



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERÍA ELECTROMECÁNICA
ESPECIALIDAD ELECTRÓNICA

DISEÑO Y DESARROLLO DEL CONTROL DE UNA CALDERA DE GAS VAILLANT ECOTEC PLUS MEDIANTE PROTOCOLO EBUS

Autor: Hugo García Tovar
Director: Álvaro Sánchez Miralles
Director: Francisco Martín Martínez

Madrid
Julio 2018

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Hugo García Tovar DECLARA ser el titular de los derechos de propiedad intelectual de la obra: **DISEÑO Y DESARROLLO DEL CONTROL DE UNA CALDERA DE GAS ECOTEC PLUS MEDIANTE PROTOCOLO EBUS**, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

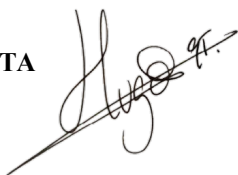
6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 12 de Julio de 2018

ACEPTA



Fdo. Hugo García Tovar

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
DISEÑO Y DESARROLLO DEL CONTROL DE UNA CALDERA DE GAS
VAILLANT ECOTEC PLUS MEDIANTE PROTOCOLO EBUS
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2017 / 2018 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.



Fdo.: Hugo García Tovar

Fecha: 14 / 07 / 2018

Autorizada la entrega del proyecto
LOS DIRECTORES DEL PROYECTO



Fdo.: Álvaro Sánchez Miralles

Fecha: 14 / 07 / 2018



Fdo.: Francisco Martín Martínez

Fecha: 14 / 07 / 2018



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERÍA ELECTROMECÁNICA
ESPECIALIDAD ELECTRÓNICA

DISEÑO Y DESARROLLO DEL CONTROL DE UNA CALDERA DE GAS VAILLANT ECOTEC PLUS MEDIANTE PROTOCOLO EBUS

Autor: Hugo García Tovar
Director: Álvaro Sánchez Miralles
Director: Francisco Martín Martínez

Madrid
Julio 2018

DISEÑO Y DESARROLLO DEL CONTROL DE UNA CALDERA DE GAS VAILLANT ECOTEC PLUS MEDIANTE PROTOCOLO EBUS

Autor: Hugo García Tovar

Director: Álvaro Sánchez Miralles

Director: Francisco Martín Martínez

Entidad colaboradora: ICAI – Universidad Pontificia Comillas

1. Introducción

Durante los últimos años, se está invirtiendo mucha ingeniería en el campo de la domótica. En el hogar, la gestión energética ha sido siempre una de las prioridades de las personas en cuanto al ahorro que ésta puede suponer.

Este proyecto aborda el objetivo de la construcción y programación de un dispositivo inteligente que tiene como fin el control y monitorización de una caldera de gas doméstica, en particular el modelo ecoTEC Plus de la marca Vaillant.

Para poder realizar el control de la caldera se utilizará una tarjeta de

circuito impreso (PCB) que contiene un microcontrolador. Éste se encarga de procesar los datos que recibirá desde un módulo Wifi para gestionar los datos de la caldera. En la Figura 1 se muestra un esquema de la configuración física del sistema, así como sus formas de comunicación entre los distintos componentes que lo forman.

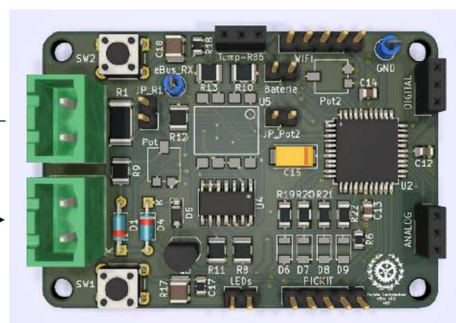
El sistema completo funcionaría de la siguiente forma. Desde una aplicación móvil, el usuario solicitaría información del estado de la caldera o le daría órdenes para configurar cualquiera de sus parámetros de funcionamiento. El dispositivo móvil utilizado tendría que estar conectado

Caldera Vaillant ecoTEC Plus



UART 1
eBus

Tarjeta Controladora



UART 2
Wifi



La Nube



Dispositivo móvil

Wifi



Módulo Wifi

Figura 1: Arquitectura del sistema

a un módulo de comunicación Wifi que está montado en la propia tarjeta. Los datos serían procesados por el módulo, que transmitiría por UART la solicitud al microcontrolador. El micro procesaría dicha solicitud y se encargaría de transmitir un telegrama en protocolo eBus a través de otra UART conectada a la caldera, la cual se encargaría entonces de procesar la solicitud y modificar sus parámetros en caso necesario.

A continuación, la caldera transmitiría un telegrama (con la información de la solicitud) de vuelta al micro y éste se encargaría de procesarlo y mandarlo por UART al módulo Wifi. El módulo lo mandaría inalámbricamente al dispositivo móvil, que actualizaría la información en la aplicación. Por último, el dispositivo móvil podría, además, hacer copias de seguridad de los datos que se quisiera en la nube para poder acceder desde otros dispositivos a esta información.

Aunque la arquitectura global del sistema contiene un módulo Wifi, una aplicación móvil y la nube, estos elementos no forman parte del desarrollo del proyecto. Éste se reduce únicamente al desarrollo de la tarjeta, la programación del protocolo de comunicaciones para realizar el intercambio de información con la caldera y el envío y recepción de información al módulo Wifi.

Desde el punto de vista de la tarjeta, todo lo que hay más allá del módulo Wifi se puede considerar como una caja negra. No es relevante lo que ocurre con la información una vez pasa hacia el módulo Wifi ni importa cómo llega desde éste a la tarjeta.

2. Metodología

Para el desarrollo del proyecto, se han seguido los siguientes pasos:

En primer lugar, se obtuvo toda la información posible sobre el protocolo de comunicaciones con el que funciona la caldera. eBus [1] (energy BUS) es un protocolo de comunicaciones que está actualmente orientado exclusivamente a aplicaciones de energía solar, calefacción y calderas de gas domésticas y es utilizado en mayor medida por marcas alemanas (Vaillant, Wolf, Hermann, etc.). Debido a su limitado uso, es un protocolo desconocido para la mayoría de los ingenieros.

Dado que eBus utiliza unos niveles de tensión anormales para representar la lógica binaria, en segundo lugar, se tuvo que diseñar un circuito que pudiera adaptar adecuadamente los niveles de tensión del micro a los del bus y viceversa.

Una vez se hubo construido y probado el adaptador, en tercer lugar, se diseñó la tarjeta controladora (Figura 2) mediante el software libre KiCAD.

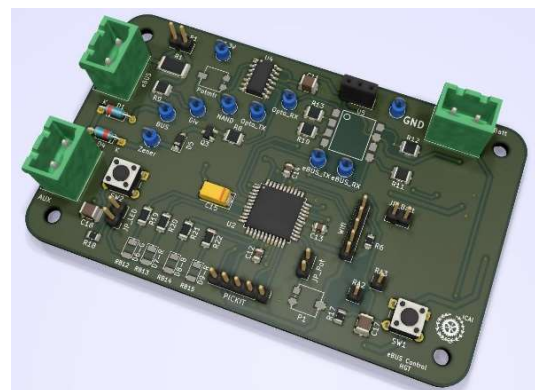


Figura 2: Tarjeta controladora. Primera versión

En cuarto lugar, se procedió a la comprobación de los módulos de la tarjeta y seguidamente a la programación del microcontrolador para poder llevar a cabo las comunicaciones con la caldera.

Por último, debido a que no ha sido posible el acceso físico a una caldera para poder hacer pruebas de comunicación, se han concentrado los últimos esfuerzos en rediseñar la tarjeta controladora (Figura 3), corrigiendo algunos errores de diseño no críticos que se hallaron durante las labores de comprobación de los distintos módulos.



Figura 3: Tarjeta controladora. Segunda versión

3. Resultados

Los resultados de las pruebas de programación fueron fructíferos, consiguiendo crear los telegramas adecuados siguiendo el estándar de eBus para comunicarse con la caldera. Por otro lado, las comprobaciones de los módulos de la tarjeta resultaron satisfactorias aunque se encontraron pequeños errores de diseño que fueron solventados en el diseño de la segunda versión. Este nuevo diseño es una tarjeta completamente funcional

en la que además se ha reducido el tamaño significativamente (más de un 50%) respecto de la primera versión. Además se hizo alguna mejora adicional, como dar acceso a 6 puertos extra del micro (3 analógicos y 3 digitales) para poder tener un acceso cómodo en caso querer conectar sensores adicionales u otros dispositivos.

4. Conclusiones

Como conclusión se puede considerar el desarrollo de este proyecto como exitoso porque:

1. Se ha conseguido diseñar una tarjeta totalmente funcional con todos sus módulos físicamente comprobados.
2. Se han creado y comprobado los algoritmos necesarios respetando el protocolo que potencialmente permiten la comunicación con una caldera.

5. Futuros desarrollos

Se considera que para finalizar el proyecto, principalmente se debe:

1. En primer lugar, poder tener acceso a una caldera real para poder realizar las pruebas necesarias y finalizar el desarrollo de los algoritmos.
2. En segundo lugar, hacer un tercer diseño de la tarjeta con una orientación comercial, eliminando los componentes prescindibles y optimizando, aún más si cabe, el tamaño de la tarjeta.

6. Referencias

- [1] Wikipedia, «EBUS (Serial Buses),» 2018. [En línea]. Available:
[https://en.wikipedia.org/wiki/EBUS_\(serial_buses\)](https://en.wikipedia.org/wiki/EBUS_(serial_buses)).

DESIGN AND DEVELOPMENT OF A VAILLANT ECOTEC PLUS BOILER CONTROL USING EBUS PROTOCOL

Author: Hugo García Tovar

Director: Álvaro Sánchez Miralles

Director: Francisco Martín Martínez

Collaborating Entity: ICAI – Universidad Pontificia Comillas

1. Introduction

Over the last few years, a lot of engineering has been invested in the field of Home Automation. Regarding savings, energy management has always been one of the biggest concerns of most people.

The aim of this project is the construction and programming of an intelligent device that will control and monitor a domestic gas boiler, the ecoTEC Plus model from Vaillant.

To control the boiler, a printed circuit board (PCB) that contains a microcontroller will be used. The

microcontroller processes the data that will receive from a Wi-fi module to manage the data of the boiler. Figure 1 shows a diagram of the physical configuration of the system, as well as its forms of communication between the different components that form it.

The whole system would work as it follows. From a mobile application, the user would request information from the state of the boiler or would give orders to configure any of its operating parameters. The mobile device used would have to be connected to the Wi-fi communication

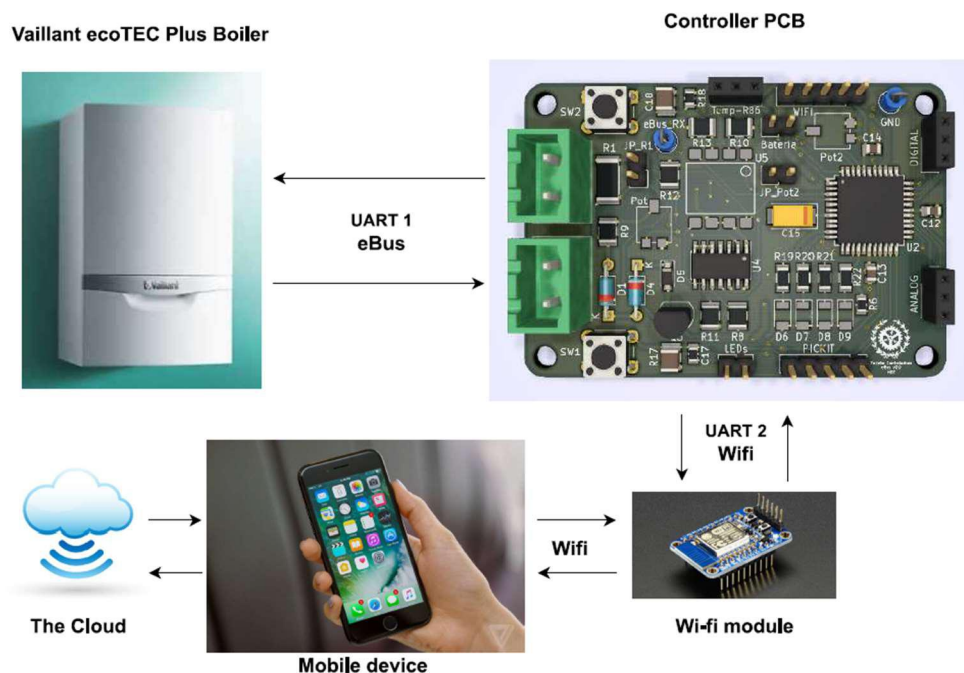


Figure 1: Architecture of the system

module that is mounted on the card itself. The data would be processed by the module, which would transmit by UART the request to the microcontroller. The microcontroller would then process this request and would be responsible for transmitting a telegram in eBus protocol through another UART connected to the boiler, which would process the request and modify its parameters if necessary. After it has processed the data, the boiler would transmit a telegram (with the information of the request) back to the microcontroller, which would process it and send it by UART to the Wi-fi module. The module would send it wirelessly to the mobile device, which would be responsible for updating the information in the application. Finally, the mobile device could also make backup copies of the data in the cloud to be able to access it from other devices.

Although the global architecture of the system contains a Wi-fi module, a mobile application and the cloud, these elements are not part of the development of the project; it is reduced only to the development of the board, the programming of the communications protocol to make the exchange of information with the boiler and the sending and reception of information to the Wi-fi module. From the PCB's point of view, everything that is beyond the Wi-fi module can be considered as a black box. It is not relevant what happens to the information once it passes to the Wi-fi module, nor how it comes from it to the board.

2. Methodology

For the development of the project, the following steps have been followed:

Firstly, it was gathered all the possible information regarding the communications protocol with which the boiler works. eBus [1] (Energy BUS) is a communications protocol that is currently used in solar energy applications, heating and domestic gas boilers and is used to a greater extent by German brands (Vaillant, Wolf, Hermann, etc.). Because of its limited use, it is an unknown protocol for the majority of the engineers.

Since eBus uses abnormal voltage levels to represent the binary logic, secondly, a circuit that could adequately adapt the voltage levels of the microcontroller to those of the bus and vice versa was designed.

Once the adapter was built and tested, in third place, the controller card was designed (Figure 2) using KiCAD freeware.

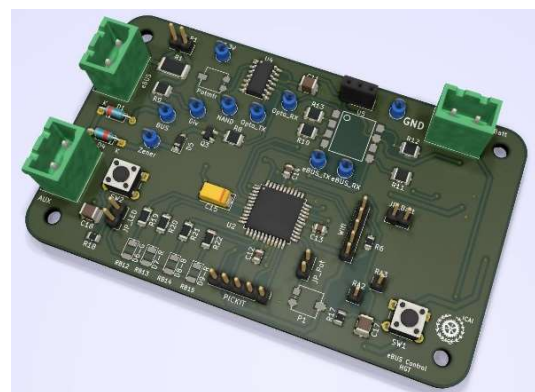


Figure 2: Controller PCB. First version

Fourthly, the board modules were checked and then the microcontroller was programmed to be able to carry out the communications with the boiler.

Finally, since it has not been possible to have physical access to a real boiler to be able to test communications, the latest efforts have been concentrated in redesigning the controller board (Figure 3), correcting some non-critical design errors that were found during the testing of the different modules.



Figure 3: Controller PCB. Second version

3. Results

On the one hand, the results of the programming tests were fruitful, creating the appropriate telegrams following the eBus standard to communicate with the boiler.

On the other hand, the modules' checks of the board were satisfactory although there were found small errors of design that were solved in the design of the second version. This new design is a fully functional board in which the size has also been significantly reduced (more than 50%) compared to the first version.

Furthermore, some additional improvements were made, such as giving access to 6 extra microcontroller ports (3 analogue and 3 digital) in order to have a comfortable access in case it is wanted in the future to connect additional sensors or other devices.

4. Conclusions

As a conclusion, the development of this project can be considered as successful because:

1. It was managed to build and design a fully functional controller board with all its modules physically checked.
2. The necessary algorithms have been created and tested following the protocol standards, which would potentially allow to establish communications with a boiler.

5. Future developments

It is considered that in order to finalize the project, at least two more tasks are yet to be done:

1. Firstly, to be able to have access to a real boiler to carry out the necessary tests and to finish the development of the algorithms.
2. Secondly, the design of a third controller board with a commercial orientation, removing the expendable components and optimizing, even more if possible, the size of the board.

6. References

- [1] Wikipedia, «EBUS (Serial Buses),» 2018. [Online]. Available:
[https://en.wikipedia.org/wiki/EBUS_\(serial_buses\)](https://en.wikipedia.org/wiki/EBUS_(serial_buses)).

Agradecimientos:

Quiero expresar mi agradecimiento a las siguientes personas, por cederme generosamente parte de su tiempo para ayudarme a despejar dudas y guiarme. El trabajo aquí desarrollado no habría sido posible sin su ayuda:

A mi director Álvaro Sánchez Miralles, por brindarme la oportunidad de trabajar en este proyecto y confiar en que sería capaz de sacarlo adelante.

A mi director Francisco Martín Martínez, por su dedicación, interés e implicación a lo largo de todo el desarrollo de este proyecto.

A los Doctores Jaime Boal y Carlos Rodríguez-Morcillo, por guiarme en las primeras fases del proyecto cuando más perdido me he encontrado.

A Antonio Vázquez Blanco, por su desinterés a la hora de resolverme dudas y por enseñarme los primeros pasos en el diseño de tarjetas de circuito impreso.

A mis padres,
por todo el
esfuerzo que
han invertido
para hacer de mí
una buena persona



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERÍA ELECTROMECÁNICA
ESPECIALIDAD ELECTRÓNICA

DISEÑO Y DESARROLLO DEL CONTROL DE UNA CALDERA DE GAS VAILLANT ECOTEC PLUS MEDIANTE PROTOCOLO EBUS

Autor: Hugo García Tovar
Director: Álvaro Sánchez Miralles
Director: Francisco Martín Martínez

Madrid
Julio 2018

Índice de la memoria

Parte I	Memoria.....	2
Capítulo 1	Introducción	3
1.1	Estado Del Arte.....	4
1.1.1	Breve historia de las calderas [1]	4
1.1.2	Tipos de calderas de gas [2]	5
1.1.2.1	Calderas ecoTEC Plus [3]	6
1.1.3	Arquitecturas de protocolos [4].....	6
1.2	Motivación del Proyecto.....	11
1.3	Objetivos.....	12
1.4	Metodología.....	12
1.5	Recursos y Herramientas.....	13
Capítulo 2	Arquitectura del Proyecto.....	14
2.1	Arquitectura Global	14
2.2	La Caldera ecoTEC Plus.....	15
2.3	La Tarjeta Controladora	16
Capítulo 3	Protocolo eBus.....	17
3.1	Introducción.....	17
3.2	Capa Física [8]	17
3.3	Capa de Enlace de Datos [8]	19
3.3.1	El Formato de Datos	19
3.3.2	Telegramas	20
3.3.2.1	Comunicación Maestro-Eslavo.....	20
3.3.2.2	Comunicación Maestro-Maestro	21

3.3.2.3 Comunicación broadcast	22
3.3.3 Componentes de los Telegramas.....	23
3.3.3.1 Símbolo de sincronización	23
3.3.3.2 CRC.....	23
3.3.3.3 ACK	23
3.3.3.4 Dirección fuente (QQ).....	24
3.3.3.5 Dirección destino (ZZ)	24
3.3.3.6 Comando primario (PB)	24
3.3.3.7 Comando secundario (SB).....	24
3.3.3.8 Número de bytes de datos (NN)	24
3.3.4 Arbitraje	25
3.3.4.1 El procedimiento de arbitraje orientado a byte.....	26
3.3.4.2 El acceso al bus tras el símbolo de sincronización	28
3.3.5 Generador Automático de Símbolos de Sincronización.....	29
3.3.6 Detección de Errores: CRC	30
3.3.6.1 Cálculo mediante división de polinomios [4].....	31
3.3.6.2 Cálculo mediante registros	32
3.3.7 Transmisión del Mensaje	33
3.4 Capa de Aplicación [9]	34
Capítulo 4 Adaptador eBus	35
4.1 Introducción.....	35
4.2 Diseño	36
4.3 Pruebas de Funcionamiento	38
4.3.1 Experimento I.....	40
4.3.2 Experimento II	42
4.3.3 Experimento III	43
Capítulo 5 Tarjeta Controladora.....	44
5.1 Introducción.....	44
5.2 Diseño	45
5.2.1 Microcontrolador	45
5.2.2 Adaptador eBus.....	47
5.2.3 Regulador.....	48
5.2.4 Sensor de Temperatura.....	49

5.2.5 Medidor de Batería.....	51
5.2.6 Conectores.....	52
5.2.6.1 eBus.....	52
5.2.6.2 Batería	52
5.2.6.3 Alimentación auxiliar	53
5.2.6.4 Sensor de temperatura externo	53
5.2.6.5 Módulo Wifi.....	53
5.2.6.6 PICKit	54
5.3 Fabricación.....	55
5.3.1 Diseño de Esquemático	55
5.3.2 Asignación de Huellas.....	56
5.3.3 Diseño de PCB.....	58
5.3.4 Archivos de Fabricación	60
5.4 Pruebas de Funcionamiento	61
5.4.1 Microcontrolador	61
5.4.2 Adaptador eBus.....	63
5.4.2.1 Problema con el transistor	63
5.4.3 Regulador.....	64
Capítulo 6 Algoritmo	66
6.1 Comunicación Maestro-Eslavo.....	66
6.2 Arbitraje.....	68
6.3 Generador de Símbolos de Sincronización.....	69
Capítulo 7 Resultados.....	70
7.1 Comunicación Maestro-Eslavo.....	70
7.2 Arbitraje.....	72
7.3 Generador de Símbolos de Sincronización.....	73
7.4 Sensores de Temperatura	75
7.4.1 Sensor Superficial	75
7.4.2 Sensor Externo	76
7.5 Fallos de Diseño Corregidos	77
7.5.1 Diseño Final de la Tarjeta Controladora	78

Capítulo 8	Conclusiones.....	80
Capítulo 9	Futuros Desarrollos.....	82
Capítulo 10	Bibliografía.....	84
Parte II	Estudio económico.....	85
Capítulo 1	Estudio económico.....	86
Parte III	Presupuesto.....	88
Capítulo 1	Mediciones	89
Capítulo 2	Precio Unitario	91
Capítulo 3	Presupuestos Parciales.....	93
Capítulo 4	Presupuesto General	95

Índice de ilustraciones

Ilustración 1: Funcionamiento básico de una caldera de gas	4
Ilustración 2: Modelo OSI frente al TCP/IP, por capas	8
Ilustración 3: Arquitectura del sistema.....	15
Ilustración 4: Arquitectura de UART 8-bit en eBus.....	19
Ilustración 5: Telegrama Maestro-Esclavo.....	21
Ilustración 6: Telegrama Maestro-Maestro	22
Ilustración 7: Telegrama Broadcast.....	22
Ilustración 8: Ejemplo de colisión.....	25
Ilustración 9: Arbitraje	27
Ilustración 10: Generador de símbolo de sincronización	30
Ilustración 11: Cálculo de CRC mediante división de polinomios	31
Ilustración 12: Cálculo de CRC mediante registros (estado inicial)	32
Ilustración 13: Cálculo de CRC mediante registros (estado final).....	33
Ilustración 14: Adaptador eBus	36
Ilustración 15: Adaptador eBus en Protoboard	39
Ilustración 16: Regulador de tensión auxiliar para pruebas	39
Ilustración 17: Transmisión I	40
Ilustración 18: Recepción I.....	41
Ilustración 19: Transmisión II	42
Ilustración 20: Tarjeta Controladora	44
Ilustración 21: Esquema conexionado del MIC	46
Ilustración 22: Regulador de tensión.....	48

Ilustración 23: Sensor de temperatura	50
Ilustración 24: Medidor de batería	51
Ilustración 25: Conectores de la PCB.....	52
Ilustración 26: PICKit3	54
Ilustración 27: Testpoint y agujeros de montaje	56
Ilustración 28: Huellas asignadas en KiCAD.....	57
Ilustración 29: Pads SMD junto a THT.....	57
Ilustración 30: Editor PCB. Estado inicial	58
Ilustración 31: Editor PCB. Estado final	59
Ilustración 32: Archivos de fabricación	60
Ilustración 33: PCB de fábrica	61
Ilustración 34: Prueba de transmisión UART	62
Ilustración 35: Prueba de transmisión adaptador	63
Ilustración 36: Transistor 2N3904 sustituyendo al MMBT3904	64
Ilustración 37: Regulación de tensión	65
Ilustración 38: Diagrama de flujo. Comunicación Maestro-Esclavo	67
Ilustración 39: Diagrama de flujo. Arbitraje	68
Ilustración 40: Diagrama de flujo. Generador de símbolos.....	69
Ilustración 41: Captura de un telegrama enviado en el osciloscopio	70
Ilustración 42: Tiempo de bit	71
Ilustración 43: Captura de un telegrama recibido en el osciloscopio.....	72
Ilustración 44: Símbolos de sincronización.....	74
Ilustración 45: Telegrama entre símbolos de sincronización	74
Ilustración 46: Capturas de temperatura por sensor superficial	75
Ilustración 47: Capturas de temperatura por sensor externo	76
Ilustración 48: Diseño renovado del regulador	77
Ilustración 49: Nueva versión de la tarjeta controladora.....	79

Índice de tablas

Tabla 1: Niveles de tensión válidos eBus.....	18
Tabla 2: Estudio de viabilidad.....	87
Tabla 3: Relación de equipos y herramientas.....	89
Tabla 4: Relación de uso de software.....	90
Tabla 5: Relación de horas de mano de obra directa.....	90
Tabla 6: Precio unitario de equipo y herramientas.....	91
Tabla 7: Precio unitario del software	92
Tabla 8: Coste unitario de mano de obra directa.....	92
Tabla 9: Coste total de equipo y herramientas	93
Tabla 10: Coste total de Software	94
Tabla 11: Coste total de mano de obra directa	94
Tabla 12: Presupuesto general.....	95

Parte I MEMORIA

Capítulo 1 INTRODUCCIÓN

Las calderas de gas forman parte de nuestra vida, aunque muchas veces no seamos conscientes de ello. Al igual que ocurre con Internet, el teléfono móvil o el agua, no nos damos cuenta de la importancia que tiene hasta que nos es despojado. Todos los habitantes del primer mundo tienen una caldera de gas en su hogar, que provee de agua caliente en grifos y duchas y un sistema de calefacción para las épocas frías. Con el crecimiento exponencial de la tecnología moderna y el surgimiento del internet de las cosas o la industria 4.0, se está rediseñando toda la electrónica de los dispositivos cotidianos. En particular, dentro de los hogares, esto incluye desde televisores inteligentes de última generación y frigoríficos que hacen la compra, hasta hornos, lavavajillas, lavadoras y las propias calderas de gas controladas y monitorizadas a distancia; todo ello para mejorar nuestro confort en el hogar. Además, siempre ha existido un interés por la gestión energética en los hogares, en cuanto al ahorro económico que este puede suponer.

