultó imposible porque ni la librería de TINA ni la de Proteus incluía ninguno de los convertidores DC-DC propuestos por Webench y tampoco encontramos el modo de poder importar esos modelos.

También nos llevó bastante tiempo comprender el funcionamiento de las múltiples opciones que ofrece Webench Design Center y los diferentes diseños que ofrecía cada una de ellas. Una de las ventajas que ofrece Webench es la posibilidad de exportar el esquema y el *layout* de la PCB en un formato compatible con Eagle, pero no pudimos aprovecharla porque Eagle no llegaba a reconocer los archivos generados por Webench.

Cuando tuvimos que simular en Proteus el programa de control del microcontrolador que interactúa con nuestro módulo Bluetooth y en cuya librería no aparece, nuestra intención fue la de emular las respuestas del módulo Bluetooth. Para ello configuramos un puerto serie virtual en nuestro ordenador pero el esfuerzo fue inútil porque no encontramos el modo de que funcionase realmente. Por ese motivo nos limitamos a comprobar que del puerto serie del microcontrolador salían las instrucciones para el módulo Bluetooth.

7.3. Soluciones aportadas

Cuando una tarea, como la de aprender a utilizar una aplicación necesaria para la realización del proyecto, ha requerido más tiempo del previsto inicialmente, hemos tenido que dedicarle un tiempo extra que hemos obtenido o bien por la buena marcha de otras tareas que se han completado en un tiempo menor al programado o bien restándolo de nuestro tiempo personal.

El seguimiento del consultor es imprescindible. Durante la iniciación y la planificación del proyecto, etapas en las que el TFC nos parece un mundo, nos hace ver que con la descomposición del proyecto en tareas y subtareas todo es posible, algo de lo que nos vamos dando cuenta a lo largo de la ejecución y que nos permite comprobar de lo que somos capaces de hacer cuando llegamos al cierre.

Nos ha ofrecido una disponibilidad y una inmediatez en sus respuestas que no me cansaré nunca de reconocer y agradecer. Hemos contado siempre con su apoyo y cuando le hemos planteado cualquier consulta nos ha ofrecido sus consejos y pistas para que nosotros mismos encontráramos el camino correcto.

7.4. Lecciones aprendidas

Como hemos podido comprobar, una correcta planificación y una acertada descomposición de las tareas en otras más pequeñas son claves para el éxito de cualquier proyecto que nos podamos plantear. La perseverancia y la constancia en el trabajo diario también son ingredientes indispensables para llevar a buen término nuestros propósitos.

Aunque ya habíamos tenido contacto con alguna herramienta de simulación de circuitos en asignaturas como la de Fundamentos Tecnológicos II, no ha sido hasta ahora cuando nos hemos metido de lleno con estas aplicaciones.

También hemos aprendido a programar microcontroladores en lenguaje C, similar al lenguaje Java que conocimos para realizar ejercicios en la asignatura de Programación orientada a objetos.

Pero quizás lo que más nos ha impactado es la aplicación Eagle y lo fácil que puede ser con ella el diseño del *layout* de una PCB. Nada que ver a los diseños que teníamos que hacer en las prácticas de laboratorio hace unos 20 años cuando estudiábamos Formación Profesional en Electrónica de comunicaciones.

7.5. Líneas de ampliación futuras

Para dar continuidad a nuestro proyecto o dar pie a otros nuevos se nos ocurre la posibilidad de modificar alguna o algunas de las características de nuestro supervisor o añadir otras nuevas, como por ejemplo:

- Dado que nuestro microcontrolador dispone de más entradas analógicas, podríamos añadir otros tipos de sensores que pudieran medir otras magnitudes físicas, como la humedad o la luminosidad por ejemplo, y esto implicaría añadir más circuitos de adaptación de señales.
- Podríamos requerir una mayor precisión en la medida de la temperatura y esto implicaría buscar otro microcontrolador que disponga de un conversor A/D con mayor resolución, de 10 bits o más.
- Podríamos reemplazar el módulo Bluetooth por otro tipo de módulo de comunicaciones inalámbricas, como WiFi, 3G, LTE...
- Añadir más circuitos de adaptación de señales, un cambio de microcontrolador o de módulo de comunicaciones, implicaría cambios en el código del programa de control, en el diseño de la fuente de alimentación, del esquemático y del *layout* de la PCB.
- Nuestro programa de control realizaba el envío de la información de temperatura cada minuto, pero podríamos proponer que dicha información se enviara con mayor periodicidad o sólo si se produjese un aumento o disminución en la temperatura.
- Podríamos proponer el diseño de la parte receptora, esto es, el receptor de temperatura vía Bluetooth, que dispondría de un módulo Bluetooth y de un microcontrolador que recibiría la información de temperatura del supervisor y que podría mostrarla a través de un panel LED, como los presentes en recintos de gran afluencia de público que muestran la temperatura y la humedad relativa en el interior del local.