El objeto de este proyecto es el diseño y desarrollo de un sistema electrónico que permita una comunicación bidireccional con una caldera de gas, en particular el modelo ecoTEC Plus de Vaillant. Las comunicaciones se llevarán a cabo mediante el protocolo eBus. Utilizando una tarjeta de circuito impreso (tarjeta controladora) que contiene un microcontrolador en que se programará el protocolo, se podrán enviar comandos a la caldera para controlar su funcionamiento y para recibir datos de ésta que se utilizarán para poder hacer una gestión de la energía mediante la optimización de sus funcionalidades. Para ello, se añadirá un módulo Wifi a la tarjeta controladora que permitirá recibir y enviar datos a un terminal móvil (Android o iOS). Los datos serán almacenados en la nube y mostrados en una aplicación móvil, desde la que se podrá realizar la monitorización y control de la caldera.

1.1 ESTADO DEL ARTE

A continuación se encuentran recopilados los conocimientos sobre calderas de gas y sus formas de comunicación.

1.1.1 BREVE HISTORIA DE LAS CALDERAS [1]

La historia de las primeras calderas está ligada a la Revolución Industrial de finales del siglo XIX con la aparición de la máquina de vapor. Aunque se empezaron usando en las industrias y trenes para el transporte de mercancías y personas, no tardaron en llegar a los hogares para proveer de calefacción a estos.

Se instalaron calderas de vapor y de carbón hasta la aparición de la caldera de gas, que supuso un cambio radical para las ciudades y el entorno rural.

El funcionamiento de estas calderas es relativamente simple. Como se aprecia en la Ilustración 1, el agua fría entra desde la red de abastecimiento y se calienta a unos 90°C mediante un quemador que se controla electrónicamente. En un intercambiador de calor se calienta el agua que sale por los grifos.

Con el paso del tiempo estas calderas se han ido sofisticando, mejorando su rendimiento y controlando sus emisiones al medio ambiente.

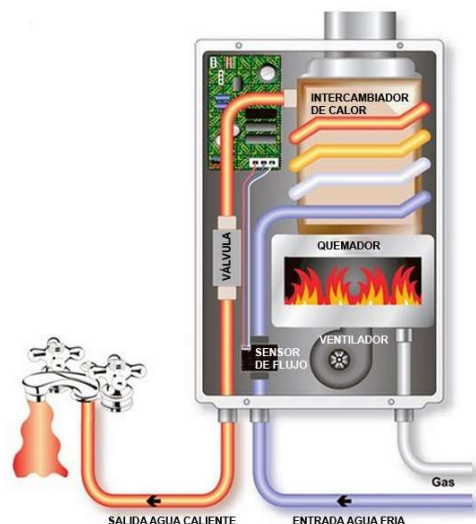


Ilustración 1: Funcionamiento básico de una caldera de gas

1.1.2 TIPOS DE CALDERAS DE GAS [2]

Las calderas pueden clasificarse en función del tipo de combustible utilizado, del tipo de producción, de la cámara de combustión y la evacuación de gases y de las emisiones al ambiente y su rendimiento.

Según el combustible:

- Gas: combustible gaseoso, pudiendo ser gas natural o propano.
- Gasóleo: combustible líquido.
- Biomasa: combustible sólido, pudiendo ser pellets de madera, huesos de aceituna, cáscaras de frutos secos, residuos forestales, etc.

Según los tipos de producción:

- Sólo calefacción: provee al hogar con calefacción, aunque se puede instalar un acumulador a parte para suministrar agua caliente sanitaria (ACS).
- Mixta instantánea: gracias a un micro acumulador, puede suministrar calefacción y ACS de forma instantánea.
- Mixta con acumulación: dan servicio a calefacción a ACS mediante un acumulador. Además, se precisa de otro acumulador de agua fría para alimentar el acumulador de agua caliente y para equilibrar el nivel de agua del sistema.

Según la cámara de combustión y la evacuación de los gases:

- Atmosféricas: la cámara es abierta, cogiendo el oxígeno para la combustión de la misma habitación. Los gases son evacuados por tiro natural. Fueron las primeras en aparecer, con el gran inconveniente de desprender monóxido de carbono en la habitación.
- Tiro forzado: la cámara es abierta y los gases son extraídos de la cámara mediante un ventilador.
- Presurizadas: la cámara es abierta y los gases son extraídos introduciendo aire en la cámara mediante un ventilador.
- Estancas: la cámara está herméticamente cerrada y la evacuación de los gases es forzada. Son más seguras.
- Estancas de bajo NOx: funcionan de forma similar a los estancos, pero consumen menos combustible y los humos son menos contaminantes.
- Estancas de condensación: son las que mejor rendimiento obtienen, al quemar el gas a una temperatura menor y por ello con mayor aprovechamiento energético del calor. Expulsan humos de bajo NOx.

Según las emisiones al ambiente y el rendimiento:

- Gas: donde encontramos calderas del tipo atmosféricas clase 3 NO_x, tiro forzado clase 3 NO_x, estanca clase 4 NO_x, 5 NO_x, estanca de bajo NO_x y estancas de condensación clase 5 NO_x.
- Gasóleo: donde se incluyen las calderas presurizadas y estancas de condensación.
- Biomasa: se les atribuye emisiones cero.

Por su eficiencia energética superior al 100% y su menor impacto ecológico, desde la aprobación de una Directiva comunitaria en septiembre del año 2015, las calderas de condensación son las únicas que se pueden fabricar e instalar y, en la actualidad, ya no se instalan las calderas atmosféricas ni las estancas, aunque si se realiza mantenimiento de las mismas.

1.1.2.1 Calderas ecoTEC Plus [3]

Las calderas de ecoTEC Plus se clasifican según los tipos de producción explicados anteriormente. Están preparadas para conjuntarlas con paneles solares o bombas de calor. Tienen un amplio rango de potencias y otras características. Son simples de usar debido al display con el que vienen equipadas, donde se muestran gran cantidad de datos sobre el estado de su funcionamiento. El sistema de comunicaciones electrónico que utilizan estas calderas se denomina eBUS. Este protocolo es compartido con otras marcas de calderas, normalmente de origen alemán.

1.1.3 ARQUITECTURAS DE PROTOCOLOS [4]

A la hora de intercambiar información entre dos dispositivos, intervienen varios procesos complejos: desde poner los datos de información en el formato predeterminado, hasta establecer el camino de comunicación, dentro de la red de comunicaciones, pasando por indicar el destinatario de la información, entre otros.

En lugar de implementar todo en un único módulo o proceso software, el problema se divide en tareas, submódulos o procesos lógicos, cada uno de los cuales se realiza independientemente el uno del otro.

Cada submódulo se representa como una capa o nivel. El conjunto de capas forma una pila vertical y en la parte superior se encuentra el nivel de aplicación, es decir, donde se utiliza la información recibida de la comunicación o donde se genera la información que se va a transmitir para que el receptor la utilice. En la capa inferior se encuentra la capa de acceso al medio físico, es decir, quien genera la energía para transmitir señales por el canal de comunicaciones (en el caso del transmisor) y quien interpreta la energía recibida por el canal de comunicaciones (en el caso del receptor). Entre medias de estas dos capas se pueden encontrar una o varias capas más que, en conjunto, se encargan de gestionar el transporte de la información desde un punto de vista lógico.

Con el desarrollo tecnológico de las comunicaciones y de las redes, el que cada fabricante desarrolle su propia arquitectura de protocolos no tiene sentido desde un punto de vista práctico, aparte de ser muy costoso. La solución a este problema, tanto para los fabricantes como para los consumidores de sus equipos, es adoptar e implementar un conjunto de convenciones comunes y plasmarlos en una norma técnica o estándar.

Existen dos estándares o modelos de referencia reconocidos: OSI y TCP/IP.

En 1984, la ISO (International Organization for Standardization) publicó la ISO 7498, que define las capas o niveles que deberían formar una arquitectura de protocolos para la comunicación entre dispositivos. A esta estructura se la conoce como modelo de referencia OSI.

La labor de ISO consistió en definir el conjunto de capas, así como los servicios a realizar por cada una de ellas. La división debía agrupar a las funciones que fueran conceptualmente próximas en un número suficiente, tal que cada capa fuese lo suficientemente pequeña, pero sin llegar a definir demasiadas, para evitar así sobrecargas en el procesamiento. El modelo de referencia resultante tiene siete capas: física, enlace de datos, red, transporte, sesión, presentación y aplicación.

Por otro lado, la arquitectura de protocolos TCP/IP es resultado de la investigación y desarrollo llevados a cabo en la red experimental ARPANET, financiada por la Agencia de Proyectos de Investigación Avanzada para la Defensa (Defense Advanced Research Projects Agency (DARPA)), y se denomina globalmente como la familia de protocolos TCP/IP. Esta familia consiste en una extensa colección de protocolos que se han especificado como estándares de Internet por parte de IAB (Internet Architecture Board).

El modelo TCP/IP estructura la comunicación entre equipos/dispositivos en cinco capas relativamente independientes entre sí: física, acceso a la red, internet, transporte y aplicación.

En la Ilustración 2, se muestra una comparativa por capas entre los dos modelos estandarizados.

OSI	TCP/IP
Aplicación	Aplicación
Presentación	
Sesión	
Transporte	Transporte (origen-destino)
Red	Internet
Enlace de datos	Acceso a la red
Física	Física

Ilustración 2: Modelo OSI frente al TCP/IP, por capas

A continuación, se explica las características y funcionalidades de cada una de las capas del modelo OSI:

Capa física:

La capa física se encarga de la interfaz física entre los dispositivos y define las reglas que rigen en la transmisión de los bits. La capa física tiene cuatro características importantes:

- Mecánicas: relacionadas con las propiedades físicas de la interfaz del equipo/dispositivo con el medio de transmisión. Normalmente, dentro de estas características se incluye la especificación del conector que transmite las señales a través de los conductores.
- Eléctricas: especifican cómo se representan los bits (por ejemplo, en términos de niveles de tensión), así como su velocidad de transmisión.
- Funcionales: especifican las funciones que realiza cada uno de los conductores de la interfaz física entre el sistema y el medio de transmisión.
- De procedimiento: especifican la secuencia de eventos que se llevan a cabo en el intercambio del flujo de bits a través del medio físico.

Capa de enlace de datos:

Mientras que la capa física proporciona exclusivamente un servicio de transmisión de datos, la capa de enlace de datos intenta hacer que el enlace físico sea fiable. Además, proporciona los medios para activar, mantener y desactivar el enlace. El principal servicio proporcionado por la capa de enlace de datos a las capas superiores es el de detección y control de errores. Así, si se dispone de un protocolo en la capa de enlace de datos completamente operativo, la capa adyacente superior puede suponer que la transmisión está libre de errores. Sin embargo, si la comunicación se realiza entre dos sistemas que no estén directamente conectados, la conexión constará de varios enlaces de datos en serie, cada uno operando independientemente. Por tanto, en este último caso, la capa superior no estará libre de la responsabilidad del control de errores.

En esta capa se procesa la información de gestión de errores (paridad, checksum, CRC) añadida a los datos de información originales.

Capa de red:

La capa de red libera a las capas superiores de la necesidad de tener conocimiento sobre la transmisión de datos, y las tecnologías y medios de transmisión utilizados para conectar los sistemas. En esta capa, el dispositivo establece un diálogo con la red para especificar la dirección destino y solicitar ciertos servicios, como, por ejemplo, la gestión de prioridades.

Capa de transporte:

Esta capa y las tres capas superiores proporcionan un mecanismo para intercambiar datos entre sistemas finales (extremo a extremo). El servicio de transporte asegura que los datos se entregan libres de errores, en orden y sin pérdidas ni repeticiones. También puede estar involucrada en la optimización del uso de los servicios de red, y en proporcionar la calidad del servicio solicitada. Por ejemplo, el nivel superior (capa de sesión) puede solicitar una tasa máxima de error determinada, un retardo máximo, una prioridad o un nivel de seguridad dado.

El tamaño y la complejidad de un protocolo de transporte dependen de cómo de fiables sean los servicios de las capas inferiores. Consecuentemente, ISO ha desarrollado una familia de cinco protocolos de transporte normalizados, cada uno de ellos especificado para un determinado servicio.

Capa de sesión:

La capa de sesión proporciona los mecanismos para controlar el diálogo entre las aplicaciones de los sistemas finales. En muchos casos, los servicios de la capa de sesión son parcialmente, o incluso totalmente, prescindibles. No obstante, en algunas aplicaciones su utilización es ineludible. La capa de sesión proporciona los siguientes servicios:

- Control del diálogo: éste puede ser simultáneo en los dos sentidos (full-duplex) o alternado en ambos sentidos (half-duplex).
- Agrupamiento: el flujo de datos se puede marcar para definir grupos de datos.

- Recuperación: la capa de sesión puede proporcionar un procedimiento de puntos de comprobación, de forma que, si ocurre algún tipo de fallo entre puntos de comprobación, la entidad de sesión puede retransmitir todos los datos desde el último punto de comprobación.

Capa de presentación:

La capa de aplicación proporciona a los programas software un medio para que accedan a la pila OSI. A esta capa pertenecen las funciones de administración y gestión de las aplicaciones que se ejecutan en el equipo. Además, en esta capa pueden encontrarse varias aplicaciones funcionando simultáneamente.

Capa de aplicación:

La capa de aplicación proporciona a los programas software un medio para que accedan a la pila OSI. A esta capa pertenece las funciones de administración y gestión de las aplicaciones que se ejecutan en el equipo. Además, en esta capa pueden encontrarse varias aplicaciones funcionando simultáneamente.

1.2 MOTIVACIÓN DEL PROYECTO

Este proyecto forma parte de uno de mayor magnitud cuyo objetivo es el control y la gestión energética de un hogar. Este tipo de proyectos están orientados a mejorar la calidad de vida de los usuarios. Las viviendas son un espacio donde pasamos gran parte de nuestro tiempo, de ahí que muchos proyectos hayan nacido orientados a mejorar nuestro confort en el hogar.

Es realmente interesante trabajar en un gran proyecto en el que se tiene una intención real de aplicar el trabajo realizado a la vida de usuarios reales en el futuro inmediato y no como un mero ejercicio de laboratorio.

1.3 OBJETIVOS

Los objetivos principales de este proyecto son los siguientes:

1. Desarrollo de un dispositivo capaz de realizar comunicaciones con una caldera mediante el protocolo eBus.
2. Integración y pruebas de funcionamiento en un entorno real.

1.4 METODOLOGÍA

Las tareas que realizar serán las siguientes:

1. Búsqueda o diseño de un chip adaptador de tensión apropiado a eBus.
2. Diseño de una PCB con el programa de software KiCAD. Es una placa de pruebas, con un microcontrolador, pulsadores, diodos LED, test-point, conectores con diferentes funciones (programador, alimentación, bus de comunicaciones...) y un circuito adaptador de tensiones para poder llevar a cabo una comunicación caldera-micro.
3. Fabricación de la PCB.
4. Recabar información sobre las variables, comandos y direcciones de memoria de las calderas ecoTEC Plus de Vaillant.
5. Programación del micro con el software MPLAB. Se programará el protocolo desde cero y se implementará por UART.
6. Pruebas con una caldera real y depuración de hardware y software.
 - a. Comunicación con la caldera.
 - b. Funcionamiento del modo manual.
7. Integración del sistema con la nube.

1.5 RECURSOS Y HERRAMIENTAS

Para el adecuado desarrollo del proyecto se empleará:

- Hardware:
 - Microcontrolador.
 - Circuito regulador de tensión.
 - Módulo inalámbrico de comunicaciones.
 - Tarjeta de circuito impreso (PCB).
- Software:
 - Software de programación MPLAB.
 - Software de diseño KiCAD.

Capítulo 2 ARQUITECTURA DEL PROYECTO

En este capítulo se explica la organización del proyecto planteado, así como el contexto en el que se encuentra.

2.1 ARQUITECTURA GLOBAL

El proyecto aquí definido tiene como objetivo la construcción y programación de un dispositivo inteligente que tiene como fin el control y monitorización de una caldera de gas doméstica de la marca Vaillant. La gestión eficiente de la temperatura como parte de un sistema domótico más amplio como el presentado en otros proyectos [5] [6], tiene su importancia debido al ahorro energético que conlleva. Por lo tanto, se debe considerar que este proyecto está enmarcado en el desarrollo de otro proyecto de mayor magnitud y, sin embargo, independiente de éste.

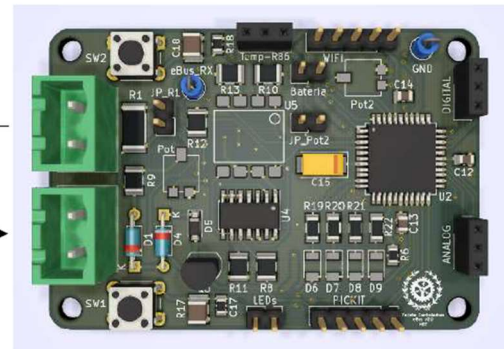
Un esquema de la arquitectura de este proyecto se muestra en la Ilustración 3. En ella, se pueden ver las relaciones que tienen entre sí los distintos dispositivos. La tarjeta de la izquierda podría considerarse como el elemento central del sistema, ya que actúa como el dispositivo que conecta el objeto a controlar con el controlador. Aunque en este esquema se incluyen un módulo Wifi, un dispositivo móvil (con su correspondiente app para gestionar la caldera) y la nube, estos no forman parte del proyecto que se ha desarrollado, estando éste únicamente limitado al desarrollo y programación de la tarjeta, las comunicaciones con la caldera y el envío y recepción de información al módulo Wifi.

Dicho eso, se podría pues considerar, todo lo que haya más allá del módulo Wifi como una caja negra. No es relevante lo que ocurre con la información una vez pasa hacia el módulo Wifi ni importa cómo llega desde éste a la tarjeta.

Caldera Vaillant ecoTEC Plus



Tarjeta Controladora



UART 1
eBus



La Nube



Dispositivo móvil
(Android o iOS)
(app móvil específica)

UART 2
Wifi



Módulo Wifi

Ilustración 3: Arquitectura del sistema

2.2 LA CALDERA ECOTEC PLUS

El dispositivo que se quiere controlar es la caldera ecoTEC Plus de Vaillant. Ésta será el esclavo en las comunicaciones con la controladora, es decir, por sí sola no hace nada más que funcionar como una caldera estándar y tendrá que ser controlada y monitorizada por el maestro para dar información y/o cambiar de estado.

En realidad, se podrían utilizar otras calderas de Vaillant (o incluso de otras marcas como Hermann o Wolf) para este proyecto; la condición suficiente es que las comunicaciones se realicen con el protocolo eBus. El resto se reduciría a una cuestión de programación del microcontrolador, en función de los comandos que utilice cada caldera.

También cabe destacar, que Vaillant ofrece reguladores y controladores que hacen las veces de maestro, con la capacidad de controlar y monitorizar la caldera incluso a distancia.

2.3 LA TARJETA CONTROLADORA

La tarjeta controladora es una tarjeta de circuito impreso (PCB) que se ha diseñado para controlar la caldera, por lo tanto, actúa como su maestro. Las partes más importantes de la PCB son el microcontrolador y un circuito adaptador para poder elevar y reducir los niveles de tensión y permitir la comunicación entre la caldera y el MIC a través del bus. Dicho adaptador no es un circuito comercial, sino que se ha diseñado y probado previamente a su implementación.

La controladora administra las comunicaciones entre la caldera y el módulo Wifi, situándose como el dispositivo central del proyecto. Si el usuario desea, por ejemplo, conocer la temperatura del agua de la caldera, deberá comunicarlo mediante la app del dispositivo móvil al módulo Wifi, el cual lo transmitirá inmediatamente a una de las UART del micro (UART 2 según la Ilustración 3). El MIC interpretará la información y preparará un telegrama eBus que mandará a la caldera a través de otra UART (UART 1). La caldera responderá con otro telegrama eBus que será captado por la UART 1, procesado por el MIC y éste se lo comunicará al módulo Wifi por la UART 2, quien lo transmitirá finalmente al dispositivo móvil. Desde la app se podrá hacer copias de seguridad de los datos recibidos en la nube, que podrán ser descargados por otros dispositivos para utilizarlos si se necesita en otros sistemas del proyecto global ya mencionado.

Por todo ello puede considerarse que la PCB es el elemento central de este proyecto, pues actúa como nodo central entre el origen (dispositivo móvil) y el elemento gestionado (caldera).

Capítulo 3 PROTOCOLO EBUS

3.1 INTRODUCCIÓN

eBUS [7] (energy BUS) es un protocolo de comunicaciones creado por la compañía Karl Dungs, la cual trabaja en procesos térmicos industriales. eBus está actualmente orientado exclusivamente a aplicaciones de energía solar, calefacción y calderas de gas domésticas y es utilizado en mayor medida por marcas alemanas (Vaillant, Wolf, Hermann, Saunier Duval...). Debido a su limitado uso, es un protocolo desconocido para la mayoría de los ingenieros. En este capítulo se pretende sintetizar la mayor cantidad de información sobre su funcionamiento.

eBus sigue el modelo de referencia OSI, si bien no consta de las siete capas que componen el modelo. En su caso particular, contiene capa física, capa de enlace de datos y capa de aplicación.

3.2 CAPA FÍSICA [8]

eBus es un protocolo de comunicación serie, activo a nivel bajo, de doble hilo y con capacidad de multimaestro. Generalmente opera en modo asíncrono, aunque se puede operar de forma síncrona para ciertas aplicaciones que no atañen al caso de estudio, por lo cual no se entrará en más detalle en este capítulo.

eBus tiene las siguientes características:

- Sólo utiliza dos cables, con los que se consiguen alimentación y comunicación simultáneamente.
- Velocidad de modulación fija a 2400 baudios, con una tolerancia de $\pm 1.2\%$.

- Bus activo a nivel bajo con niveles de tensión como se muestran en la Tabla 1. Estos niveles de tensión son anormales y bastante diferentes a los de otros protocolos de comunicaciones.

Nivel lógico	Transmisor	Receptor
Alto = '1'	> 15 VDC	> 15 VDC y < 24 VDC
Bajo = '0'	≥ 8 VDC y ≤ 10 VDC	≥ 8 VDC y ≤ 12 VDC

Tabla 1: Niveles de tensión válidos eBus

- Capacidad de multimaestro de hasta 25 maestros.
- Capacidad de 228 esclavos.
- Tipos de comunicaciones posibles:
 - Maestro – Maestro
 - Maestro – Esclavo
 - Broadcast (transmisión dirigida a todos los equipos del bus).
- Integración mediante UART (8 bits con 1 bit de inicio y 1 bit de stop; no lleva bit de paridad).
- CRC de 8 bits para la detección de errores en la comunicación.
- Datos útiles de longitud variable entre 0 bytes y 16 bytes.
- 254 comandos primarios y 254 secundarios, aunque podrían existir hasta 64516 comandos.
- Varias opciones de alimentación al bus.

3.3 *CAPA DE ENLACE DE DATOS [8]*

En esta capa se procesa la información de gestión de errores (CRC), añadida a los datos de información originales. También se realiza el proceso de arbitraje y de la sincronización del bus como se detalla en los apartados siguientes.

3.3.1 EL FORMATO DE DATOS

eBus utiliza el formato de 8 bits con UART, que funciona de la siguiente forma: por cada 8 bits de información útil se envía en serie un bit de start (0 lógico), seguido de los 8 bits útiles y finalizando con un bit de stop (1 lógico), sin añadir bit de paridad, tal como muestra la Ilustración 4.

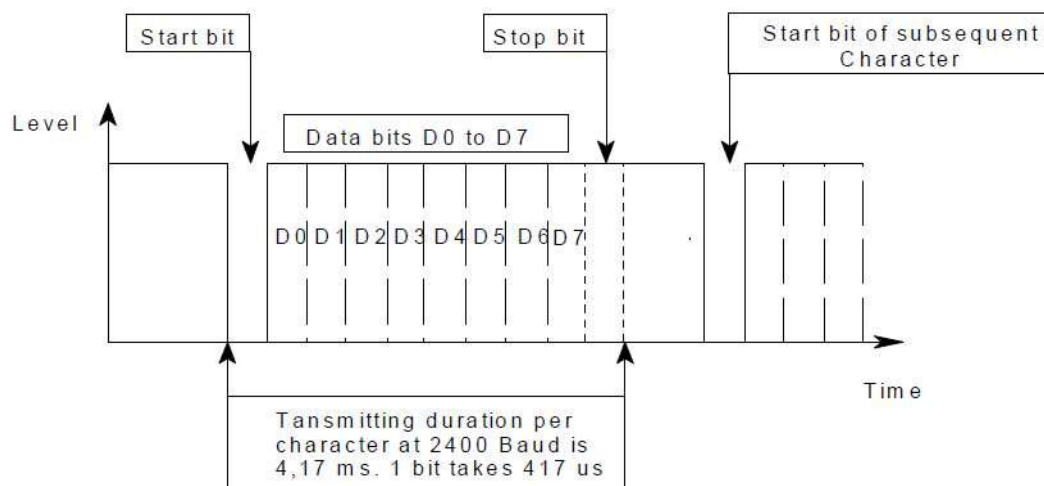


Ilustración 4: Arquitectura de UART 8-bit en eBus

(Fuente: eBus Interest Group)

Como las tramas completas son de varios bytes, se repite esta estructura hasta haber enviado la trama completa y añadiendo un CRC de toda la trama (únicamente de los bits útiles) al final de ésta.

3.3.2 TELEGRAMAS

Los telegramas son secuencias de caracteres individuales que el participante debe enviar para poder acceder al bus. Para que pueda haber una comunicación efectiva, la estructura de los telegramas tiene que seguir un estándar.

Aunque los participantes pueden ser maestros y esclavos, los maestros son los únicos que pueden iniciar las comunicaciones. Los esclavos no acceden al bus a no ser que un maestro les obligue a responder. Las comunicaciones se pueden efectuar entre maestros, entre maestro y esclavo, o en modo broadcast, es decir, dirigido a todos los participantes del bus.

3.3.2.1 Comunicación Maestro-Esclavo

Es la más común, el maestro pregunta al esclavo por información y este responde a continuación. La estructura de la comunicación se puede ver en la Ilustración 5 y se describe como sigue:

Maestro:

- QQ: 1 byte con la dirección del maestro.
- ZZ: 1 byte con la dirección del esclavo.
- PB: 1 byte con el código primario del comando.
- SB: 1 byte con el código secundario del comando.
- NN₁: 1 byte con el indicador de la longitud de los datos que transmite a continuación.
- DB_M: Entre 0 y 16 bytes de datos.
- CRC: 1 byte con el CRC.

Esclavo:

- ACK: 1 byte con el acuse (acknowledgement) del esclavo.
- NN₂: 1 byte con el indicador de la longitud de los datos que transmite a continuación.
- DB_S: Entre 0 y 16 bytes de datos.
- CRC: 1 byte con el CRC.

Maestro:

- ACK: 1 byte de acuse del maestro.
- SYN: 1 byte con el símbolo de sincronización.

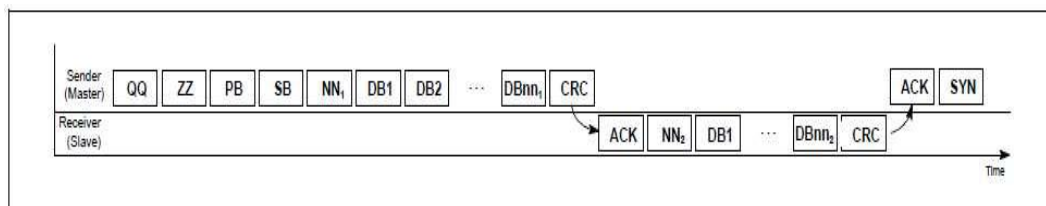


Ilustración 5: Telegrama Maestro-Esclavo

(Fuente: eBus Interest Group)

3.3.2.2 Comunicación Maestro-Maestro

En algún caso puede interesar mandar información de un maestro a otro. Los maestros pueden acceder al bus siempre que quieran, aunque deben cumplir ciertas condiciones como se verá en el apartado sobre arbitraje. Todos los maestros tienen una dirección de esclavo, para que un maestro pueda dirigirse a otro maestro. La dirección de esclavo de un maestro se halla sumando 5 a la dirección de maestro del maestro. Por ejemplo, si un maestro tiene dirección 0x03, su dirección de esclavo será la 0x08. En el caso de ser de las últimas direcciones, por ejemplo, 0xFF, su dirección de esclavo será la 0x04. La estructura de la comunicación Maestro-Maestro es la siguiente (ver Ilustración 6):

Maestro 1:

- QQ: 1 byte con la dirección del maestro.
- ZZ: 1 byte con la dirección del segundo maestro.
- PB: 1 byte con el código primario del comando.
- SB: 1 byte con el código secundario del comando.
- NN: 1 byte con el indicador de la longitud de los datos que transmite a continuación.
- DB_M: Entre 0 y 16 bytes de datos.
- CRC: 1 byte con el CRC.

Maestro 2:

- ACK: 1 byte con el acuse del segundo maestro.

Maestro 1:

- SYN: 1 byte con el símbolo de sincronización.

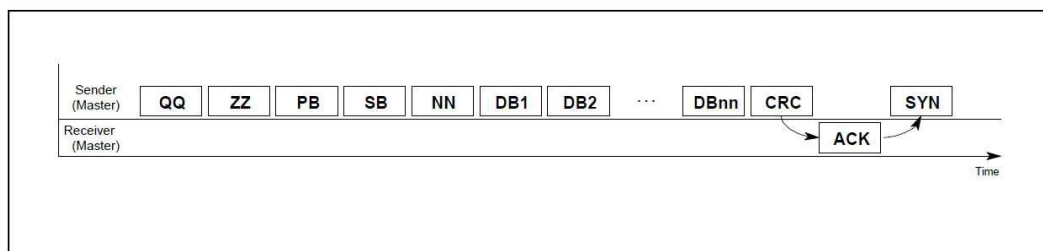


Ilustración 6: Telegrama Maestro-Maestro

(Fuente: eBus Interest Group)

3.3.2.3 Comunicación broadcast

El último modo de comunicación es el modo broadcast. Esto significa que el mensaje va dirigido a todos los participantes que estén conectados al bus. Además, normalmente ningún participante responde a este mensaje, ni siquiera con el acuse. La estructura del telegrama es la siguiente (ver Ilustración 7):

Maestro:

- QQ: 1 byte con la dirección del maestro.
- ZZ: 1 byte con la dirección de destino, la genérica para broadcast (0xFF).
- PB: 1 byte con el código primario del comando.
- SB: 1 byte con el código secundario del comando.
- NN: 1 byte con el indicador de la longitud de los datos que transmite a continuación.
- DB_M: Entre 0 y 16 bytes de datos.
- CRC: 1 byte con el CRC.
- SYN: 1 byte con el símbolo de sincronización.

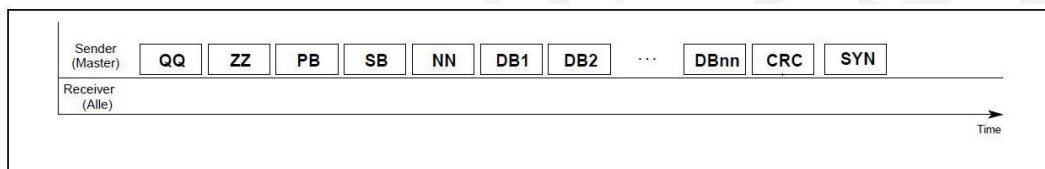


Ilustración 7: Telegrama Broadcast

(Fuente: eBus Interest Group)

3.3.3 COMPONENTES DE LOS TELEGRAMAS

Como ya se ha visto en el apartado anterior, existen distintos componentes que forman los telegramas y que se detallan a continuación.

3.3.3.1 *Símbolo de sincronización*

La condición para que los maestros puedan comenzar las comunicaciones es que deben haber recibido previamente el símbolo de sincronización. En cuanto esto ocurre, se entiende que el bus está libre y los maestros pueden enviar sus telegramas. Si hay varios que empiezan la comunicación al mismo tiempo, comienza un proceso de arbitraje, cuyo funcionamiento se detalla en el siguiente apartado.

El símbolo de sincronización se compone de la secuencia 10101010 (0xAA). Este símbolo está reservado y no debe ser usado en las comunicaciones. Por ello, si en la trama de datos de una comunicación que se está enviando aparece el byte 0xAA, éste debe ser convertido en dos bytes de la forma 0xA9 + 0x01. Si el byte 0xA9 aparece en la trama de datos, debe ser igualmente convertido en dos bytes de la forma 0xA9 + 0x00. De esta forma, se asegura que el bus no quede libre por error.

3.3.3.2 *CRC*

El código de redundancia cíclica se utiliza para detectar errores en las comunicaciones. En eBus el polinomio generador utilizado es: $x^8 + x^7 + x^4 + x^3 + x + 1$. Este polinomio es diferente al utilizado en el resto de los protocolos (por ejemplo, Modbus). Es decir, el CRC de eBus es único. El método para su cálculo se explica en el apartado 3.3.6.

3.3.3.3 *ACK*

El acuse o acknowledgement es un elemento que establece si la comunicación se ha realizado correctamente o no. Cuando se recibe un mensaje de, por ejemplo, un maestro, el esclavo correspondiente calcula el CRC del mensaje. A continuación, compara el CRC calculado con el que venía en la trama del maestro, y, si son

iguales, da el valor de 0x00 al ACK. En caso de no coincidir, devuelve el valor de 0xFF, indicando que el mensaje no ha sido recibido correctamente.

3.3.3.4 Dirección fuente (QQ)

También llamado source address, es la dirección del maestro que envía el telegrama. Esta dirección se utiliza en el arbitraje, como se verá en el próximo apartado. En total, puede haber un máximo de 25 direcciones fuente.

3.3.3.5 Dirección destino (ZZ)

Es la dirección del destinatario del telegrama. Existen 254 direcciones de destino.

3.3.3.6 Comando primario (PB)

Con el comando primario se clasifica el mensaje que viene detrás. Debe ser conocido por el receptor, o sino no podría saber qué hacer con el telegrama recibido. Este comando puede ser definido libremente entre los participantes del bus y tener unas instrucciones definidas con un significado común para todos los participantes. Hay un máximo de 254 comandos primarios dado que no se pueden usar el 0xAA ni el 0xA9.

3.3.3.7 Comando secundario (SB)

Este comando añade más información sobre el comando primario y con ello otorga un total de 254*254 comandos en total (excluyendo 0xAA y 0xA9).

3.3.3.8 Número de bytes de datos (NN)

Este número indica cuantos bytes de información útil vienen a continuación. Este número puede estar entre 0 y 16 incluidos. Además, cabe destacar, que, aunque se sustituyan en la trama los caracteres de 0xAA o 0xA9, extendiendo la trama a 1 byte extra, NN será invariable.

3.3.4 ARBITRAJE

Si varios maestros acceden al bus después de recibir el símbolo de sincronización, se aplica el procedimiento de arbitraje para evitar un conflicto en el bus. El criterio de decisión para el arbitraje es la propia dirección que tienen los maestros. El maestro con prioridad al bus será el que tenga la dirección más baja, como se explicará a continuación. El proceso de arbitraje es diferente al de otros protocolos como I2C por ser comprobados bit a bit. En eBus sólo se puede ir comprobando byte a byte. El problema de esto se comprueba en la Ilustración 8:

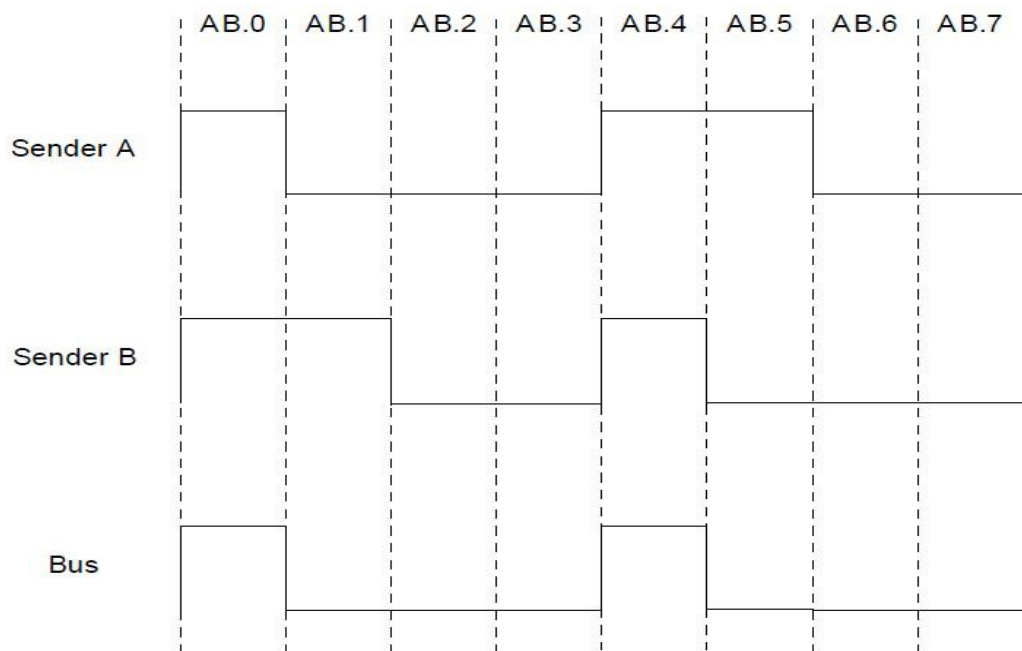


Ilustración 8: Ejemplo de colisión

(Fuente: eBus Interest Group)

Cuando los maestros A y B mandan su telegrama se produce una colisión. Esto resulta en que cada uno de los maestros recibe en su buffer de recepción lo que se ha generado en el Bus (que sigue una lógica AND, siendo la entrada todos los participantes y la salida el estado del bus), pero ninguno de los maestros se va a reconocer. En este caso particular, ambos maestros volverían a esperar al símbolo

de sincronización para volver a transmitir lo mismo una y otra vez y de esta forma el bus quedaría totalmente paralizado. Mediante el uso de un arbitraje orientado a byte se resuelve este problema.

3.3.4.1 El procedimiento de arbitraje orientado a byte

La dirección de 1 byte de los maestros se divide en 2 palabras de 4 bits. Estas dos palabras se conocen como clase prioritaria y subclase, tal como se muestra a continuación:

	<u>Subclase</u>	<u>Clase prioritaria</u>
Maestro 1:	0000	0000
Maestro 2:	0001	0000
Maestro 3:	0011	0000
Maestro 4:	0111	0000
Maestro 5:	1111	0000
Maestro 6:	0000	0001
Maestro 7:	0001	0001
.....		
Maestro 23:	0011	1111
Maestro 24:	0111	1111
Maestro 25:	1111	1111

Este es el motivo por el que sólo puede haber 25 maestros. Las direcciones son limitadas porque si no el proceso de arbitraje orientado a byte no funcionaría.

Aun usando estas direcciones, cuando los maestros envían sus telegramas (empiezan con su dirección), pueden no reconocerse en su buffer de recepción, como se aprecia en la Ilustración 9 (nótese que UART envía los bits desde el bit menos significativo al más significativo).

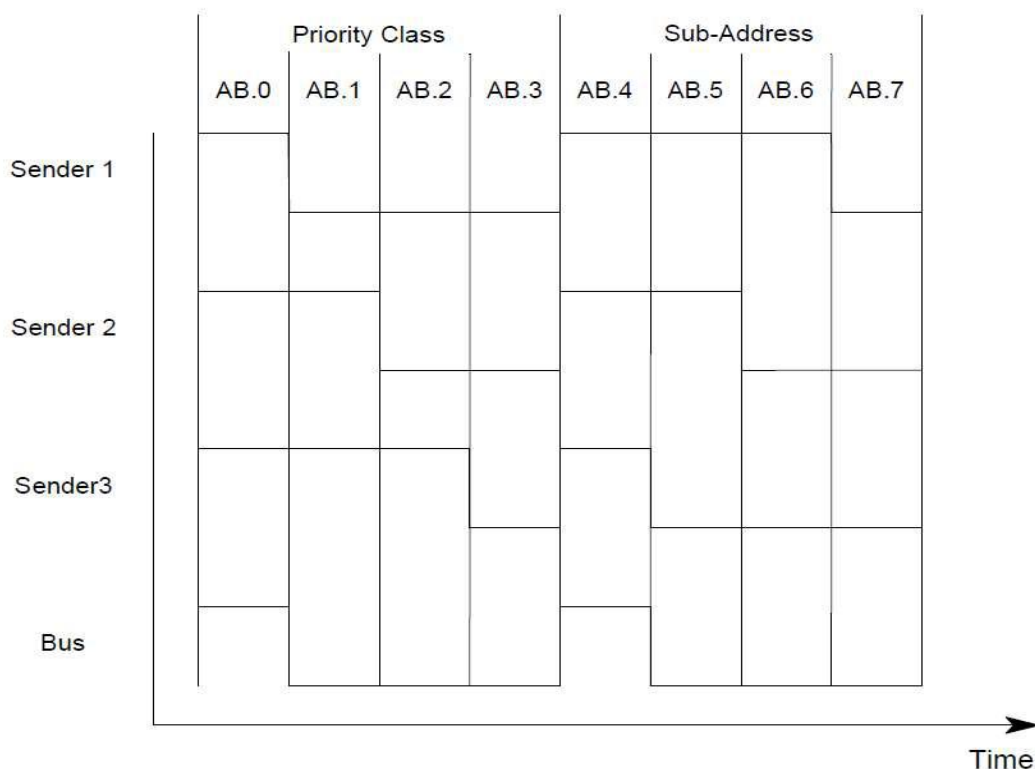


Ilustración 9: Arbitraje

(Fuente: eBus Interest Group)

En este caso el Maestro 1 tiene la dirección 0111 0001, el Maestro 2 la 0011 0011 y el Maestro 3 la 0001 0111. En el bus se obtiene el 0001 0001 que no coincide con ninguno de las direcciones de los maestros.

En este momento entra en juego la clase prioritaria. Los maestros dejan de enviar el telegrama en cuanto comprueban que en el bus no ha salido su dirección. En este momento comprueban los primeros 4 bits (que es la clase prioritaria) y si coincide con su dirección prioritaria, volverán a transmitir su telegrama la próxima vez que reciban el símbolo de sincronización. Los maestros que no encuentren la coincidencia en la clase prioritaria no pueden acceder al bus en la siguiente transmisión.

En el ejemplo mostrado en la Ilustración 9, gana el arbitraje el Maestro 1 al coincidir la clase prioritaria y en la siguiente transmisión, al enviar el telegrama únicamente él, va a ver su dirección en el buffer y va a transmitir su telegrama completo.

Suponiendo un Maestro 4 con la dirección 0011 0001, los Maestros 1 y 4 pasarían la primera fase, pues ambos tendrían la clase prioritaria 0001, mientras que los Maestros 2 y 3 se retiran igual que en ejemplo anterior. Sin embargo, al llegar un nuevo símbolo de sincronización, los Maestros 1 y 4 transmiten su telegrama y el Maestro 4 verá en su buffer su dirección por lo que sigue transmitiendo su telegrama hasta el final, mientras que el Maestro 1, que en anterior ejemplo ganó el arbitraje, ahora ve otra dirección que no es la suya, por lo que se retira.

En resumen, el arbitraje se puede resolver en la primera transmisión o como máximo en la segunda, en caso de haber varios participantes con la misma clase prioritaria.

3.3.4.2 El acceso al bus tras el símbolo de sincronización

Como se ha indicado, el acceso al bus debe ocurrir tras un símbolo de sincronización, para que el arbitraje sea válido. Cada participante necesita un cierto tiempo de proceso para decidir si quiere mandar un telegrama o no, el cual viene dado por el tiempo del programa (cargado en un microprocesador) que se inicia tras captar un símbolo de sincronización. Normalmente estos tiempos son diferentes entre cada participante, pero deben estar comprendidos en un rango de tiempos para que el arbitraje pueda ser válido.

Para dos participantes que acceden al bus, existe un tiempo $t_{\max}$ que indica cuánto tiempo puede desfasarse el segundo participante respecto del primero. En caso de haber más participantes, $t_{\max}$ será el tiempo máximo entre el primer participante y el último en acceder al bus.

En UART cada tiempo de bit se divide en 16 segmentos. Como eBus tiene una tolerancia de $\pm 1.2\%$ y su velocidad de modulación es de 2400 bps se puede calcular que el tiempo de cada segmento varía entre 25.73 y 26.36 μs . Los segmentos 7, 8 y 9 son los únicos válidos para el escaneo. Teniendo esto en cuenta y haciendo una

serie de cálculos en los que no se entrará en detalle, se puede calcular $t_{\max}$ como 104.40 μ s. Este es el máximo retardo de tiempo que garantiza que los bits de stop de todos los transmisores se escanean en sus segmentos de forma adecuada.

3.3.5 GENERADOR AUTOMÁTICO DE SÍMBOLOS DE SINCRONIZACIÓN

Cuando el bus se queda en nivel alto de forma permanente, entra en juego el generador automático de símbolos de sincronización. Como se ha explicado anteriormente, el símbolo de sincronización forma parte de los telegramas y es enviado por el último participante de una comunicación para indicar que el bus ha quedado libre. En este momento cualquier participante que tenga datos para enviar almacenados en su buffer de envío, está autorizado para comenzar la transmisión. Si ningún participante transmite, o no estaban listos para transmitir en el momento del símbolo de sincronización, nadie accederá al bus, y tendrán que esperar a un nuevo símbolo de sincronización (Ilustración 10). De esta forma, nadie será habilitado para transmitir y el bus quedará paralizado. En ese momento, después de que expire el periodo de validez del anterior símbolo, el generador automático enviará un símbolo para que todos los participantes que querían transmitir (si es que hay alguno), puedan acceder al bus.

Para evitar retrasar las comunicaciones, en caso de que después de una comunicación el bus quede en reposo durante 35 ms (como máximo), el generador enviará un símbolo. Si los últimos caracteres registrados en el bus no corresponden a un símbolo de sincronización, el generador debe esperar al menos 30 ms. Por último, entre símbolos de sincronización se permite la generación de otro símbolo a partir de 5 ms.

En conclusión, el bus nunca estará paralizado más de 35 ms por inactividad.

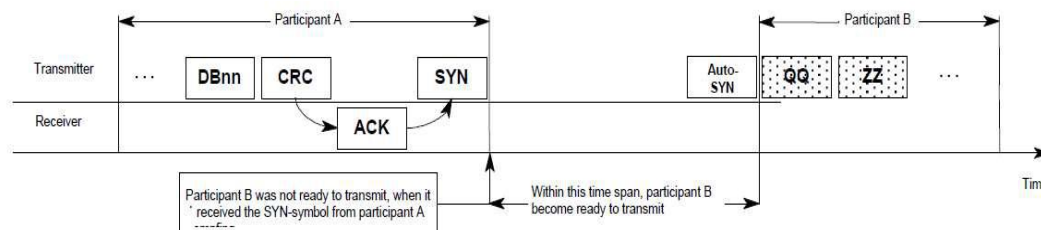


Ilustración 10: Generador de símbolo de sincronización

(Fuente: eBus Interest Group)

Al menos debe existir un generador automático de símbolos en el bus. Éstos pueden estar integrados en la alimentación del bus o en los maestros. Para los generadores, en conteo debe comenzar en cualquier flanco de subida del bus, aunque sea en medio de una comunicación. Con cada flanco de bajada se resetea el programa y el generador no debe bajo ningún concepto enviar un símbolo, ya que podría comprometer las comunicaciones que se estén llevando a cabo en ese momento. Sin embargo, así debe funcionar, iniciando el conteo en los flancos de subida para poder ser preciso a la hora de enviar un símbolo de sincronización legítimo.

3.3.6 DETECCIÓN DE ERRORES: CRC

La presencia de ruido en el canal de transmisión y los demás efectos de las transmisiones pueden provocar que el receptor interprete erróneamente la energía recibida en forma de señal, lo que se conoce como error. Existen varias técnicas para la detección de errores, como la paridad, el checksum, el CRC, etc. En eBus el único que se utiliza es el CRC, situado como último elemento del telegrama, tal como se ha explicado anteriormente.

CRC significa código de redundancia cíclica (cyclic redundancy check). Los CRC pueden ser de 1 o varios bytes, pero en eBus es de 1 byte. Además, los CRC llevan asociados un polinomio generador. Existen diversos polinomios y cada uno genera un CRC único, es decir, que no coincide con el generado en otros protocolos como

podría ser Modbus. Por lo tanto, hay que tener un especial cuidado en el cálculo del CRC y no usar códigos válidos de otros protocolos.

En eBus, el polinomio generador es: $x^8 + x^7 + x^4 + x^3 + x + 1$. Para su cálculo se pueden usar varios métodos, entre los que destacan los dos que se detallan a continuación.

3.3.6.1 Cálculo mediante división de polinomios [4]

Dada una trama M de k bits, el transmisor genera una secuencia F de n bits (CRC) de tal manera que la trama resultante a enviar T, que es de (k + n) bits, sea exactamente divisible por un patrón determinado P de n + 1 bits (este patrón es el polinomio generador).

El receptor, una vez que ha recibido la trama T, la divide por el patrón P y, si el resto es 0, se asume que no ha ocurrido ningún error. La secuencia F es el resto de dividir los bits del mensaje desplazados n posiciones a la izquierda por el patrón P. En la Ilustración 11 se muestra un ejemplo de cálculo del CRC, que en este caso vale 01110 en binario. Este ejemplo no correspondería con un CRC de eBus porque el patrón es tan sólo de 6 bits y en eBus debe ser de 8 bits. El patrón 110101 del ejemplo correspondería con el polinomio generador $x^5 + x^4 + x^2 + 1$.

Trama a enviar (M)	1010001101 (10 bits)
Patrón (P):	110101 (6 bits)
Checksum (F):	a calcular (5 bits)
Operación <pre> 101000110100000 110101 110101 0111011 110101 00111010 110101 00111110 110101 00101100 110101 0110010 110101 0001110 </pre>	
Trama enviada	101000110101110 (15 bits)

Ilustración 11: Cálculo de CRC mediante división de polinomios

3.3.6.2 Cálculo mediante registros

Este método es realmente una implementación hardware de la división con polinomios, es decir, verdaderamente el funcionamiento de los registros es matemáticamente equivalente al método anterior.

En la Ilustración 12 se muestra un ejemplo de registros que calcularían el CRC de eBus. Funciona de la siguiente manera: las 8 cajas representan el octavo grado del polinomio característico tal como se muestran y por ello el CRC será de 8 bits. Delante de las cajas que coinciden con el polinomio ($x^8 + x^7 + x^4 + x^3 + x + 1$) se colocan puertas XOR. En el instante inicial los valores de cada caja son 0.

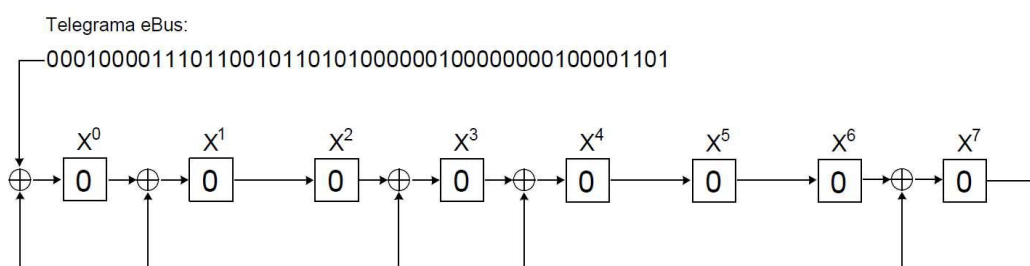


Ilustración 12: Cálculo de CRC mediante registros (estado inicial)

Bit a bit, los datos en forma de vector (0x10ECB504010D) van entrando por la izquierda, a la vez que el registro pasa datos a la derecha y a la realimentación. Cuando el último bit de la trama ha entrado en el registro, los valores que han quedado en las cajas forman el CRC de eBus. Como se puede observar en la Ilustración 13, el CRC sería 11000101 en binario (0xC5). El telegrama completo se conseguiría añadiendo el CRC como último elemento, quedando 0x10ECB504010DC5.

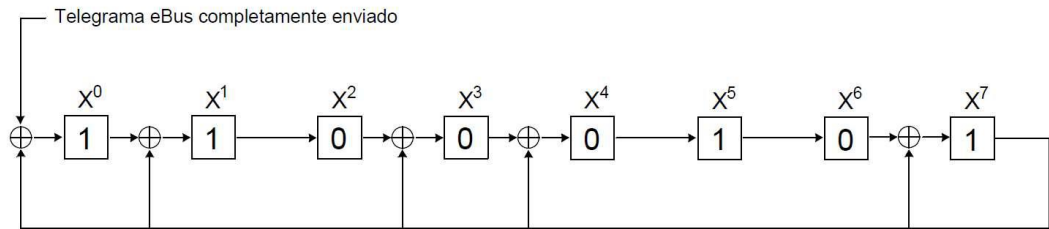


Ilustración 13: Cálculo de CRC mediante registros (estado final)

3.3.7 TRANSMISIÓN DEL MENSAJE

Para concluir la explicación de la capa de enlace de datos, a continuación, se resume como se produce una comunicación completa:

1. El maestro que haya ganado el arbitraje manda la dirección de destino seguido del resto del mensaje. El último símbolo enviado es el CRC calculado a partir de la trama. El CRC podría ser transmitido de forma expandida, tal como se ha explicado en el apartado 3.3.3.1, debido a que el símbolo 0xAA está reservado.