Glosario

Amplificador operacional es un dispositivo amplificador electrónico de alta ganancia acoplado en corriente continua que tiene dos entradas y una salida.

Bimetal, unión de dos metales de diferentes coeficientes de expansión de manera que la pieza se dobla en una dirección cuando se calienta, y en la otra cuando se enfría, y se puede utilizar para abrir o cerrar circuitos eléctricos.

Bps o bit/s en una transmisión de datos es el número de impulsos elementales (1 o 0) transmitidos en cada segundo. Los bits por segundo son utilizados para expresar la velocidad de transmisión de datos o *bit rate*.

Bluetooth es una especificación para Redes Inalámbricas de Área Personal (WPAN – *Wireless Personal Area Network*) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda de los 2,4 GHz.

CISC (*Complex Instruction Set Computing*) es un tipo de microprocesador con un conjunto de instrucciones muy amplio que permite operaciones complejas entre operandos situados en la memoria o en los registros internos, en contraposición a la arquitectura RISC.

Conversor Analógico-Digital es un dispositivo electrónico capaz de convertir una señal analógica en una señal digital mediante un cuantificador y codificándose en un código binario en particular. Cada valor numérico binario se hace corresponder a un solo valor analógico.

dBm es una unidad de medida utilizada en telecomunicaciones para expresar la potencia absoluta mediante una relación logarítmica. El dBm se define como el nivel de potencia en decibelios en relación a un nivel de referencia de 1 mW.

Layout es la disposición de componentes y rutas sobre un diseño o la disposición de los elementos que forman un equipo electrónico.

Microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Incluye en su interior las tres principales unidades funcionales: unidad central de procesamiento, memoria y periféricos de entrada/salida.

MOSFET (*Metal-Oxide-Semiconductor Field-Effect Transistor* o transistor de efecto de campo metal-óxido semiconductor) es un transistor utilizado para amplificar o conmutar señales electrónicas. Tiene cuatro terminales llamados fuente (S, *Source*), drenador (D, *Drain*), puerta (G, *Gate*) y sustrato (B, *Body*), que suele estar conectado internamente al terminal de fuente y por este motivo se pueden encontrar dispositivos MOSFET de tres terminales.

NTC (*Negative Temperature Coefficient*) es un termistor con coeficiente de temperatura negativo, es decir, disminuye su resistencia al aumentar la temperatura.

PCB (*Printed Circuit Board* o placa de circuito impreso) es la superficie constituida por pistas o buses de material conductor laminadas sobre una base no conductora. Se utiliza para conectar eléctricamente a través de las pistas conductoras, y sostener mecánicamente, por medio de la base, un conjunto de componentes electrónicos. Las pistas son generalmente de cobre mientras que la base se fabrica generalmente de resinas de fibra de vidrio, cerámica, plástico, teflón o polímeros como la baquelita.

Piconet es una red cuyos nodos se conectan utilizando Bluetooth, puede constar de dos a siete dispositivos donde habrá siempre un maestro y los demás serán esclavos.

PTC (*Positive Temperature Coefficient*) es un termistor con coeficiente de temperatura positivo, es decir, aumenta su resistencia al aumentar la temperatura.

Puente de Wheatstone se utiliza para medir resistencias desconocidas mediante el equilibrio de los brazos del puente. Estos están constituidos por cuatro resistencias que forman un circuito cerrado, siendo una de ellas la resistencia desconocida.

RAM (*Random Access Memory* o memoria de acceso aleatorio) memoria principal de la computadora, donde residen programas y datos, sobre la que se pueden efectuar operaciones de lectura y escritura.

RISC (*Reduced Instruction Set Computer*) es un tipo de microprocesador con instrucciones de tamaño fijo y presentadas en un reducido número de formatos donde sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos. Suelen disponer de muchos registros de propósito general.

RTD (*Resistance Temperature Detector*) es un detector de temperatura resistivo, es decir, un sensor de temperatura basado en la variación de la resistencia de un conductor con la temperatura.

Scatternet es un grupo de *piconets*.

Seguidor de tensión es el circuito que proporciona a la salida la misma tensión que a la entrada. La impedancia de entrada es elevada, la de salida prácticamente nula, y es útil para eliminar efectos de carga o para adaptar impedancias y realizar mediciones de tensión de un sensor con una intensidad muy pequeña que no afecte sensiblemente a la medición.