A no ser que el telegrama sea en modo broadcast, el maestro esperará una respuesta de acuse (ACK), ya sea positiva o negativa.

2. Si el receptor recibe el mensaje correctamente, es decir, el CRC que calcula coincide con el recibido, mandará un ACK positivo, de valor 0x00. Si no coincide con exactitud, el ACK será negativo, con valor 0xFF. Si el mensaje era en modo broadcast, el receptor no mandará ACK.
3. En caso de que un participante (maestro o esclavo) reciba un ACK negativo, volverá a mandar el telegrama lo más rápido posible, antes de que se produzca un símbolo de sincronización por parte de un generador automático. Para evitar el bloqueo del bus por parte de un participante que esté continuamente reenviando el telegrama, el número de repeticiones se fija a 1, es decir, si el participante vuelve a fracasar en su envío, no debe

seguir intentándolo, haciendo que el generador envíe un símbolo de sincronización para que otros participantes puedan acceder al bus.

4. Cuando la comunicación es entre maestros, no hay una respuesta directa, sino que antes de liberar el bus puede haber varios mensajes mandados entre otros maestros.

Cuando la comunicación es maestro-esclavo, el esclavo debe mandar su respuesta (empezando por el ACK) antes de que el generador automático pudiera mandar un símbolo de sincronización. Con la respuesta del esclavo, el maestro calcula el CRC y lo compara con el recibido. Dependiendo de si coincide o no, se actúa de la misma forma que se ha descrito en el tercer punto.

5. Después de haber recibido un ACK positivo tras todos los telegramas de una comunicación, el maestro manda el símbolo de sincronización para liberar el bus y dejar que comience de nuevo el proceso. Que el maestro envíe el símbolo solo sirve realmente para acortar el tiempo de espera del generador automático de símbolos.

3.4 CAPA DE APLICACIÓN [9]

En eBus, la capa de aplicación define todos los comandos con los que se puede llevar a cabo la lectura y escritura de datos en el dispositivo de destino. Esta información no suele ser proporcionada por el fabricante, sino que los usuarios, mediante técnicas de espionaje en el bus, obtienen la forma y significado de los distintos telegramas.

Este es el precisamente el caso con eBus, la información de la capa de aplicación puede ser encontrada en internet, pero es un documento incompleto y no está respaldada por ningún organismo ni es oficial, por lo que deberá ser utilizada bajo la responsabilidad de cada individuo.

Capítulo 4 ADAPTADOR EBUS

En este capítulo se explica en detalle el diseño y funcionamiento del adaptador de tensión eBus, un elemento indispensable para hacer posible el establecimiento de comunicaciones con la caldera.

4.1 INTRODUCCIÓN

Para poder llevar a cabo las comunicaciones con la caldera ecoTEC Plus, se debe utilizar el protocolo de comunicaciones eBus. Como se ha mencionado en el Capítulo 3, este protocolo trabaja con tensiones muy diferentes a las que acostumbran otros protocolos. En este caso y, en resumen, los ceros lógicos son representados por tensiones de unos +10V y los unos por tensiones de aproximadamente +20V. Por supuesto, hay un cierto margen en estas tensiones que permite algo de flexibilidad en las comunicaciones. Es decir, este protocolo utiliza únicamente tensiones positivas, a diferencia de la inmensa mayoría de protocolos como pueden ser RS-232, RS-485, SPI, I2C o MODBUS, que bien utilizan valores de tensión positivos y negativos, bien usan el valor de 0V para codificación de los ceros y valores positivos para los unos.

Por otro lado, para codificar la lógica binaria, el microcontrolador utilizado usa 0V y 3.3V (para ceros y unos lógicos respectivamente).

Es evidente, que para que el microprocesador pueda comunicarse con los elementos conectados al bus, se deben adaptar sus tensiones a las tensiones de bus. Para la mayoría de los protocolos, existen soluciones comerciales muy simples que consisten en un chip que adapta los ceros y unos lógicos a los niveles adecuados del bus. Sin embargo, en el caso de eBus, debido a ser un protocolo bastante desconocido con aplicaciones realmente restringidas, no existen soluciones comerciales que permitan adaptar los niveles de tensión. Es por ello por lo que se

ha decidido diseñar y probar un circuito que permita adaptarlas tensiones para poder tanto transmitir como recibir información.

4.2 DISEÑO

El diseño para adaptar los niveles de tensión consiste en un circuito analógico con varios componentes discretos e integrados (Ilustración 14). El funcionamiento es relativamente simple, aunque se debe entender con precisión cada uno de los elementos y su utilidad en el circuito.

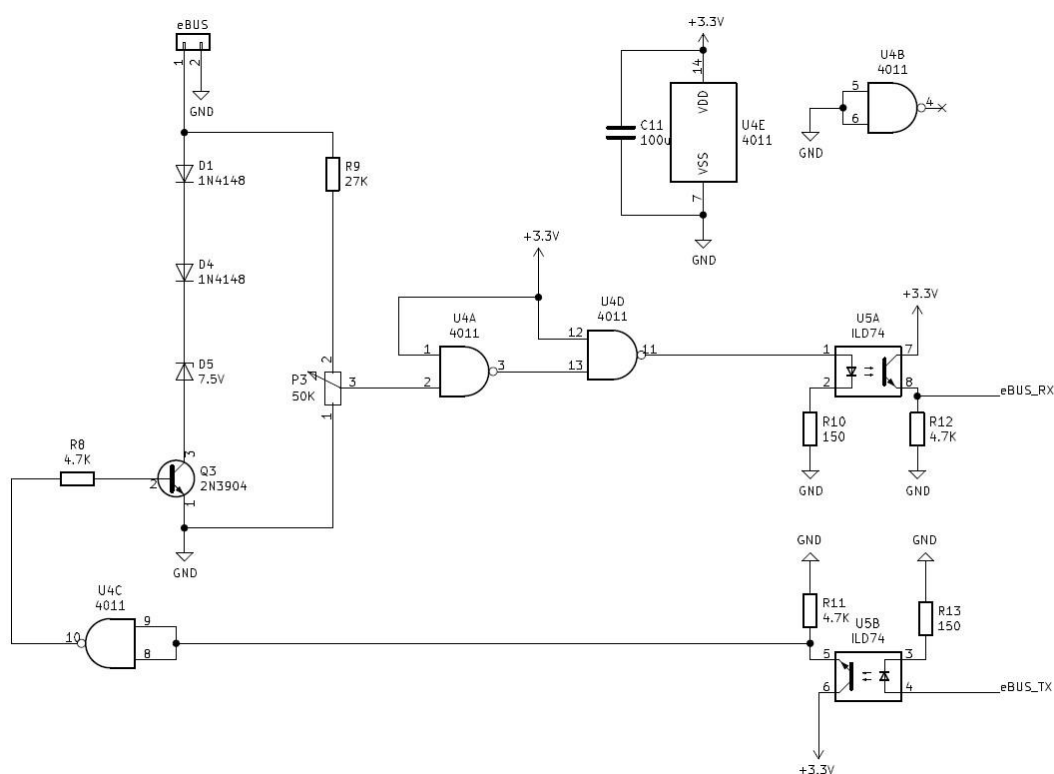


Ilustración 14: Adaptador eBus

El circuito permite elevar los niveles de tensión que vienen desde la transmisión del microprocesador (eBus_TX) y disminuir las tensiones del bus para conducir las a la recepción (eBus_RX). Consta de los siguientes elementos:

- Integrado con 4 puertas lógicas NAND.
- Integrado con un optoacoplador dual.
- Transistor NPN.
- Diodo Zener de 7.5V.
- Diodos de conmutación.
- Potenciómetro de 50K Ω .
- Resistencias de distintos valores de 0.25W.

Para comprender adecuadamente el funcionamiento del circuito, hay que tener claros los siguientes puntos:

- La tensión de reposo del bus es de unos 20V, es decir, se encuentra en el estado de '1' lógico.
- Para transmitir un '0' lógico, lo que se hace es "tirar" la tensión del bus a unos 10V, es decir, forzar una bajada de tensión en el bus. Cuando se quiere transmitir el '1' simplemente no hay que tocar el bus, que recupera automáticamente su tensión.

Con estos conceptos, se puede entender fácilmente como transmitir información al bus. Supóngase que se quiere transmitir un '0'. En este caso, por la UART del micro se transmite una tensión de 0V. Como el optoacoplador no se excita, la tensión al otro lado es también de 0V. La puerta NAND con las entradas puenteadas actúa con la lógica de una puerta NOT, es decir, en este caso daría un '1' a su salida y por tanto haría que el transistor trabaje en saturación. El diodo Zener con los otros dos diodos de conmutación, obligan a que la tensión en el bus sea de aproximadamente 10V (7.5 del Zener, 0.7V de cada diodo y aproximadamente 1V del transistor). Al haber bajado la tensión en el bus, todos los dispositivos conectados verán el '0'.

Si lo que se quiere es transmitir el '1', siguiendo la lógica anterior, en este caso el transistor trabaja en corte y la tensión del bus no se ve afectada al no haber conducción por los diodos, quedándose en sus 20V de reposo que todos los dispositivos ven como '1'. Se puede ahora entender la importancia de que la UART en reposo esté con tensión alta, puesto que en caso de tener 0V en reposo, estaría tirando la tensión del bus, incapacitando a cualquier dispositivo de la posibilidad de transmitir.

En el caso de la recepción, utilizando un simple divisor resistivo y un par de puertas NAND para estabilizar la tensión, 20V se transmiten a la recepción de la UART a través del optoacoplador como '1' lógico y 10V como '0'. La utilidad del potenciómetro es calibrar el divisor a la tensión para que el integrado NAND reconozca adecuadamente las tensiones que reciba en los dos valores binarios.

Nótese que debido a que todos los dispositivos conectados al bus ven siempre el estado del bus (sus variaciones de tensión), cuando un micro transmite una trama la va recibiendo a la vez que la transmite. Esto no supone un problema, pero habrá que tenerlo en cuenta en la programación, para no confundir la transmisión propia con la respuesta de otro dispositivo.

4.3 PRUEBAS DE FUNCIONAMIENTO

Se ha procedido a continuación al montaje físico para la comprobación de sus características. Se ha empleado una protoboard, un Arduino y material del laboratorio, como se muestra en la Ilustración 15.

Para la alimentación del integrado de puertas NAND se ha utilizado un regulador de tensión estándar. En la Ilustración 16 se muestra un esquema del circuito, aunque no se utilizará en el circuito definitivo. La conexión desde la R4 es directa al bus, ya que el regulador puede alimentarse siempre desde el bus, a tensión alta o baja.

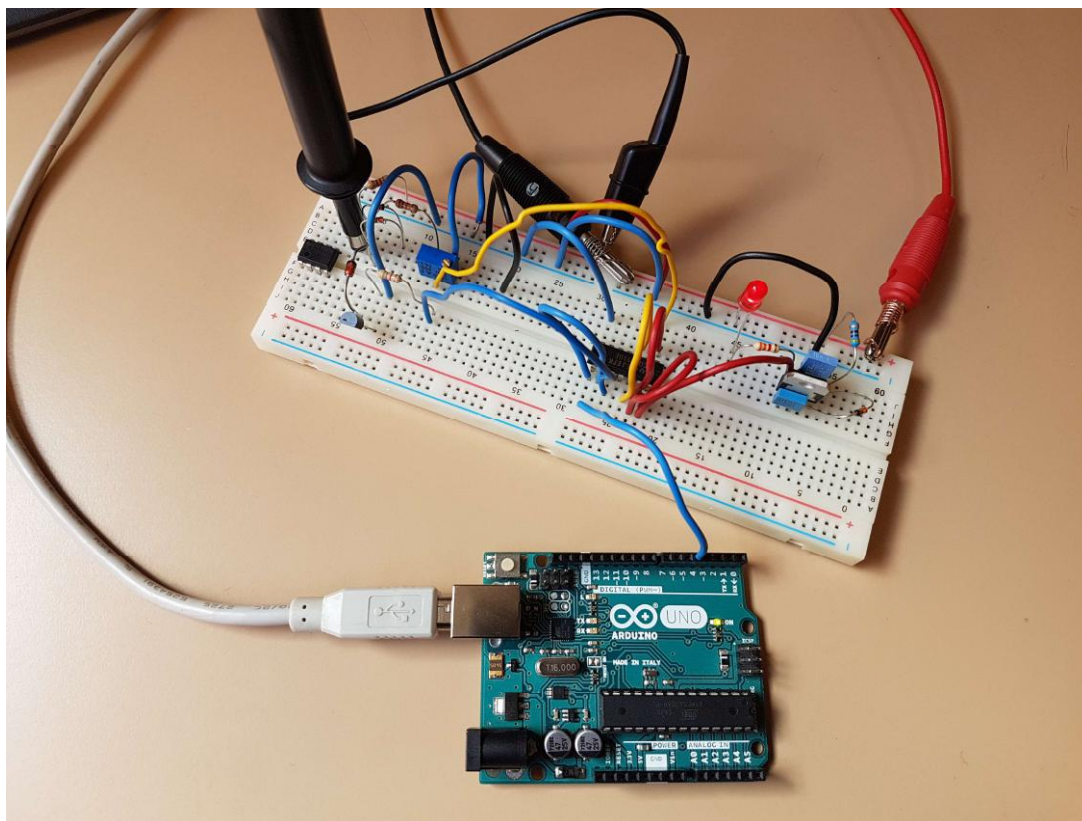


Ilustración 15: Adaptador eBus en Protoboard

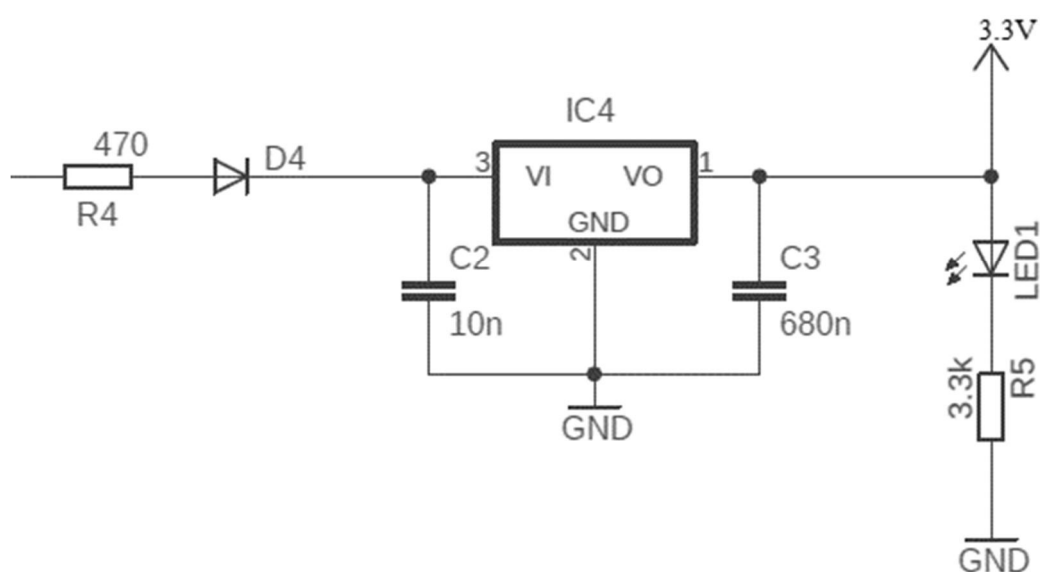


Ilustración 16: Regulador de tensión auxiliar para pruebas

Se han llevado a cabo 3 experimentos para comprobar que el circuito es capaz de adaptar las tensiones adecuadamente. Dichos experimentos se describen a continuación.

4.3.1 EXPERIMENTO I

Dado que no se tiene disponible un bus tipo eBus real, se ha decidido utilizar una fuente de tensión a 20V emulando la conexión real al bus. Como se ha mencionado anteriormente, el circuito hace caer la tensión en el bus, pero al tener conectada una fuente fija de tensión, no se puede hacer variar su valor. Por ello se ha conectado una resistencia (R_{bus}) entre el diodo D1 y eBus (véase Ilustración 14). R_{bus} no será un elemento del circuito final, por ello no se muestra en la ilustración, aunque cabe destacar que un valor de 100Ω y $1/2W$ es razonable, pues no debe caer mucha tensión, pero tampoco debe quemarse, por lo que $1/4W$ no es lo más adecuado.

La transmisión resultó exitosa, en la Ilustración 17 se puede comprobar la variación de la tensión del “bus” con la transmisión de los valores lógicos desde el Arduino. La tensión es oscilante entre 9.15V y 19.8V, lo que entra dentro de los rangos que marca el protocolo.



Ilustración 17: Transmisión I

Para la recepción se ha retirado la alimentación fija y se alimenta el circuito con una onda cuadrada a unos 2.5kHz (más velocidad que la que emplea eBus) de 7V pico-pico con un offset de 12.5V para poder seguir alimentando al regulador de tensión sin dar problemas a los integrados. El osciloscopio se ha conectado en la recepción del eBus (ver Ilustración 14). Sólo tras el recalibrado del potenciómetro se ha logrado obtener la señal con las tensiones adaptadas (Ilustración 18).

Según se ha podido comprobar, el potenciómetro parece tener un margen de trabajo relativamente amplio. Variando desde 10k Ω , se ha obtenido la recepción al llegar a unos 33k Ω y hasta los 40k Ω se ha mantenido.

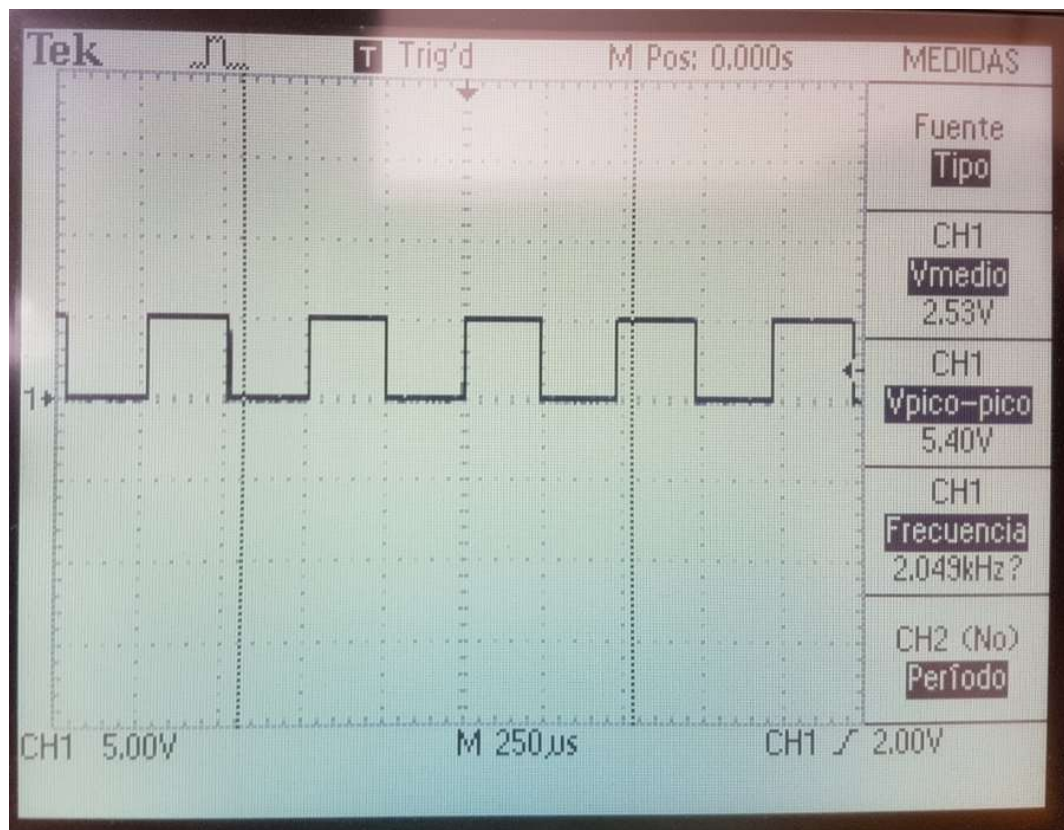


Ilustración 18: Recepción I

4.3.2 EXPERIMENTO II

En el segundo experimento se ha podido testear la transmisión a la velocidad que marca el protocolo, 2400bps (2.4kHz) usando el Arduino. El resultado ha sido satisfactorio como se muestra en la Ilustración 19, con una variación de tensiones en el “bus” de 11.3V a 19.9V, que siguen en el rango válido según el protocolo.

Se ha observado que si la tensión del bus es mayor de 20V (hasta los 24V) cuesta cada vez más la transmisión de los ceros, hasta que a más de 24V los ceros se salen del rango válido.

En este caso se ha detectado un calentamiento inusual del diodo Zener. Se ha dejado la transmisión durante más de 1 hora sin que se estropee ningún componente, por lo que se asume que la transmisión es válida.

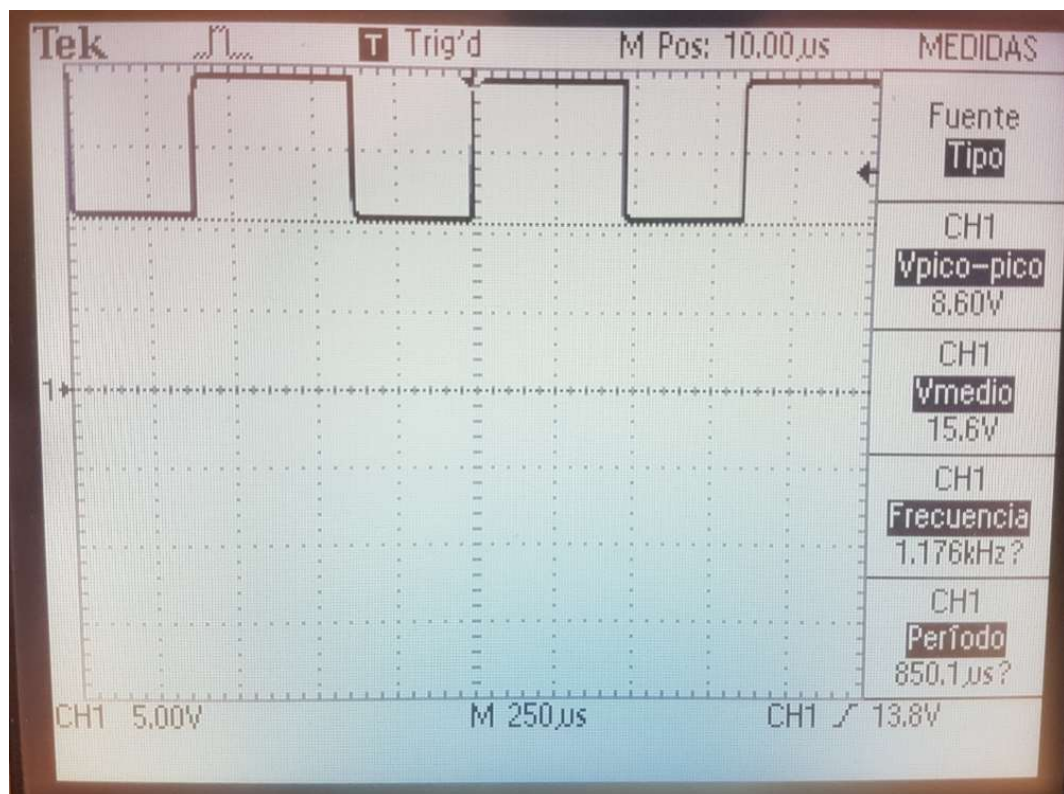


Ilustración 19: Transmisión II

4.3.3 EXPERIMENTO III

En el último experimento se ha introducido el optoacoplador ILD74 en el circuito, para comprobar que puede funcionar adecuadamente a frecuencias de 2400bps.

Primeramente, se han diseñado los valores de las resistencias para limitar intensidad, obteniendo los valores finales que pueden verse en la Ilustración 14. Sin embargo, estos valores son perfectamente variables, pues se ha podido comprobar que el optoacoplador tiene bastante flexibilidad de trabajo.

Los resultados han resultado exitosos, tanto en la transmisión como en la recepción, se han obtenido resultados similares a los Experimentos I y II.

Capítulo 5 TARJETA CONTROLADORA

En este capítulo se detalla el diseño y funcionamiento del elemento más importante del proyecto, la tarjeta controladora, que se puede considerar como el centro de la arquitectura del proyecto.

5.1 INTRODUCCIÓN

Como se mencionó en el Capítulo 2, la tarjeta controladora (Ilustración 20) es el elemento central, ya que actúa como nodo central y es la encargada de la transmisión y procesamiento de información desde el usuario (dispositivo móvil) al objetivo (la caldera) y viceversa. Está gobernada por un microcontrolador que actúa como el cerebro de la placa, procesando los telegramas que llegan desde la caldera y el módulo Wifi y enviando los suyos propios.

La controladora consta de varios módulos que se detallan en el siguiente apartado.

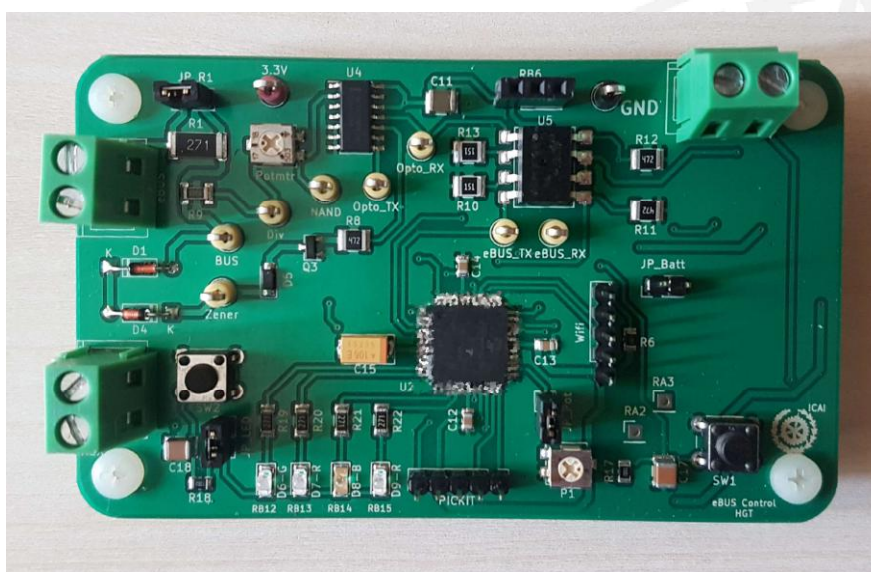


Ilustración 20: Tarjeta Controladora

5.2 DISEÑO

El diseño de la controladora se podría dividir en los siguientes módulos: microcontrolador (incluyendo 2 pulsadores, 4 diodos LED y 1 potenciómetro), adaptador eBus, regulador, sensor de temperatura, medidor de batería y conectores. Todos ellos se explican con detalle en los siguientes subapartados.