SMD (*Surface Mounting Device* o dispositivo de montaje superficial) es un componente electrónico que se suelda directamente en la superficie de la PCB.

Termistor es un sensor resistivo de temperatura. Su funcionamiento se basa en la variación de la resistividad que presenta un semiconductor con la temperatura.

Termopar está por la unión de dos metales distintos que produce una diferencia de potencial, del orden de los milivoltios, que es función de la diferencia de temperatura entre uno de los extremos denominado punto caliente o de medida y el otro llamado punto frío o de referencia.

Through hole es un tipo de tecnología que utiliza los agujeros que se practican en las placas de los circuitos impresos para el montaje de los diferentes elementos electrónicos y para crear puentes eléctricos entre una de las caras de la placa de montaje a la otra, mediante un tubo conductor, que por lo general es una aleación de cinc, cobre y plata, para evitar su oxidación y permitir su soldadura. Los *Through hole* suelen ser bastante delicados y sensibles al calor. Y si se calientan demasiado se puede comprometer el contacto entre las pistas de una de las caras del circuito y la otra, resultando inoperante y dejando inútil la placa.

UART (*Universal Asynchronous Receiver-Transmitter* o Transmisor-Receptor Asíncrono Universal) es el dispositivo que controla los puertos y dispositivos serie, y se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo.

Ultra wide band es una especificación para WPAN que permite paquetes de información muy grandes (480 Mbps) en distancias cortas de unos pocos de metros.

Wibree también denominada como Bluetooth LE (*Low Energy*), Bluetooth ULP (*Ultra Low Power*) y Bluetooth Smart, es una nueva tecnología digital de radio inalámbrica interoperable para pequeños dispositivos.

Zigbee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo basada en el estándar IEEE 802.15.4 de WPAN.

Bibliografía

- ⁱ **Sáenz Higuera, Nita; Vidal Oltra, Rut.** *Redacción de textos científico-técnicos*. Barcelona: UOC.
- ⁱⁱ **Beneito Montagut, Roser.** *Presentación de documentos y elaboración de presentaciones*. Barcelona: UOC.
- ⁱⁱⁱ **Clarís Viladrosa, Robert.** *Introducción al trabajo final*. Barcelona: UOC.
- ^{iv} **Ramón Rodríguez, José.** *El trabajo final como proyecto*. Barcelona: UOC.
- ^v **Ramón Rodríguez, José.** *La gestión del proyecto a lo largo del trabajo final*. Barcelona: UOC.
- ^{vi} **Bataller Díaz, Alfons.** *Gestión y desarrollo de proyectos*. Barcelona: UOC.
- ^{vii} **Satue Villar, Antonio.** *Redes personales sin hilos*. Barcelona: UOC.
- ^{viii} **García, Albert.** *Que es Bluetooth*. Bluezona [artículo en línea]. [Fecha de consulta: 9 de marzo de 2016]. <<http://www.bluezona.com/que-es-bluetooth/>>
- ^{ix} *Bluetooth*. CCM [artículo en línea]. [Fecha de consulta: 9 de marzo de 2016]. <<http://es.ccm.net/contents/70-bluetooth>>
- ^x *Bluetooth*. Wikipedia [artículo en línea]. [Fecha de consulta: 9 de marzo de 2016]. <<https://es.wikipedia.org/wiki/Bluetooth>>
- ^{xi} **Millán Tejedor, Ramón Jesús.** *La explosión de Bluetooth: tecnología, aplicaciones y productos* [artículo en línea]. [Fecha de consulta: 9 de marzo de 2016]. <<http://www.ramonmillan.com/tutoriales/bluetooth.php>>
- ^{xii} *Módulo Bluetooth RN42XVP-I/RM*. RS [artículo en línea]. [Fecha de consulta: 11 de marzo de 2016]. <<http://es.rs-online.com/web/p/modulos-bluetooth/7985359/>>
- ^{xiii} *RN41XV & RN42XV Bluetooth module*. Roving Networks [pdf]. [Fecha de consulta: 11 de marzo de 2016]. <<http://ww1.microchip.com/downloads/en/DeviceDoc/RN41XV-RN42XV-ds-v1.0r.pdf>>
- ^{xiv} **Roselló Canal, Mar.** *Aplicaciones electromagnéticas y electrónicas*. Barcelona: UOC.
- ^{xv} *Sensor de temperatura de resistencia de platino RS Pro PT1000*. RS [artículo en línea]. [Fecha de consulta: 13 de marzo de 2016]. <<http://es.rs-online.com/web/p/sensores-de-temperatura-con-resistencia-de-platino/7621130/>>
- ^{xvi} *Platinum silicone patch sensors*. RS [pdf]. [Fecha de consulta: 13 de marzo de 2016]. <<http://docs-europe.electrocomponents.com/webdocs/10d2/0900766b810d29b2.pdf>>
- ^{xvii} *Getting started with TINA-TI*. Texas Instruments [pdf]. [Fecha de consulta: 15 de marzo de 2016]. <<http://www.ti.com/lit/ug/sbou052a/sbou052a.pdf>>
- ^{xviii} *Amplificador operacional UA741*. RS [artículo en línea]. [Fecha de consulta: 17 de marzo de 2016]. <<http://es.rs-online.com/web/p/amplificadores-operacionales/0305311/>>