5.2.1 MICROCONTROLADOR

Algunas de las características del microcontrolador utilizado son las siguientes:

- Procesa hasta 40 MIPS (millones de instrucciones por segundo)
- Rango de temperatura de trabajo desde -40°C hasta +125°C
- Tensión de alimentación entre 3V y 3.6V
- Arquitectura Harvard modificada
- 2 módulos de UART
- 2 conversores AD de 10 bits (1.1Msps) o 12 bits (500Ksps) con 13 canales
- 5 timers de 16 bits
- 16 pines remapeables
- 35 pines Input/Output

En la Ilustración 21 se puede ver el conexionado del MIC con sus 44 pines. Hay pines que no han sido conectados (12, 13, 25, 26, 27, etc.) porque no ha habido necesidad de utilizarlos. Otros llevan etiquetas (PGED, PGEC, SCL, SDA, etc.), que son pines que van conectados a otras partes del circuito que se explicarán en los próximos apartados. Por último, hay otros elementos muy simples que se han dejado en la imagen y que se detallan en este mismo apartado.

Es importante que el micro tenga condensadores de desacoplo en las alimentaciones como se puede ver en la parte superior del micro en la Ilustración 21. Estos condensadores, aparte de proveer de alimentación al micro en caso de caídas puntuales de tensión o cuando el micro trabaja de forma intensiva, ayudan a filtrar la señal de alimentación. Por ello se recomienda que sean condensadores cerámicos de $0.1\mu\text{F}$ con la frecuencia de resonancia en el rango de los 20MHz. En el caso particular de la V_{CAP} (pin 7), que sirve para estabilizar la tensión de salida del regulador interno del micro, se recomienda que el condensador sea de entre $4.7\mu\text{F}$ y $10\mu\text{F}$, pudiendo ser cerámico o de tántalo. Para el diseño se ha escogido uno de tántalo de $10\mu\text{F}$ (es importante recordar que tiene éste tiene polaridad).

Los condensadores de desacoplo deben conectarse lo más cerca posible de los pines del micro, al ser posible en la misma cara de la placa y no se deben sobrepasar los 6 mm de longitud en la pista (cable en la PCB).

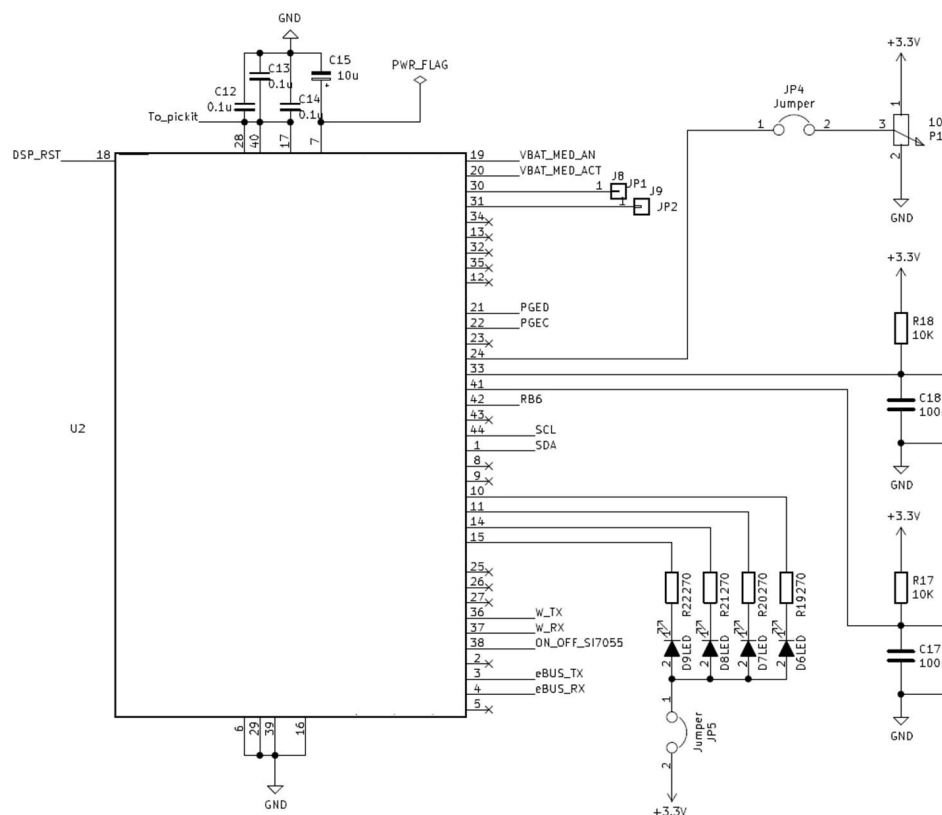


Ilustración 21: Esquema conexionado del MIC

El potenciómetro del pin 24 se ha integrado como una opción de poder hacer controles analógicos mediante un divisor resistivo. No es segura ni estrictamente necesaria su utilización, por lo que se ha puesto un jumper para no consumir energía innecesariamente (aunque realmente está mal posicionado y siempre estará consumiendo corriente (0.33mA)). Su conexión debe ser a un pin analógico, o no podrá servir a su propósito.

Hay dos pulsadores conectados a los pines 33 y 41, que son puertos digitales del micro y que se deben configurar como entradas. Son muy útiles tanto en la aplicación real (por ejemplo, para cambiar entre varios modos) como en técnicas de depuración. Los condensadores que llevan en paralelo sirven para evitar falsos positivos en un efecto conocido como rebote, debido a las características mecánicas del pulsador. En reposo la señal que llega al micro es de nivel alto (3.3V). Cuando se pulsa el condensador se descarga relativamente lento, evitando los efectos del rebote, y llevando la tensión que ve el micro a 0V. Cuando el pulsador se suelta, el condensador obliga a que la tensión que llega al micro se recupere lentamente, para evitar de nuevo los efectos del rebote.

Por último, se han conectado 4 diodos LED a los pines 10, 11, 14 y 15, que también son digitales. Se configuran como salidas y resultan muy útiles para mostrar estados del sistema en la aplicación real. Llevan resistencias en serie con la función de limitar intensidad para ahorrar energía, aunque realmente esto sirve para no fundir los diodos por sobreintensidad. De la misma forma que en el potenciómetro, se han integrado con un jumper para poder desconectarlos si se desea ahorrar energía.

5.2.2 ADAPTADOR EBUS

El adaptador de tensiones es un elemento crucial para poder efectuar comunicaciones con la caldera. Al no existir un circuito comercial, se ha diseñado y probado desde cero. En el Capítulo 4 se explica en detalle su funcionamiento.

5.2.3 REGULADOR

El regulador de tensión es el elemento que proporciona la alimentación a todos los elementos de la tarjeta. Se ha elegido un regulador del cual se dan a continuación algunas de sus características:

- Rango de tensiones de entrada desde 4V a 14V
- Tensión de salida ajustable a 1.8V, 2.5V o 3.3V
- Protecciones contra sobre corriente y altas temperaturas

En la Ilustración 22 se muestra el conexionado. La conexión está hecha para ajustar la tensión de salida a 3.3V, ya que es la tensión de alimentación del MIC. El resto de los componentes pueden ajustarse a esta tensión sin problema.

La tensión de salida se calcula mediante la fórmula $V_{out} = 1.238 * \left(1 + \frac{R5}{R4}\right)$ que da como resultado 3.30V.

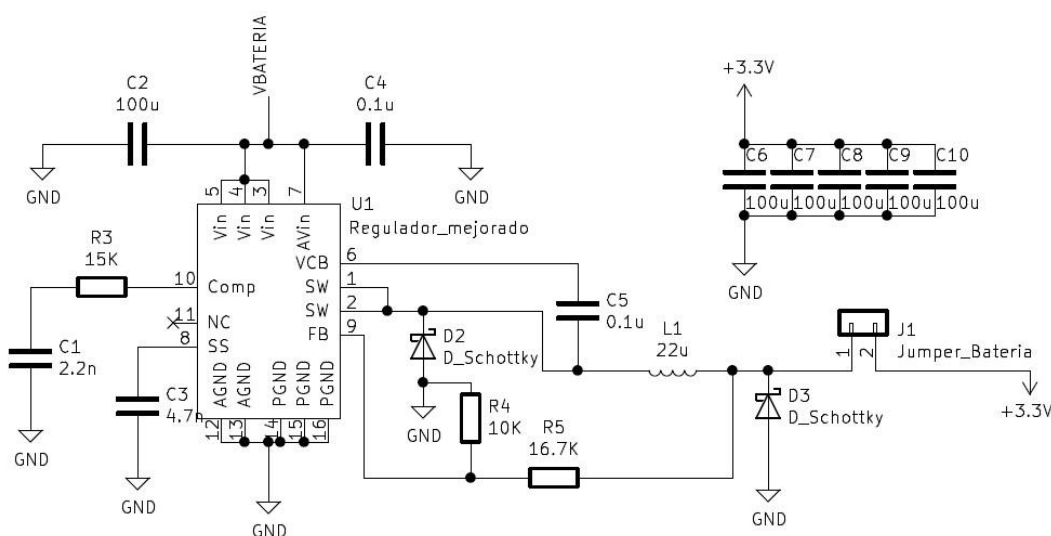


Ilustración 22: Regulador de tensión

El regulador trabaja a una frecuencia constante de 300 KHz, con un modo PWM de corriente para alta carga y cambia automáticamente a un modo de histéresis cuando funciona en baja carga.

El jumper que está a la salida del circuito sirve para poder desconectar el regulador del resto de la placa, ya sea para hacer pruebas sin comprometer el resto de los componentes o, como se verá más adelante, para protegerlo al alimentar la placa por una conexión auxiliar directa a 3.3V.

En la parte superior derecha de la Ilustración 22 se muestran unos condensadores (C6 a C10) conectados en batería. El sentido de esto es que actúen como una pequeña batería de respaldo. Cuando el regulador es alimentado, carga estos condensadores que tendrán la capacidad de asumir alguna hipotética caída de tensión del regulador o que algún componente exija un pico de consumo que el regulador no tenga capacidad de generar.

5.2.4 SENSOR DE TEMPERATURA

Como medida de seguridad se ha decidido montar un sensor de temperatura en la PCB. Hay que tener en cuenta que, al estar pegado literalmente a la tarjeta, la temperatura que mide es la de la propia placa y no la temperatura ambiente.

El sensor elegido también tiene capacidad de lectura de humedad, aunque en principio esta característica no se va a utilizar. Respecto a la temperatura, cuenta con buena precisión de medida, con una tolerancia de $\pm 0.5^{\circ}\text{C}$, suficientemente precisa para la aplicación de este proyecto. Otras características que tiene este sensor son:

- Tensión de alimentación entre 1.9V y 3.6V.
- Rango de operación desde -40°C a 125°C .
- Comunicaciones mediante I2C.
- Consumo bajo, de 195nA a 1Hz.

- Resolución en la medida de 14 bits.

En la Ilustración 23 se muestra el conexionado de este sensor en la placa.

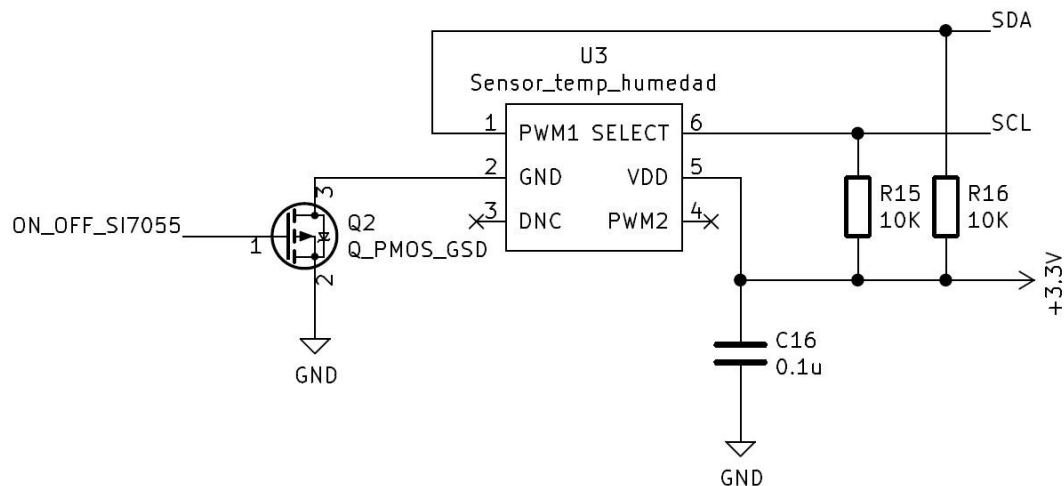


Ilustración 23: Sensor de temperatura

En la imagen, el transistor Q2 es un MOSFET que sirve para obtener la medida de la temperatura sólo cuando el usuario lo desee. Es decir, la señal ON_OFF_SI7055 se puede dejar a 0V de forma que el sensor no consuma, aunque cabe destacar que su consumo es muy bajo de todas maneras.

Las señales de PWM1 y SELECT que van a los pines del micro SDA y SCL respectivamente, se encargan de las comunicaciones por I2C, siendo SCL el reloj y SDA la línea de datos.

DNC y PWM2 se deben dejar sin conectar, aunque se recomiendan ser soldados a la PCB por simple estabilidad mecánica. Se pueden conectar a la alimentación (VDD) si se quiere, o dejarse sin conectar como se ha hecho en este caso, pero no se deben conectar a tierra.

5.2.5 MEDIDOR DE BATERÍA

El medidor de batería es un circuito muy simple con el que se puede calcular el porcentaje de batería que queda en la alimentación, para saber cuándo debe ser recargada. En la Ilustración 24 se muestra el circuito empleado.

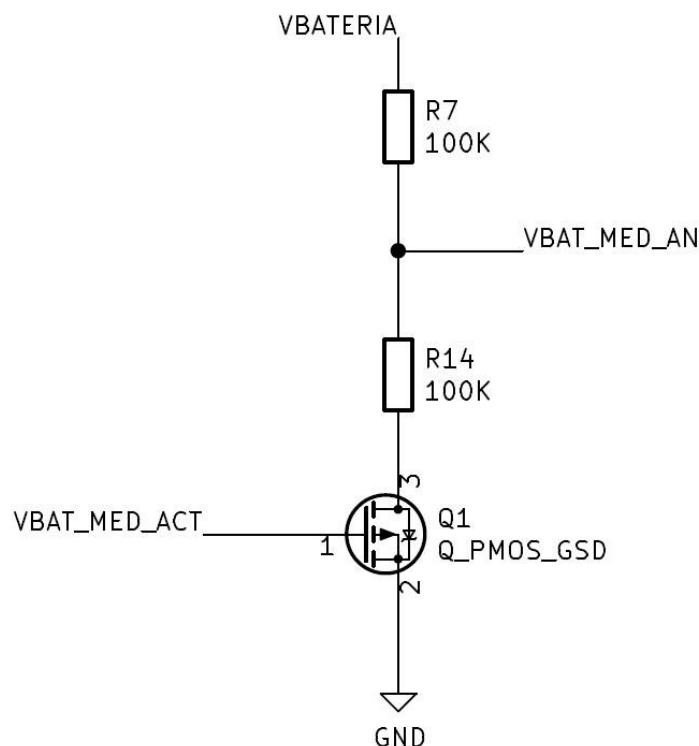


Ilustración 24: Medidor de batería

Su funcionamiento es tan simple como un divisor resistivo. La señal VBAT_MED_AN lleva la división de tensión a un puerto analógico del MIC que dará un porcentaje de la batería restante. A pesar su sencillez, obtener ese porcentaje no es tarea fácil, ya que una batería no se descarga de forma lineal ni hasta los 0V. Por lo tanto, para poder tener una medida fiable, se debe hacer un estudio sobre la descarga de las baterías para comprobar su linealidad y sus extremos de voltaje.

Por otra parte, para no consumir innecesariamente, se ha colocado un transistor MOSFET que no permite el paso de la corriente excepto cuando se habilita desde la señal VBAT_MED_ACT que es una salida digital del micro.

5.2.6 CONECTORES

Los conectores le sirven a la controladora para conectarse con otros periféricos, alimentaciones y realizar las comunicaciones necesarias. En resumen, le sirven para comunicarse con el mundo exterior. En la Ilustración 25 se muestran los conectores utilizados en este diseño:

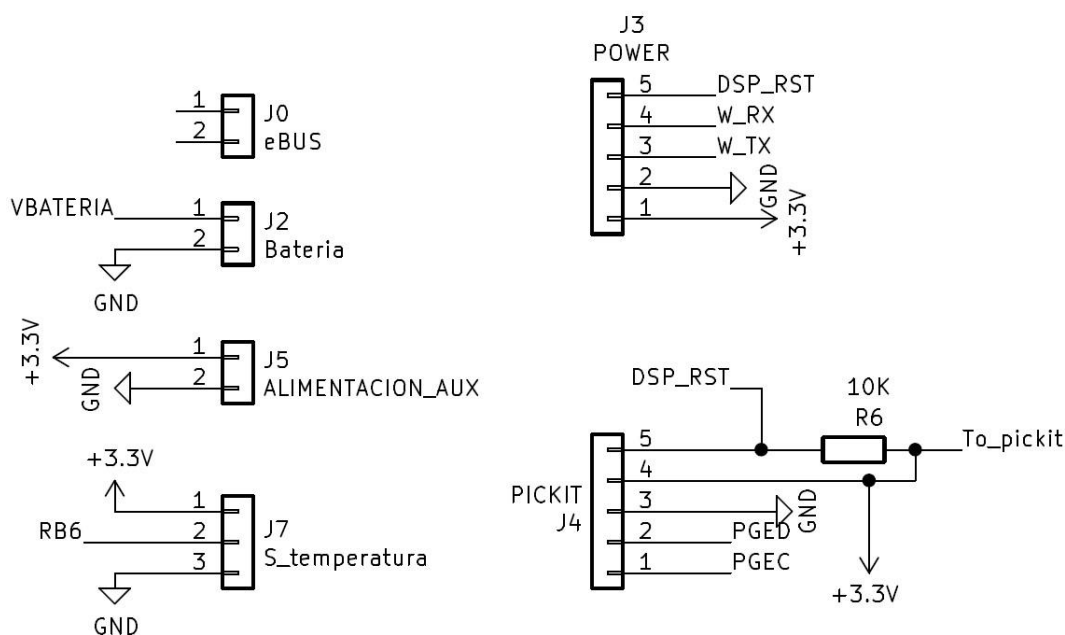


Ilustración 25: Conectores de la PCB

A continuación, se explica cada uno de estos conectores y su utilidad en el proyecto.

5.2.6.1 eBus

A través del conector eBus se establecen las comunicaciones con la caldera. Esta conexión forma parte del circuito del adaptador, explicado en el Capítulo 4.

5.2.6.2 Batería

En este conector se acopla la batería, que debe ser entre 4V y 14V, pues es la entrada directa al regulador de tensión, encargado de alimentar todo el circuito.

5.2.6.3 Alimentación auxiliar

Esta alimentación sustituye a la del regulador. Alimenta directamente a todos los elementos del circuito sin pasar por ningún regulador, por lo que debe ser de 3.6V como máximo, pues es la tensión máxima que soporta el MIC. Está pensada para ser alimentado desde una fuente de tensión variable de laboratorio. Si se usa este conector, se debe quitar el jumper del regulador, porque si éste se alimenta desde la salida se fundirá.

Este conector, que realmente no sería necesario se ha integrado por dos motivos:

1. Para trabajo en laboratorio y pruebas, pues es más cómodo no tener que cargar con baterías que se descargan.
2. En caso de que el regulador de tensión se estropee o no funcione, se tiene una conexión de respaldo independiente que siempre funciona.

5.2.6.4 Sensor de temperatura externo

El conector J7 de la Ilustración 25 nada tiene que ver con el sensor de temperatura superficial citado anteriormente. Este conector está reservado para conectar un sensor externo que mida la temperatura ambiente, a diferencia del otro sensor, que medía la temperatura de la superficie de la PCB.

La conexión tiene alimentación, tierra y un puerto exclusivamente digital del micro, ya que para hacer las pruebas se va a utilizar un sensor digital que funciona con el protocolo 1-Wire, es decir, por un único cable el sensor y el micro transmiten y reciben la información. El micro funciona como maestro y el sensor como esclavo.

5.2.6.5 Módulo Wifi

El conector J3, con el nombre de POWER es la conexión al módulo Wifi, con el que el micro podrá establecer las comunicaciones con el usuario.

5.2.6.6 PICkit

Este último conector, uno de los más importantes, sirve para poder reprogramar el microcontrolador desde el MPLAB. Sin esta conexión, el micro es totalmente inútil. El PICkit es un programador y el que se ha utilizado en este proyecto es el PICkit3, mostrado en la Ilustración 26:



Ilustración 26: PICkit3

5.3 FABRICACIÓN

Una vez diseñado el circuito se debe construir. Para ello, primero hay que utilizar un programa de diseño que permita crear un modelo digital de la PCB. Hecho esto, se manda a una industria especializada en la fabricación de PCB. Normalmente son industrias que se dedican exclusivamente a esta tarea. En China existen muchas empresas que realizan la tarea rápidamente y a precios asequibles.

El software de diseño que se ha empleado en este proyecto es el KiCAD, que tiene la gran ventaja de ser un freeware, es decir, que no se necesita una licencia para su utilización. Además, posee una gran comunidad de usuarios que se ayudan unos a otros en foros, suben a dominio público sus diseños y librerías de huellas y símbolos que el resto de los usuarios pueden utilizar con total libertad. A continuación, se describe cómo diseñar una PCB en KiCAD.

5.3.1 DISEÑO DE ESQUEMÁTICO

En primer lugar, KiCAD tiene un editor de esquemáticos donde se debe dibujar el circuito con símbolos (obtenidos de librerías o creados por el usuario).

Además, en el esquemático se pueden añadir otros símbolos que luego formarán parte del montaje, como los testpoint o los agujeros de montaje (Ilustración 27).

Los testpoint son puntos de medida del circuito. Se conectan a cualquier punto del circuito en que se desee conocer la tensión y son útiles para depurar problemas de funcionamiento en la tarjeta.

Los agujeros de montaje son simples agujeros que lleva la PCB para su montaje. No tienen conexión eléctrica, pero hay que indicar dónde deben estar (se sitúan normalmente en las esquinas de la placa).

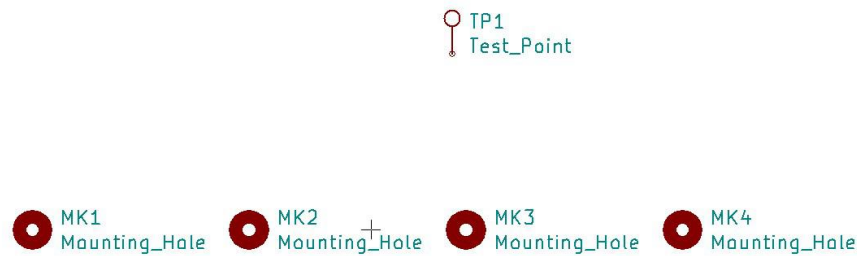


Ilustración 27: Testpoint y agujeros de montaje

5.3.2 ASIGNACIÓN DE HUELLAS

Cuando se ha terminado el diseño del esquemático, se debe asignar una huella a cada símbolo utilizado en éste. Las huellas son simplemente el tamaño real que tendrá el componente en la tarjeta. En la Ilustración 28 se puede ver cómo funciona la asignación de huellas en KiCAD. Tan solo se muestra a modo de idea las huellas de algunos componentes. Las huellas suelen tener métricas estandarizadas y son fáciles de encontrar en la datasheet del componente deseado.

La elección de algunas de las huellas no es crítica, como las resistencias THT (through-hole), que, aunque tengan una huella un poco más grande o pequeña se pueden encajar sin problema en la PCB gracias a sus flexibles extremos. Sin embargo, otras huellas pueden ser muy críticas y un error en su diseño puede tirar por tierra todo el trabajo. En componentes SMD (montaje superficial), si el componente real no encaja en su huella, puede resultar muy complicado o imposible hacer la conexión eléctrica. En la Ilustración 29 (que no forma parte de este proyecto), se pueden ver los “Pads” de las huellas de una PCB antes de que se le hayan soldado los componentes.

14	C14 -	0.1u : Capacitor_SMD:C_0805_2012Metric
15	C15 -	10u : Capacitor_Tantalum_SMD:CP_EIA-6032-28_Kemet-C
16	C16 -	0.1u : Capacitor_SMD:C_0805_2012Metric
17	C17 -	100n : Capacitor_SMD:C_1210_3225Metric
18	C18 -	100n : Capacitor_SMD:C_1210_3225Metric
19	D1 -	1N4148 : Diode_THT:D_DO-35_SOD27_P7.62mm_Horizontal
20	D2 -	D_Schottky : Diode_SMD:D_SOD-323
21	D3 -	D_Schottky : Diode_SMD:D_SOD-323
22	D4 -	1N4148 : Diode_THT:D_DO-35_SOD27_P7.62mm_Horizontal
23	D5 -	7.5V : Diode_SMD:D_SOD-123
24	D6 -	LED : LED_SMD:LED_1206_3216Metric
25	D7 -	LED : LED_SMD:LED_1206_3216Metric
26	D8 -	LED : LED_SMD:LED_1206_3216Metric
27	D9 -	LED : LED_SMD:LED_1206_3216Metric
28	J0 -	eBUS : Connector_Phoenix_MSTB:PhoenixContact_MSTBVA_2,5_2-G-5,08_1x02_P5.08mm_Vertical
29	J1 -	Jumper_Bateria : Connector_PinHeader_2.54mm:PinHeader_1x02_P2.54mm_Vertical
30	J2 -	Bateria : Connector_Phoenix_MSTB:PhoenixContact_MSTBVA_2,5_2-G-5,08_1x02_P5.08mm_Vertical
31	J3 -	POWER : Connector_PinHeader_2.54mm:PinHeader_1x05_P2.54mm_Vertical
32	J4 -	PICKIT : Connector_PinHeader_2.54mm:PinHeader_1x05_P2.54mm_Vertical
33	J5 -	ALIMENTACION_AUX : Connector_Phoenix_MSTB:PhoenixContact_MSTBVA_2,5_2-G-5,08_1x02_P5.08mm_Vertical
34	J7 -	S_temperatura : Connector_PinSocket_2.54mm:PinSocket_1x03_P2.54mm_Vertical
35	J8 -	JP1 : Connector_PinHeader_2.54mm:PinHeader_1x01_P2.54mm_Vertical
36	J9 -	JP2 : Connector_PinHeader_2.54mm:PinHeader_1x01_P2.54mm_Vertical
37	JP1 -	Jumper : Connector_PinHeader_2.54mm:PinHeader_1x02_P2.54mm_Vertical
38	JP4 -	Jumper : Connector_PinHeader_2.54mm:PinHeader_1x02_P2.54mm_Vertical
39	JP5 -	Jumper : Connector_PinHeader_2.54mm:PinHeader_1x02_P2.54mm_Vertical
40	L1 -	22u : Inductor_SMD:L_0805_2012Metric
41	MK1 -	Mounting_Hole : MountingHole:MountingHole_3.2mm_M3_DIN965

Ilustración 28: Huellas asignadas en KiCAD

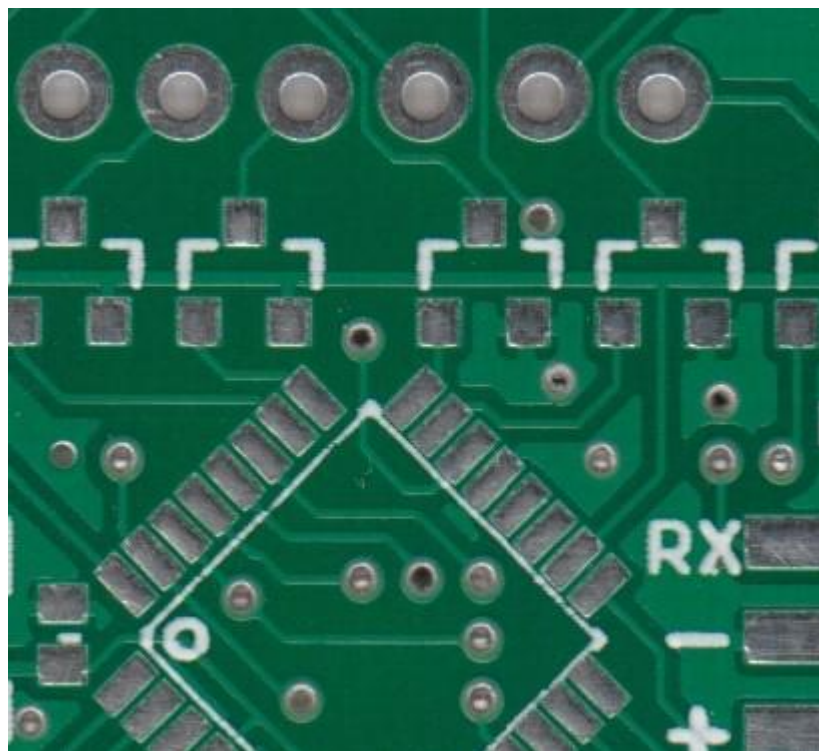


Ilustración 29: Pads SMD junto a THT

En la parte superior se encuentran 6 contactos THT (podrían ser el conector para el PICKit) y el resto son SMD (posiblemente los de arriba son transistores y el de abajo el microcontrolador).

5.3.3 DISEÑO DE PCB

Una vez se ha asignado su huella a cada componente, se utiliza el editor de PCB de KiCAD. En primera instancia, todos los componentes aparecen montados unos sobre otros sumidos en el caos, como se aprecia en la Ilustración 30.

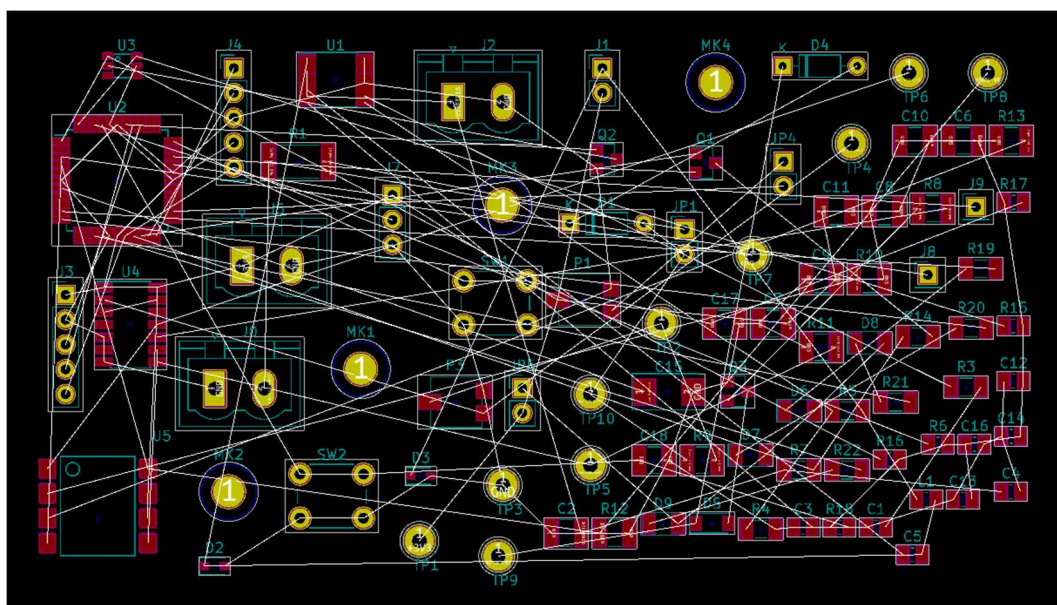


Ilustración 30: Editor PCB. Estado inicial

Las líneas blancas indican las conexiones eléctricas. El usuario debe separar los componentes a su gusto y poner pistas (es el nombre que reciben los cables en las PCB) conectando los componentes. Se debe tener especial cuidado con el ancho de las pistas; es un parámetro muy importante a tener en cuenta en función de la intensidad que circula por ellas. KiCAD contiene herramientas que ayudan al usuario a calcular el ancho de pista que necesita.

También hay que tener cuidado con la distancia entre pistas, aunque esto solo resulta relevante cuando la PCB maneja tensiones muy altas, ya que de estar muy cerca, podrían saltar arcos entre las pistas.

Tenido todo esto en cuenta, se puede hacer la tarjeta de muchas maneras, una de ellas se muestra en la Ilustración 31.

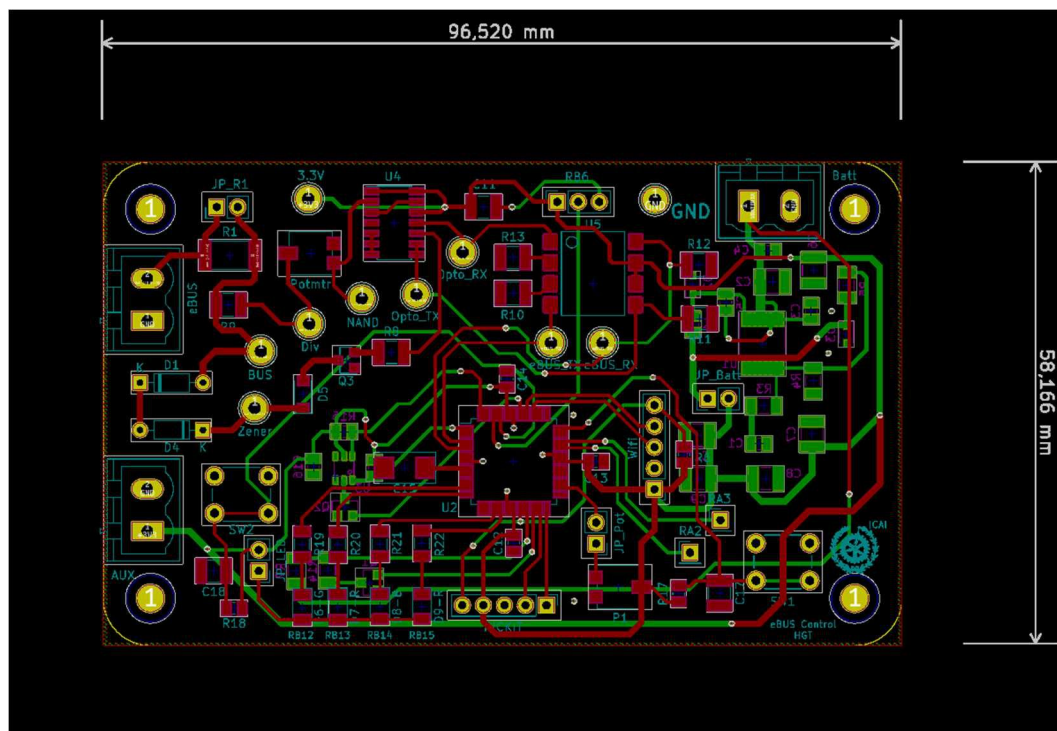


Ilustración 31: Editor PCB. Estado final

Las pistas y componentes de color rojo están situados en la cara de arriba, mientras que los verdes están en la capa de abajo. Los elementos amarillos, son taladros que atraviesan la placa, para componente THT, conectores y agujeros de montaje.

También se recomienda hacer un buen uso de la capa Silkscreen, que es la última capa que se da a las PCB. Es una capa de pintura, muy útil para reconocer los componentes en la placa e incluso para colocar logos o dibujos.

Esta tarjeta, como se muestra en la imagen mide 9.6 x 5.8 cm.

5.3.4 ARCHIVOS DE FABRICACIÓN

El último paso, consiste en crear los archivos de fabricación de la PCB, que serán enviados a la industria que se encargue de su fabricación. Es muy simple de hacer desde un menú de KiCAD dentro del editor de PCB (Ilustración 32).

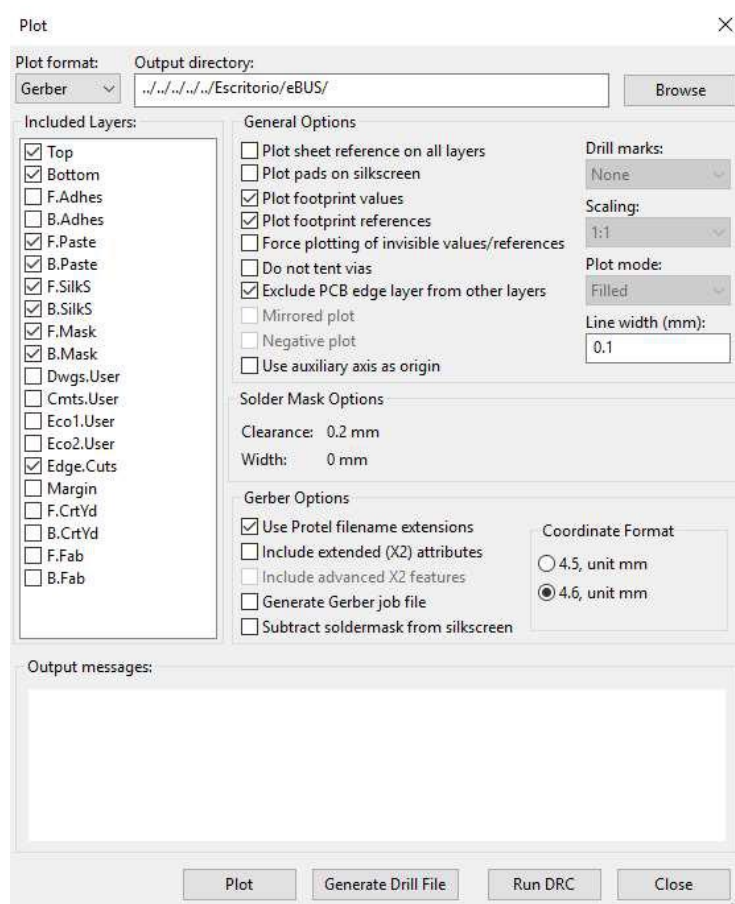


Ilustración 32: Archivos de fabricación

Desde este menú, el usuario elige las capas que quiere generar.

La extensión oficial de fabricación de PCB (no sólo en KiCAD) es .gerber. Sin embargo, depende mucho de cada industria, pues algunas piden los archivos en las extensiones antiguas.

Normalmente los operarios comprueban las capas en busca de errores antes de mandar el proyecto a la fase de fabricación.

5.4 PRUEBAS DE FUNCIONAMIENTO

Cuando la PCB ha llegado de la fábrica (Ilustración 33) se han soldado del resto de los componentes, quedando finalmente como en la Ilustración 20.

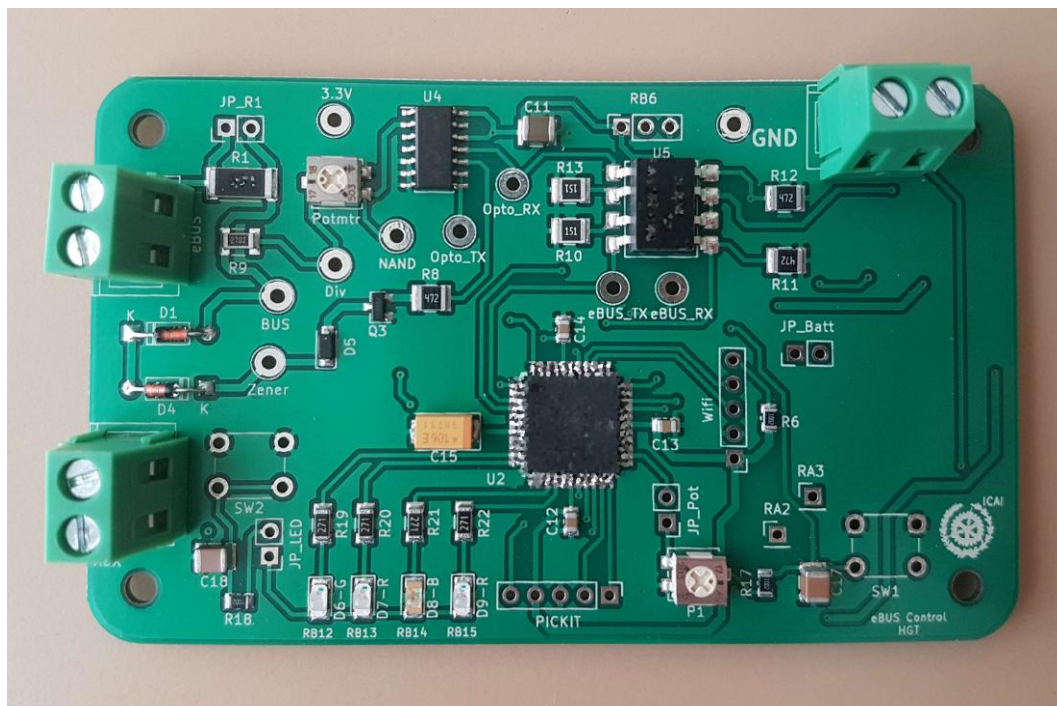


Ilustración 33: PCB de fábrica

A continuación, se han comprobado los módulos en busca de posibles errores de funcionamiento.

5.4.1 MICROCONTROLADOR

Para comprobar las UART del MIC, se ha cargado un programa a través del PICKit que permite enviar por UART la trama 0xAA00, tal como se muestra en la Ilustración 34.

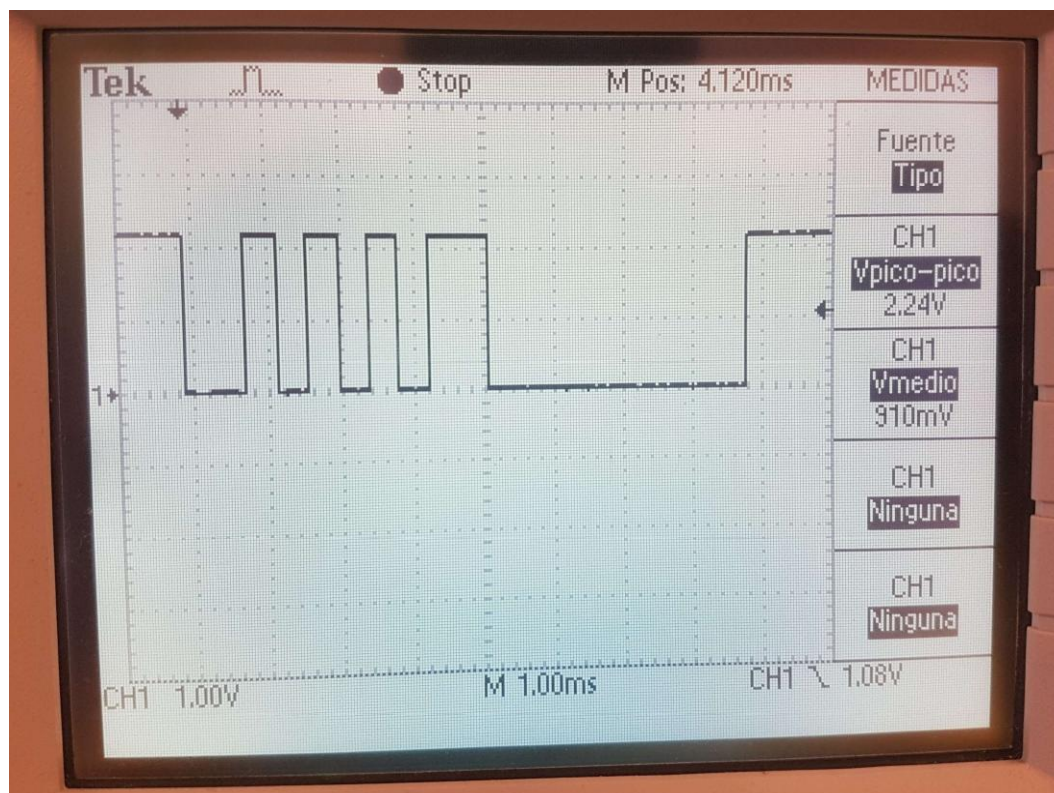


Ilustración 34: Prueba de transmisión UART

La UART en reposo está a nivel alto (1 lógico). Para el envío de la trama, UART pone un bit de start a 0, seguido de 1 byte de información (01010101, 0xAA del revés) y finalizando con un bit de stop a 1 para continuar de la misma forma con el resto de la información. Como se ve en la imagen, es una transmisión estable a las tensiones propias de un micro.

Por otro lado, se ha comprobado el funcionamiento de pequeños componentes como los diodos LED, con resultados satisfactorios.

5.4.2 ADAPTADOR eBUS

Con el mismo programa que en el apartado anterior, se ha medido la tensión en el conector de salida eBus del adaptador, siguiendo los pasos explicados en el Capítulo 4. En la Ilustración 35 se ven los resultados.

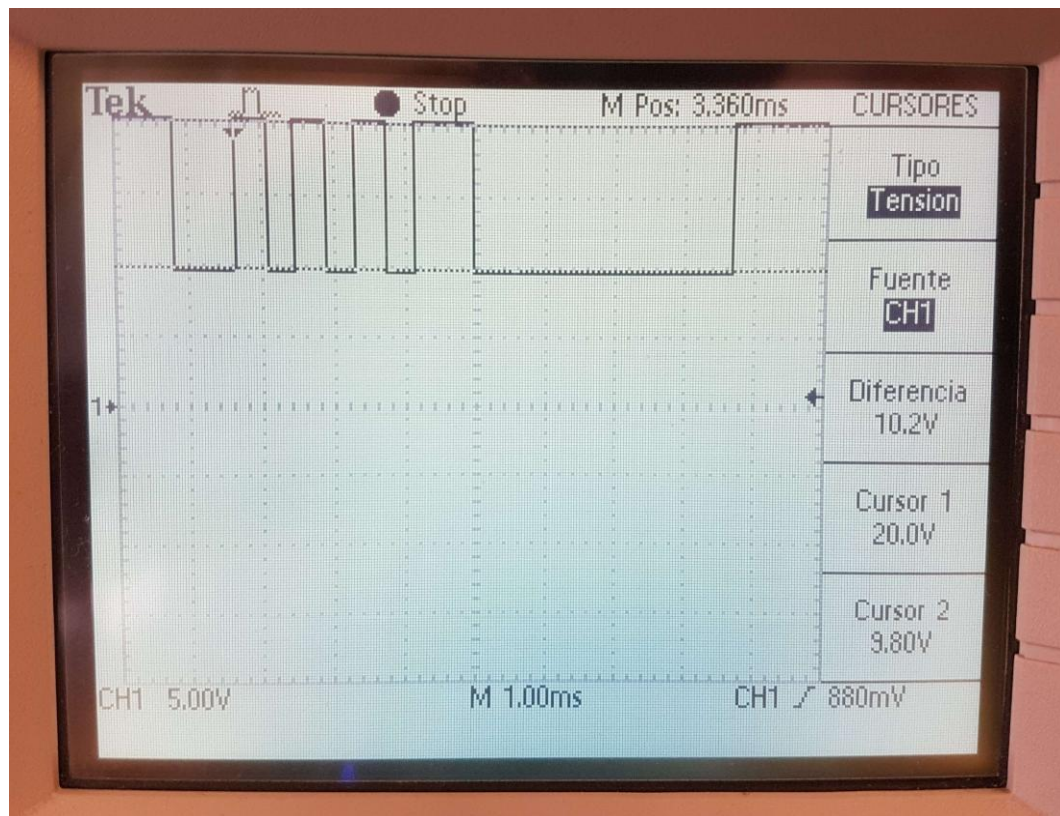


Ilustración 35: Prueba de transmisión adaptador

Como se observa, la tensión del adaptador es igual de estable. La señal es idéntica a la original, con los valores de tensión necesarios para una transmisión válida en protocolo eBus (20V para el 1 lógico, 9.80V para el 0).

5.4.2.1 Problema con el transistor

Se ha descubierto, que el transistor utilizado en el adaptador, un MMBT3904 de montaje en superficie estaba mal montado por un error en el diseño.

La huella utilizada y el transistor coincidían, pero con el colector y emisor cambiados. Como resultado, el adaptador no ha funcionado hasta que no se ha quitado ese componente y se ha soldado en su lugar el transistor que se utilizó en el circuito original del Capítulo 4, un 2N3904 THT. En la Ilustración 36 se muestra el resultado de dicha sustitución.

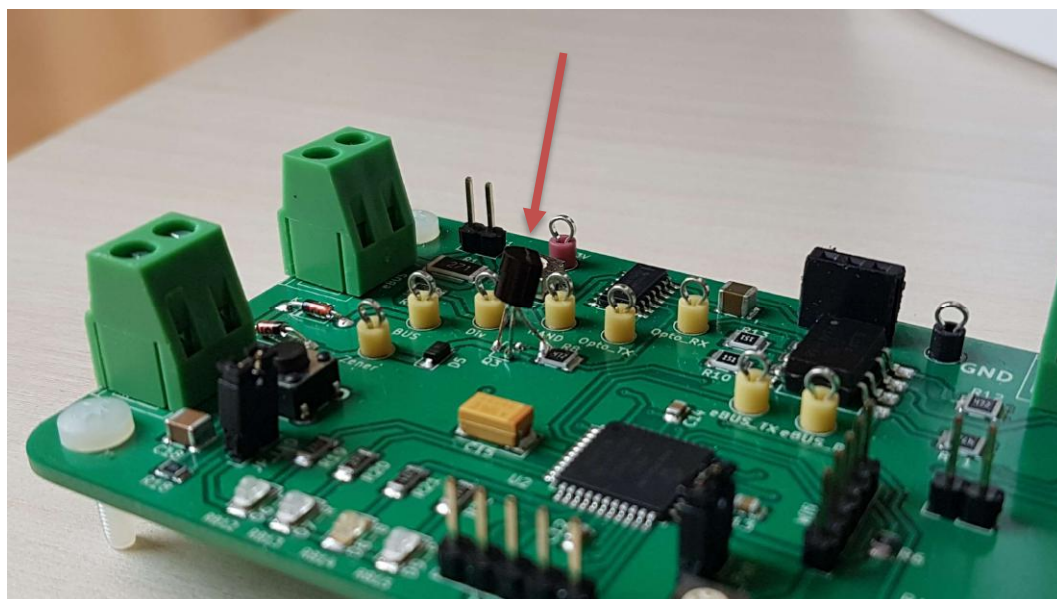


Ilustración 36: Transistor 2N3904 sustituyendo al MMBT3904

5.4.3 REGULADOR

El funcionamiento del regulador se ha probado con la fuente de alimentación del laboratorio, variando su tensión desde los 4V a los 14V. Se ha notado que no realiza la conversión de acuerdo al ajuste hecho en el diseño, con el que debería dar una salida de 3.3V. Aparentemente la tensión de salida es una simple copia de la entrada. Tras una investigación, se ha sabido que la industria encargada de la fabricación de la PCB ha facilitado una versión diferente del mismo componente, con lo que, aunque el diseño mostrado en 5.2.3 es perfectamente válido, se ha decidido modificar el diseño para este nuevo chip. Basta con retirar los diodos y un par de resistencias, como se muestra más adelante en el punto 7.5.

Una vez corregido el diseño, se obtienen una regulación de tensión esperada, tal como se muestra en la Ilustración 37.