^{xi} *μA741 General purpose operational amplifiers*. Texas Instruments [artículo en línea]. [Fecha de consulta: 17 de marzo de 2016]. <<http://www.ti.com/product/uA741>>

^{xx} **Roselló Canal, Mar**. *Aplicaciones electromagnéticas y electrónicas*. Barcelona: UOC.

^{xxi} *Microcontrolador PIC16F723-I/SP*. RS [artículo en línea]. [Fecha de consulta: 19 de marzo de 2016]. <<http://es.rs-online.com/web/p/microcontroladores/6668649/>>

^{xxii} *PIC16F723*. Microchip [artículo en línea]. [Fecha de consulta: 19 de marzo de 2016]. <<http://www.microchip.com/wwwproducts/en/PIC16F723>>

^{xxiii} *PIC16(L)F722/3/4/5/6/7*. Microchip [pdf]. [Fecha de consulta: 19 de marzo de 2016]. <<http://ww1.microchip.com/downloads/en/DeviceDoc/40001341F.pdf>>

^{xxiv} *CCS C Compiler Manual*. Custom Computer Services, Inc. [pdf]. [Fecha de consulta: 19 de abril de 2016]. <https://www.ccsinfo.com/downloads/ccs_c_manual.pdf>

^{xxv} *RN41/RN42 Bluetooth data module command & advanced information user's guide*. Microchip [pdf]. [Fecha de consulta: 19 de abril de 2016]. <http://ww1.microchip.com/downloads/en/DeviceDoc/bluetooth_cr_UG-v1.0r.pdf>

^{xxvi} *Tutorial del simulador Proteus ISIS con PIC16F628A*. Tecmikro [artículo en línea]. [Fecha de consulta: 24 de abril de 2016]. <<http://programarpicenc.com/articulos/tutorial-del-simulador-proteus-isis-con-pic16f628a/>>

^{xxvii} *Tipos de fuentes de alimentación*. [artículo en línea]. [Fecha de consulta: 7 de abril de 2016]. <<http://tiposdefuentesdealimentacion.blogspot.com.es/2011/05/tipos-de-fuentes-de-alimentacion.html>>

^{xxviii} *Webench Design Center*. Texas Instruments [artículo en línea]. [Fecha de consulta: 7 de abril de 2016]. <<http://www.ti.com/lstds/ti/analog/webench/overview.page>>

^{xxix} *Convertidores DC/DC - Buck*. [artículo en línea]. [Fecha de consulta: 7 de abril de 2016]. <<https://nomadaselectronicos.wordpress.com/2015/04/12/convertidores-dcdc-buck/>>

^{xxx} *LM2674 Step-Down Regulator*. Texas Instruments [artículo en línea]. [Fecha de consulta: 7 de abril de 2016]. <<http://www.ti.com/product/LM2674>>

^{xxxi} *LMZ34002 Negative Output Power Module*. Texas Instruments [artículo en línea]. [Fecha de consulta: 7 de abril de 2016]. <<http://www.ti.com/product/LMZ34002>>

^{xxxii} *LMR16006 Step-Down Regulator*. Texas Instruments [artículo en línea]. [Fecha de consulta: 7 de abril de 2016]. <<http://www.ti.com/product/LMR16006>>

^{xxxiii} *CadSoft EAGLE: Tutorials*. CadSoft [artículo en línea]. [Fecha de consulta: 19 de abril de 2016]. <<http://www.cadsoftusa.com/training-service/tutorials/>>

^{xxxiv} *CadSoft EAGLE: Manuals & Books*. CadSoft [artículo en línea]. [Fecha de consulta: 19 de abril de 2016]. <<http://www.cadsoftusa.com/training-service/manuals-books/>>