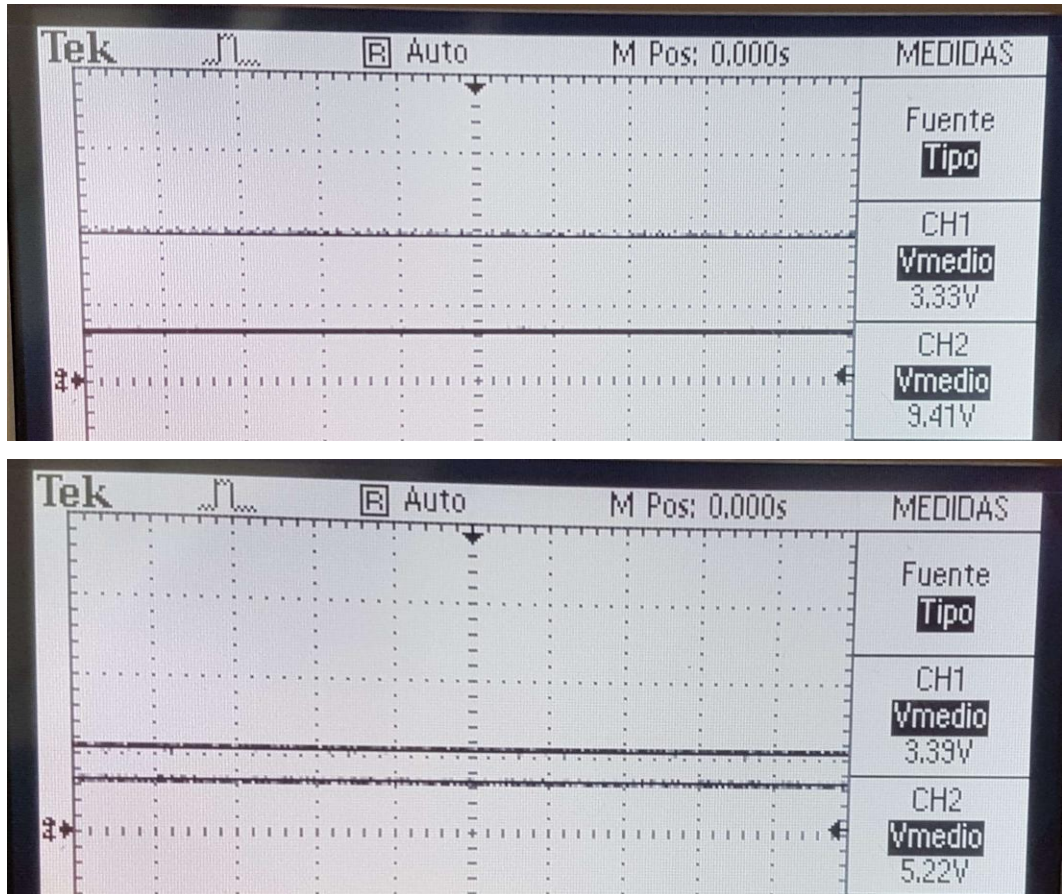


Ilustración 37: Regulación de tensión

Como se puede ver, a distintas tensiones de entrada (9.41V y 5.22) el regulador fija su salida a unos 3.35V. Se ha comprobado, que variando la tensión de entrada desde 4V a 14V, la salida se mantiene constante en el rango 3.31V - 3.40V1.

Capítulo 6 ALGORITMO

En este capítulo se va a explicar el funcionamiento de los programas que tienen lugar en el programa principal (main), sin entrar en detalle en las capas inferiores del protocolo eBus. El programa que se ha utilizado para la escritura del código es el freeware MPLAB X IDE v4.05 con el compilador XC16.

Los algoritmos mostrados a continuación funcionan de forma conjunta, si bien por claridad se explica cada uno de forma independiente.

6.1 COMUNICACIÓN MAESTRO-ESCLAVO

En el Capítulo 3 ya se explicó con detalle cómo se lleva a cabo el intercambio de información entre maestro y esclavo. En la Ilustración 38 se muestra un diagrama del funcionamiento de la comunicación entre maestro y esclavo. En el diagrama no se muestra el algoritmo del arbitraje, sino que se detalla de forma independiente en el siguiente apartado.

En el diagrama, SYN es el símbolo de sincronización y ACK es el acuse o acknowledgement.

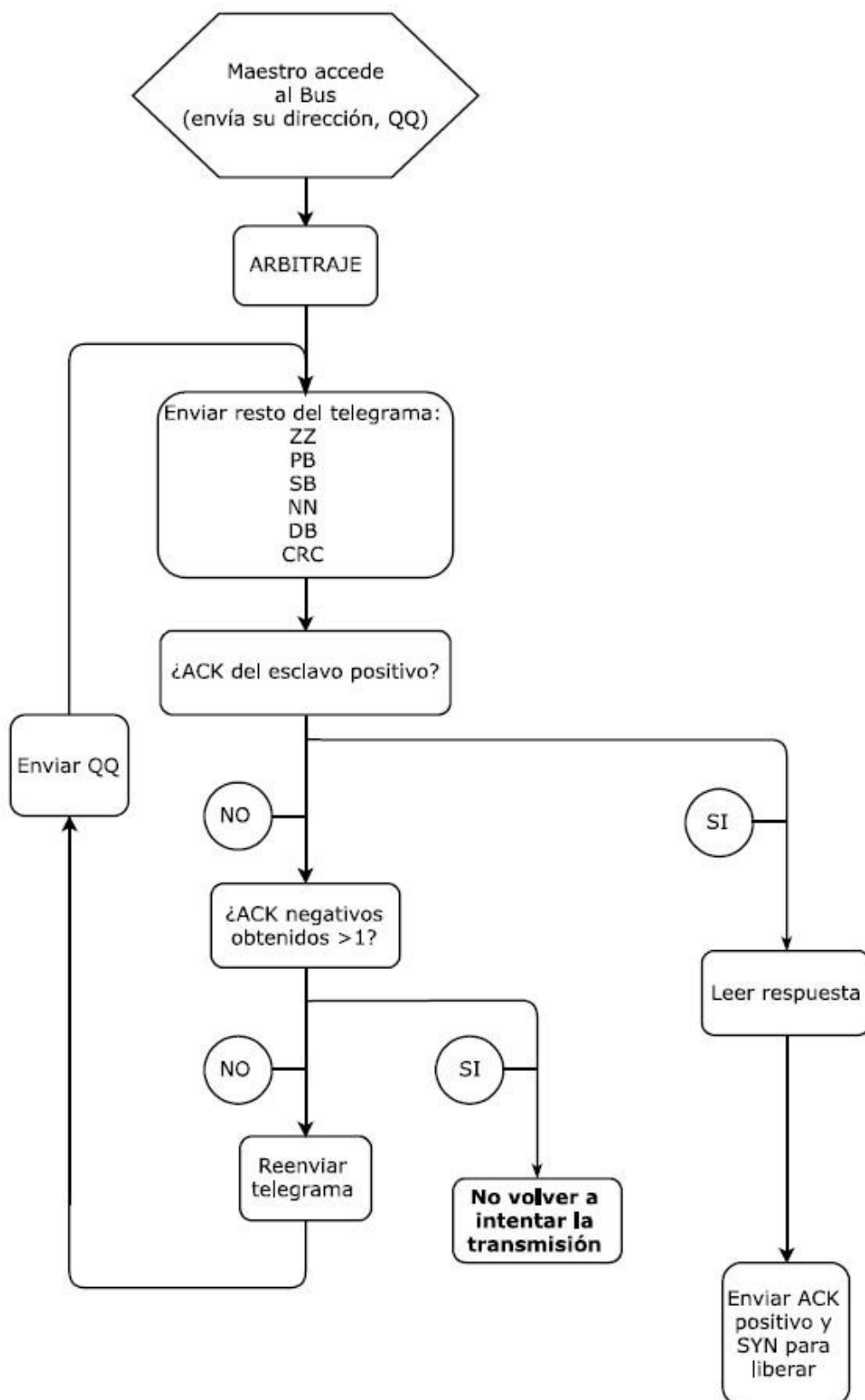


Ilustración 38: Diagrama de flujo. Comunicación Maestro-Eslavo

6.2 ARBITRAJE

El arbitraje (3.3.4 Arbitraje) tiene lugar cuando varios maestros tratan de enviar sus telegramas de forma simultánea. En la Ilustración 39 se muestra cómo actúa el maestro durante el arbitraje.

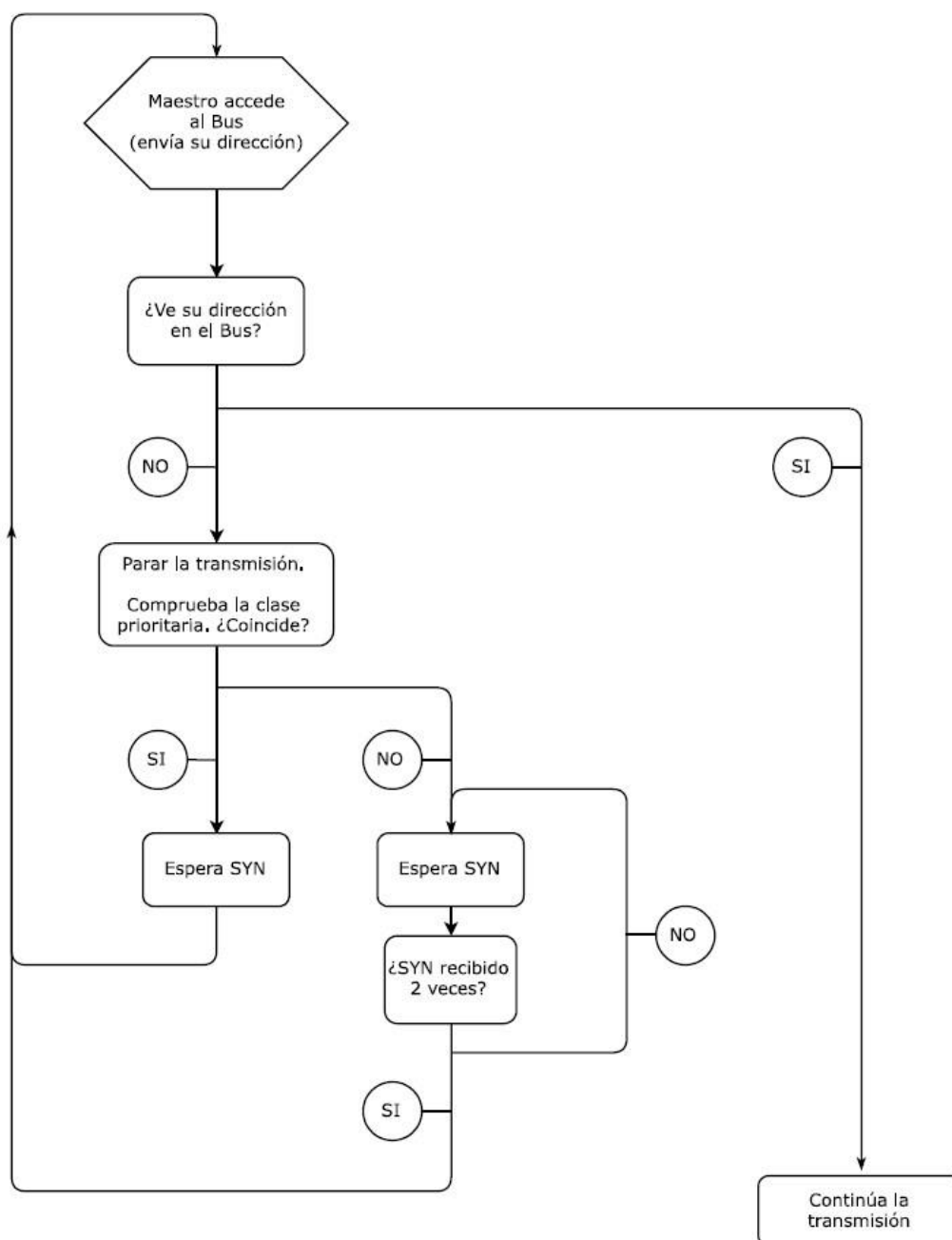


Ilustración 39: Diagrama de flujo. Arbitraje

6.3 GENERADOR DE SÍMBOLOS DE SINCRONIZACIÓN

Como se explicó en el 3.3.5 Generador Automático de Símbolos de Sincronización, el bus puede quedarse paralizado por inactividad. El generador de símbolos de sincronización lee los cambios en el bus y, en caso de estar inactivo durante un determinado tiempo (que se fijará en 35ms), procede a enviar un símbolo al bus para que los participantes que estaban esperando puedan acceder al bus.

En la Ilustración 40 se muestra cómo funciona el algoritmo de un generador. El timer del generador funciona por interrupciones, al igual que la recepción en la UART, para no quitarle tiempo al programa principal.

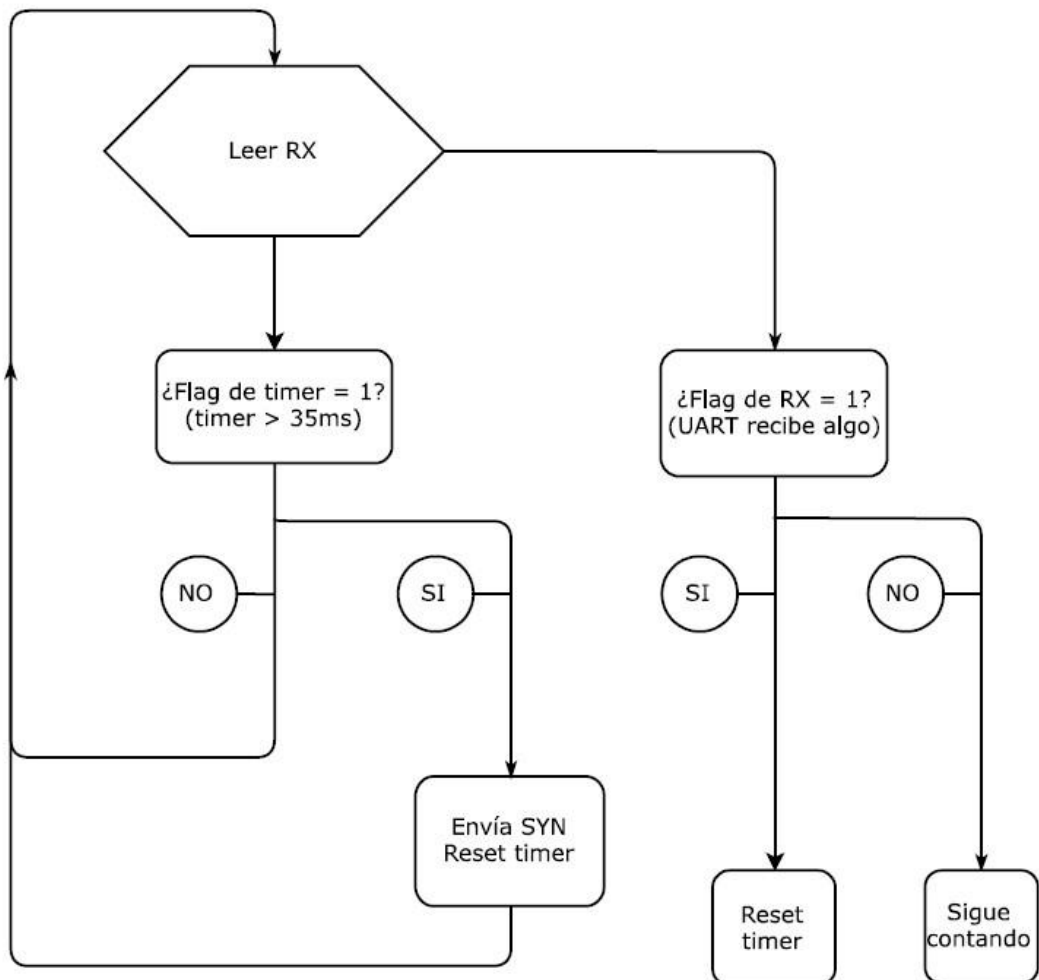


Ilustración 40: Diagrama de flujo. Generador de símbolos

Capítulo 7 RESULTADOS

En este capítulo se muestran los resultados de distintas pruebas realizadas con la tarjeta controladora. Al no disponer de una caldera real con la que poder experimentar, las pruebas realizadas se han limitado a la utilización de varias tarjetas controladoras individualmente y entre sí. Se ha probado cada módulo por separado, como se ve en los siguientes apartados.

Por otro lado, se comentan también en este capítulo las correcciones de los fallos de diseño encontrados en la tarjeta controladora.

7.1 COMUNICACIÓN MAESTRO-ESCLAVO

En la Ilustración 41 se muestra un telegrama completo enviado desde la tarjeta controladora del Capítulo 5. Este telegrama (0x 10 08 B5 04 01 01 3C) debería dar como respuesta el Modo De Operación de la caldera como podría ser: on, off, auto, eco, heating, night, etc., dependiendo del telegrama enviado por el esclavo.

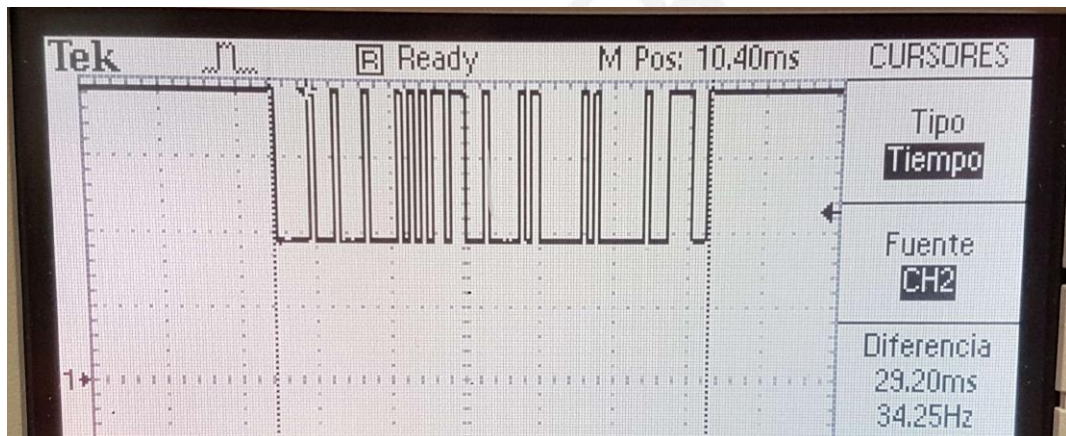


Ilustración 41: Captura de un telegrama enviado en el osciloscopio

El tiempo total de la transmisión es 29ms y hay 69 bits en el telegrama (falta el último bit de stop en la medida realizada). Según esto, la velocidad de transmisión de este telegrama es: $f = \left(\frac{29.2ms}{69}\right)^{-1} = 2.363KHz = 2363Hz$. Esto indica que el telegrama está bien enviado, a la velocidad que marca el protocolo, respetando los bits de start y stop y sin usar otros bits extra como el de paridad.

Por otro lado, se ha comprobado el tiempo de bit de este telegrama (véase Ilustración 42) ampliando la escala temporal del osciloscopio y, como se puede ver en la Ilustración 42, teniendo un t_{bit} de 416 μs , la velocidad de transmisión de las comunicaciones se está dando a 2403Hz que equivalen en este caso a 2403bps. Esta medida es bastante más precisa que la hecha anteriormente. Se verifica una vez más que se cumple el estándar de eBus, por lo que se espera que un dispositivo eBus pueda enviar un telegrama como respuesta a esta transmisión.

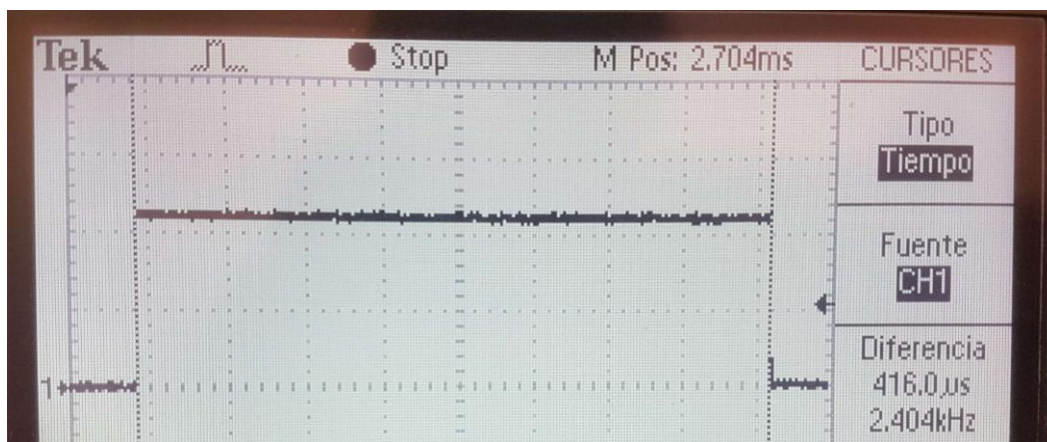


Ilustración 42: Tiempo de bit

Desafortunadamente, sin disponer de una caldera, no se ha podido comprobar que respuesta se obtiene de ésta. Sin embargo, se ha utilizado otra tarjeta para responder con un telegrama de la misma forma, como se muestra en la Ilustración 43.

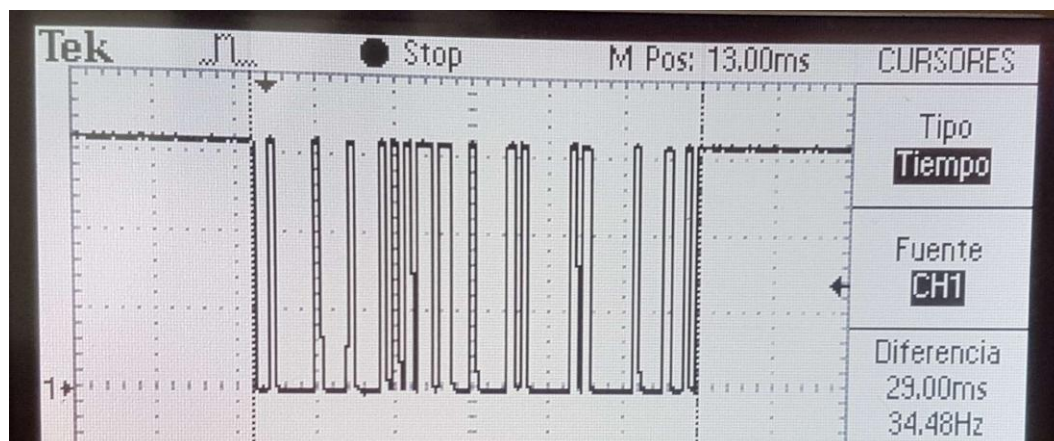


Ilustración 43: Captura de un telegrama recibido en el osciloscopio

El tiempo de transmisión es el mismo que en el telegrama enviado. Sin embargo, ambos telegramas se diferencian en que, en este último, la dirección del origen (primera palabra) es diferente, por lo que el CRC (última palabra) es consecuentemente distinto.

Además, se puede ver que, en la recepción, la señal aparece un poco deformada. Esto es por el adaptador de tensión y el paso de la señal por el optoacoplador que podría tener las resistencias mal ajustadas. De cualquier forma, se ha comprobado que el MIC es capaz de interpretar el telegrama correctamente.

7.2 ARBITRAJE

La programación del algoritmo para el arbitraje no ha sido un éxito rotundo. Para una trama simple, el MIC es capaz de interpretar si la primera palabra de su telegrama (su dirección) coincide con lo que ocurre en el bus. Si la coincidencia es positiva, envía el resto del telegrama sin un retardo aparente (igual que como muestra la Ilustración 41) ya que el procesamiento del micro es muchísimo más rápido que la velocidad de transmisión.

Por el contrario, si lo que se “ve” en el bus no coincide con la dirección del maestro, éste detiene la transmisión y espera a 1 o 2 símbolos de sincronización en función de la clase prioritaria. Es aquí donde surge el problema, pues no se ha logrado con suficiente éxito gestionar los datos que se están recibiendo en la UART constantemente (con el generador de símbolos de sincronización activado) para poder realizar el arbitraje de forma correcta.

7.3 GENERADOR DE SÍMBOLOS DE SINCRONIZACIÓN

Se ha programado un algoritmo relativamente simple utilizando un timer. Si el bus está en reposo durante más de 35ms, el generador envía un símbolo de sincronización (0xAA). En la Ilustración 44 se puede ver, que, al no haber cambios en el bus, cada 35ms se envía un símbolo. Aunque el tiempo entre símbolos puede parecer muy pequeño, esto permite que las comunicaciones sean más rápidas (ya que el símbolo permite a los maestros el acceso al bus) y además es el tiempo que se especifica en las características del protocolo.

En la Ilustración 45 se ha capturado un telegrama entre símbolos de sincronización. Como se puede ver en la imagen, desde que termina la trama hasta que llega el siguiente símbolo transcurren los 35ms, demostrando que el generador respeta las transmisiones de telegramas en el bus, sin provocar colisiones de datos.

La calidad tan baja de esta imagen se debe a que, como cada 35ms (como máximo) hay cambios en el bus, no se puede utilizar el disparo del osciloscopio para congelar la imagen. Esta captura se ha realizado congelando manualmente la pantalla del osciloscopio, perdiendo por lo tanto mucha calidad.

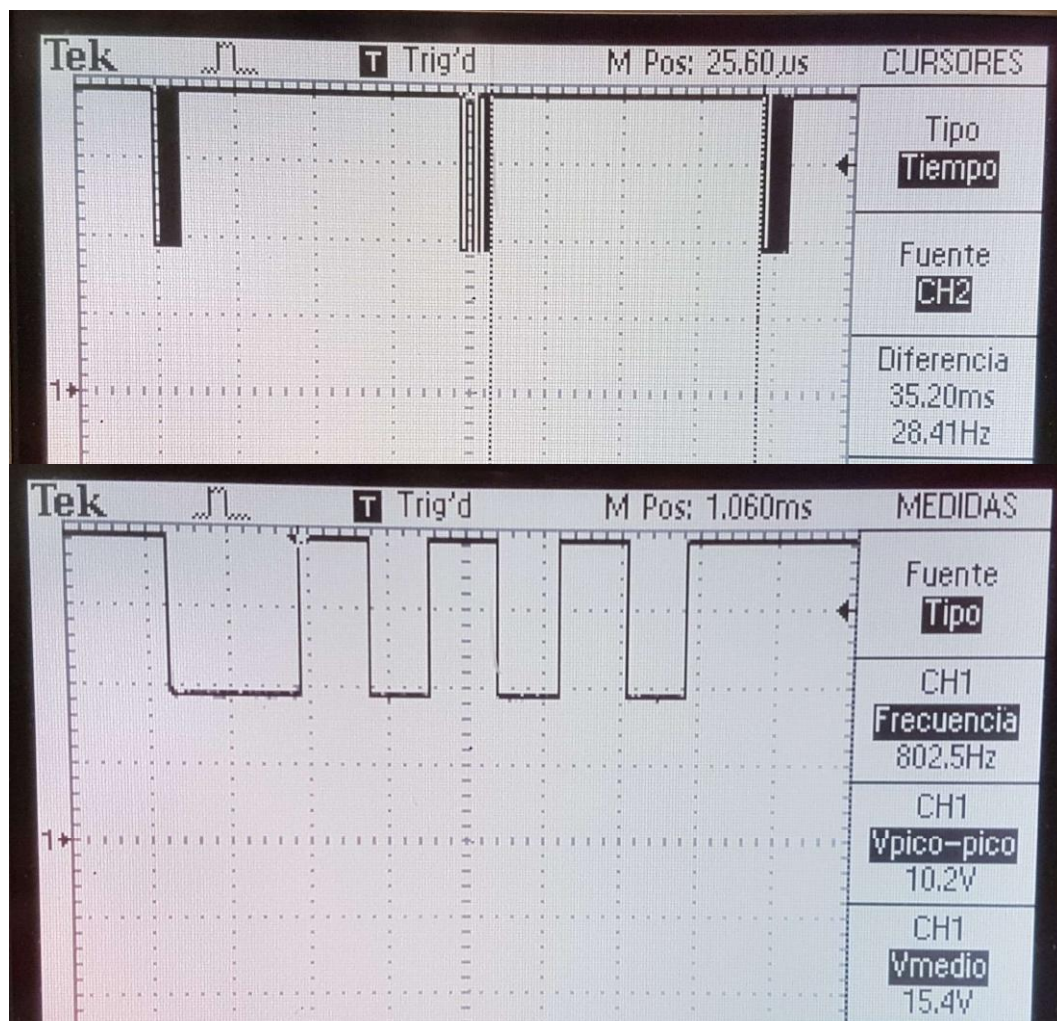


Ilustración 44: Símbolos de sincronización

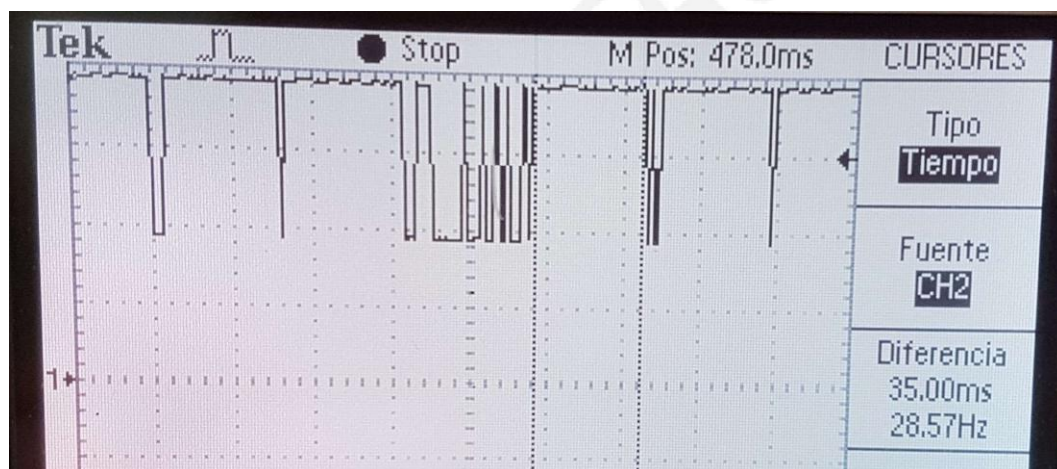


Ilustración 45: Telegrama entre símbolos de sincronización

7.4 *SENSORES DE TEMPERATURA*

7.4.1 SENSOR SUPERFICIAL

Como ya se mencionó, el sensor de superficie funciona mediante protocolo I2C. Se ha programado el PIC para que cuando se utilice el pulsador 2 (RB5), el sensor devuelva la temperatura de la placa. En la Ilustración 46 se muestran las temperaturas recogidas por el sensor en reposo y tras calentar el sensor deliberadamente con un dedo para provocar un cambio de medida.

```

ant2 = act2;
act2 = PORTB & 32;
if ((ant2 != act2) && act2==0) {
    uint8_t ok2 = readTemp(temp2);
    if (ok2==0){
        test2=1;
    }
}

ant2 = act2;
act2 = PORTB & 32;
if ((ant2 != act2) && act2==0) {
    uint8_t ok2 = readTemp(temp2);
    if (ok2==0){
        test2=1;
    }
}

```

Address = 0x187A, temp2 = 0x0000
Address = 0x0, *temp2 = 25.962402

Address = 0x187A, temp2 = 0x0000
Address = 0x0, *temp2 = 30.402832

Ilustración 46: Capturas de temperatura por sensor superficial

En reposo el sensor midió 25.96°C, lo cual es razonable teniendo en cuenta que el laboratorio donde se encontraba estaba a unos 24°C. Tras calentarlo llegó a 30.40°C que también es una medida esperable.

7.4.2 SENSOR EXTERNO

En sensor externo utilizado funciona con el protocolo 1-Wire como se explicó en el epígrafe 5.2.6.4. Tras un infructuoso intento de hacer funcionar el sensor mediante un algoritmo, se ha detectado un error de diseño de hardware: el puerto no tenía conectada una resistencia de pull-up. El sensor, que funciona “tirando” a 0V la tensión del bus para enviar los datos, no podía hacerlo dado que el bus en reposo siempre estaba a 0V. Una vez conectada una resistencia de 4.7K Ω , se ha podido solventar el problema.

En la Ilustración 47 se pueden ver los valores que devuelve el sensor de temperatura en dos situaciones.

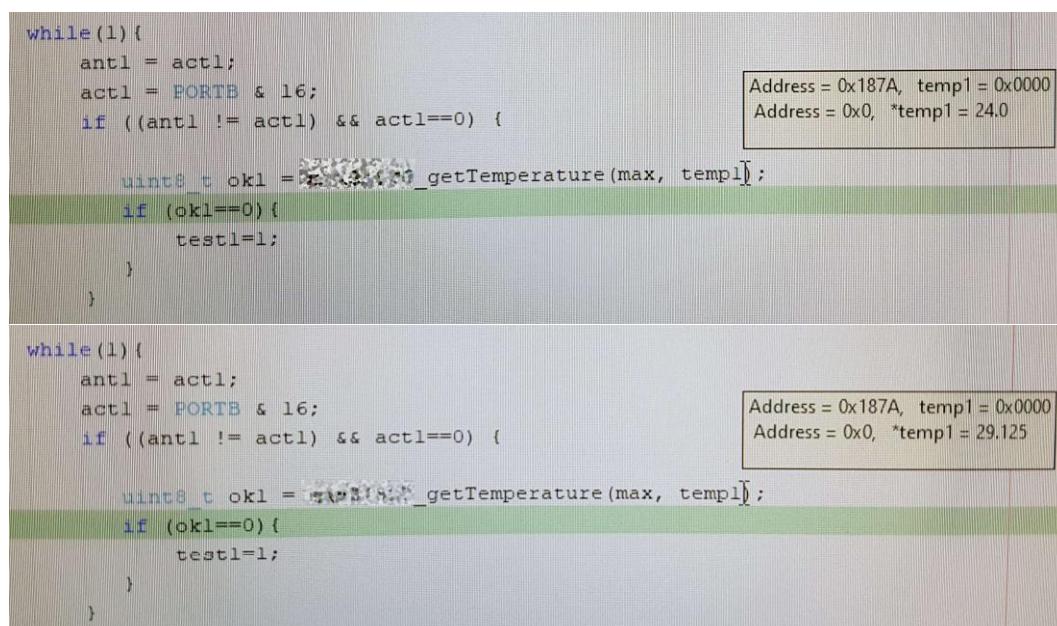


Ilustración 47: Capturas de temperatura por sensor externo

En la imagen superior, el sensor ha registrado una temperatura ambiente de 24°C en el laboratorio con el aire acondicionado encendido. En la imagen inferior, tras calentar el sensor con los dedos, ha devuelto una lectura de 29.13°C. Ambas medidas son perfectamente razonables, por lo que se asume su correcto funcionamiento.

7.5 FALLOS DE DISEÑO CORREGIDOS

A lo largo del capítulo anterior se han detectado errores en el diseño de la tarjeta que se deben solventar. Probablemente el error más grande de diseño ha sido el problema con el regulador de tensión. Aunque el diseño anterior era funcional, se ha decidido utilizar un diseño alternativo para el componente que se ha montado en la tarjeta. En la Ilustración 48 se muestra el nuevo diseño. La diferencia entre los dos componentes es simple. El diseño original era para un regulador con tensión de salida ajustable, mientras que el regulador que se tiene ajusta directamente su salida a 3.3V, por lo que se puede prescindir de varios componentes.

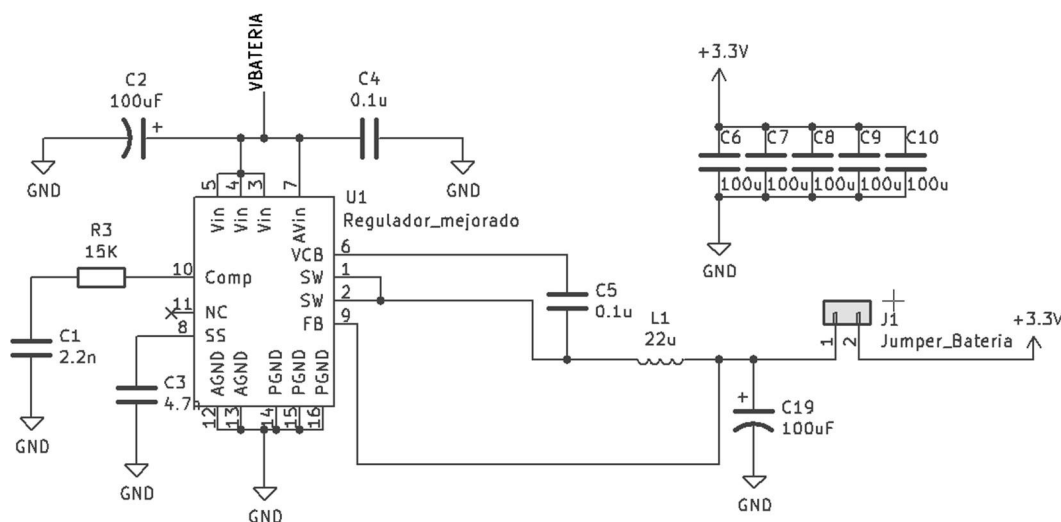


Ilustración 48: Diseño renovado del regulador

Seguidamente, se han corregido los siguientes problemas:

- La huella del transistor del adaptador eBus (5.4.2.1). Se ha cambiado el componente a uno THT en lugar de SMD con una huella acorde.
- El jumper mal colocado del potenciómetro (5.2.1) que no tiene utilidad alguna ya que siempre está consumiendo intensidad (0.33mA).
- Resistencia pull-up para el sensor de temperatura externo.
- Cambio de la inductancia del adaptador eBus por otra con mejores características en cuanto a intensidad máxima. Se sospecha que la actual puede no soportar las intensidades del regulador a máxima potencia.

7.5.1 DISEÑO FINAL DE LA TARJETA CONTROLADORA

En última instancia, se ha rediseñado la tarjeta controladora en una segunda versión más definitiva y totalmente funcional. Se han eliminado pequeños componentes innecesarios (algunos testpoint y el conector de alimentación auxiliar), se han implementado las correcciones de diseño, añadido acceso a más puertos analógicos y digitales y se ha optimizado el espacio en la tarjeta, recolocando los componentes y reduciendo el tamaño de la PCB un 52% respecto a la primera versión. En la Ilustración 49 puede verse el nuevo diseño de la controladora junto con su renderizado 3D.

Esta versión de la controladora no debe considerarse definitiva por dos motivos: primero, porque hay elementos que pueden ser rediseñados como se explicará en el Capítulo 9; segundo, porque en una próxima versión comercial, hay elementos que tiene esta tarjeta que no tendría sentido conservar.

Entre algunos de los componentes que podrían ser rediseñados o removidos se podrían citar:

- Pulsadores. Aunque son de gran utilidad, en una versión comercial podrían no serlo tanto, aunque siempre es recomendable tener un Reset del sistema.
- Testpoint. Aún queda un testpoint de la recepción del bus, pues como se explicó en el Capítulo 4, se necesita ajustar un potenciómetro para poder recibir adecuadamente los telegramas. Este testpoint todavía es crítico para hacer dichas comprobaciones.
- Potenciómetro. Hay un segundo potenciómetro sin una funcionalidad precisa, aunque puede ser útil para la luminosidad de los diodos LED.
- Diodos LED. Actualmente la tarjeta cuenta con 4 diodos: 2 rojos, 1 azul y 1 verde. Se podría decidir prescindir de alguno de ellos.
- Puertos analógicos y digitales. En esta versión se ha decidido dar acceso a 3 puertos analógicos y 3 digitales extra en caso de querer conectar más elementos a la placa para hacer pruebas de algún tipo. En una versión comercial se debería elegir cuales se van a usar y retirar el resto.

Con estos nuevos cambios, se espera que, en la versión comercial de la tarjeta, ésta pueda reducir incluso más su tamaño.

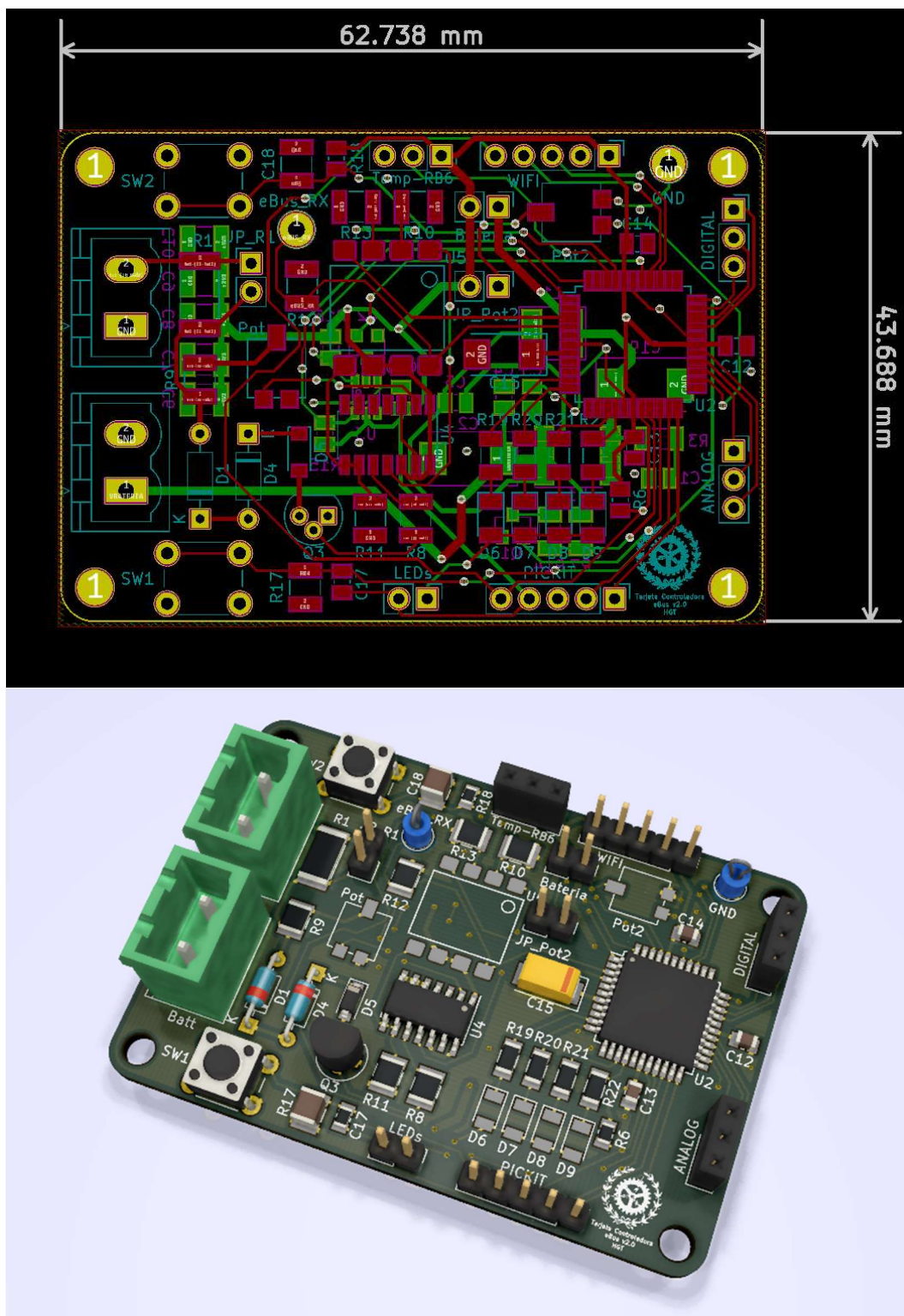


Ilustración 49: Nueva versión de la tarjeta controladora

Capítulo 8 CONCLUSIONES

El objeto de este proyecto ha sido construir un sistema que controle y monitorice una caldera de gas doméstica y está orientado a optimizar y reducir el consumo energético en el hogar. En el trabajo desarrollado, no se ha completado el sistema entero, pero sí se han logrado grandes objetivos en lo que se podría considerar un sistema parcial funcional. Algunos de los logros son:

- ➔ Se ha diseñado, testado e implementado un circuito que permite adaptar los niveles de tensión de un microprocesador a los del protocolo eBus.
- ➔ Se ha diseñado una tarjeta controladora que permite potencialmente la comunicación entre el usuario y la caldera. Se han comprobado todos sus módulos con resultados positivos.
- ➔ Se han programado algoritmos para poder establecer comunicaciones con la caldera mediante el protocolo eBus.

Por otro lado, caben destacar ciertas complicaciones que han surgido durante el desarrollo del proyecto, a saber:

- ➔ El protocolo eBus es de origen alemán y no está tan extendido como podrían ser I2C o Modbus. Como consecuencia, la recolección de información sobre este protocolo ha resultado complicada. Además, al ser un protocolo tan poco utilizado, no existe una comunidad activa que lo respalde.
- ➔ La información técnica de la caldera no es de dominio público. Esta información se refiere a los espacios de la memoria interna de las calderas donde se guardan los parámetros a los que precisamente se quiere acceder o

modificar. Se ha preguntado a la marca, la cual ha aclarado que dicha información es propiedad intelectual y no la pueden facilitar. Es por esto, que la única información que se conoce es de usuarios que han espiado el bus de comunicaciones entre la caldera y un maestro comercial, y han recopilado la información en un documento. El problema es que esta información, aparte de ser incompleta y no oficial, está obsoleta y no se sabe si con una caldera real de última generación podría llegar a funcionar.

- ➔ No se ha tenido acceso físico a una caldera ecoTEC Plus de Vaillant por lo que el desarrollo del proyecto ha quedado paralizado y no se han podido hacer ningún tipo de pruebas reales de comunicación.

Capítulo 9 FUTUROS DESARROLLOS

Durante el desarrollo de este proyecto se han encontrado errores de diseño que han sido solventados. Sin embargo, aunque se tenga una tarjeta controladora totalmente funcional, el proyecto es incompleto, y hay ideas y mejoras que pueden implementarse en el futuro. Entre ellas se podrían incluir:

- ➔ Crear una caja mediante impresión 3D utilizando software de diseño (como podría ser SolidWorks) donde se pueda meter la tarjeta controladora con baterías, para que formen un sistema más compacto y de aspecto profesional, orientado a la venta al público.
- ➔ Relacionado con el punto anterior, utilizar diodos LED THT (through-hole) en lugar de SMD para que puedan sobresalir levemente de la caja.
- ➔ Programar las comunicaciones entre la controladora y el módulo Wifi a través de una de las UART del MIC.
- ➔ Reprogramar el algoritmo del arbitraje para que tenga una funcionalidad total respetando totalmente los estándares de eBus.
- ➔ Utilizar un maestro comercial y una caldera real para espiar las comunicaciones del bus. De esta forma se podría recopilar información sobre la estructura de los telegramas en un documento actualizado.
- ➔ Desarrollar una app en Android e iOS para poder gestionar la caldera remotamente.
- ➔ Rediseñar el adaptador eBus para poder eliminar el potenciómetro. En una versión comercial no sería óptimo que el usuario tuviera que configurar la

controladora para su adecuado funcionamiento. Mediante un preciso divisor de tensión podría solventarse el problema. Para ello hay que saber exactamente la tensión que tiene el bus y, por ello, para el diseño de este divisor resistivo hay que tener acceso físico a una caldera.

- ➔ Reajustar las resistencias del optoacoplador. Aunque el microcontrolador es capaz de procesar correctamente las señales que éste le comunica, podría obtenerse una tensión un poco más limpia con un ajuste más fino para evitar posibles errores en la recepción de datos.

Capítulo 10 BIBLIOGRAFÍA

- [1] «Soluciones Integrales,» [En línea]. Available:
<https://www.solucionesintegralesendesa.com/blog/calderas/evolucion-calderas/>.
- [2] «Aircal,» [En línea]. Available:
<https://www.aircal.es/instalaciones/calderas/tipos-de-calderas/>.
- [3] Vaillant. [En línea]. Available:
<https://www.vaillant.es/usuarios/productos/ecotec-plus-25344.html>.
- [4] C. Rodríguez-Morcillo, S. Alexandres y J. Matanza, «Comunicaciones Industriales,» 2017.
- [5] F. M. Martinez, Gestión de energía en una casa inteligente, 2013.
- [6] R. d. M. Peirotén, Sistema domótico para una casa inteligente.
- [7] Wikipedia. [En línea]. Available:
[https://en.wikipedia.org/wiki/EBUS_\(serial_buses\)](https://en.wikipedia.org/wiki/EBUS_(serial_buses)).
- [8] e. I. Group. [En línea]. Available: http://ebus-wiki.org/lib/exe/fetch.php/ebus/spec_prot_12_v1_3_1_e.pdf.
- [9] e. I. Group. [En línea]. Available: http://ebus-wiki.org/lib/exe/fetch.php/ebus/vaillant_ebus_v0.6.0.pdf.

Parte II ESTUDIO

ECONÓMICO

Capítulo 1 ESTUDIO ECONÓMICO

El proyecto expuesto permite la potencial creación de un sistema completo que controle y monitorice una caldera de gas, con el objetivo de optimizar el consumo y controlar mejor los gastos en el hogar. Este proyecto no es el sistema completo, como ya se ha explicado en otros capítulos. Un sistema completo constaría además de un módulo de comunicaciones Wifi y una aplicación en un terminal móvil, aparte de la programación necesaria para que se puedan comunicar adecuadamente.

Aunque el proyecto desarrollado aún no tiene una orientación a la venta al público, próximamente se podrían realizar los ajustes necesarios para crear una versión comercial.

Una vez desarrollado el sistema completo, primeramente, se debería tratar de promocionar entre un público que pueda estar interesado en la optimización energética del hogar, que sean aficionados a las nuevas tecnologías y estén cómodos con el uso de aplicaciones móviles. Gente joven, con mucho interés en el ahorro de sus facturas podrían ser el objetivo óptimo. En esta primera fase, el precio del producto sería relativamente caro, ya que, al no tener grandes ventas, no se producirían unidades en masa. Se estima que el coste de una unidad podría llegar a los 300€ en la primera fase, antes de abaratar costes.

Cuando debido a la utilidad del sistema se consiga la atención de otros sectores del público, se podría bien incrementar las funcionalidades del sistema para hacerlo aún más atractivo al público, o bien asociarse con una empresa que cuente con sistemas de domótica similares al desarrollado. Llegado este momento, se comenzaría a producir unidades en masa del sistema, con lo que el coste unitario del mismo disminuiría considerablemente, aumentando en consecuencia los beneficios. A medida que se hace popular se iría incrementando el precio, tratando de mejorar también las funcionalidades para conseguir mayor margen de beneficios.

En la Tabla 2 se muestra un gráfico con el crecimiento esperado del valor del sistema a lo largo de 5 años, mostrando la variación del coste y precio unitario y el beneficio obtenido como resta de ambos.

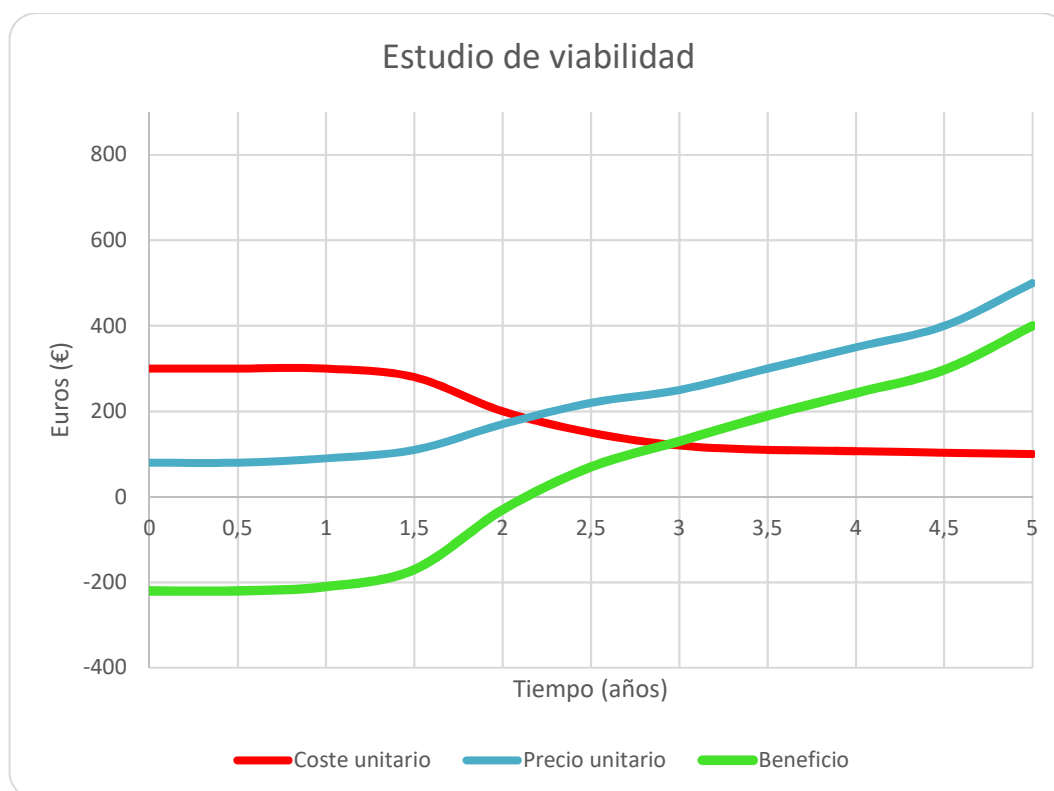


Tabla 2: Estudio de viabilidad

Con este crecimiento se esperarí­a obtener beneficios a partir de los 2 años.

Parte III PRESUPUESTO

Capítulo 1 MEDICIONES

En este capítulo se agrupan los elementos necesarios para la consecución del proyecto que implican una aportación económica, así como las cantidades requeridas y las horas trabajadas en función de la tarea realizada.

<i>Equipo y herramientas</i>			
MATERIAL	CANTIDAD	HORAS DE PROYECTO	HORAS DE USO AL AÑO
<i>Ordenador</i>	2	600	3000
<i>PCB Controladora</i>	3	300	700
<i>Osciloscopio</i>	1	150	400
<i>Polímetro digital</i>	1	20	70
<i>Soldador</i>	1	15	50
<i>Fuente de alimentación</i>	2	150	350
<i>Generador de señales</i>	1	10	100
<i>Protoboard</i>	2	150	400
<i>Componentes varios*</i>	10	130	300

Tabla 3: Relación de equipos y herramientas

*Los componentes varios, se refieren a elementos de laboratorio (sensores, diodos, resistencias, reguladores, etc. que se han utilizado en el proyecto).

<i>Software</i>		
PROGRAMA	HORAS DE PROYECTO	HORAS DE USO AL AÑO
<i>MPLAB X IDE v4.05</i>	300	700
<i>KiCAD r9359</i>	150	500
<i>AutoCAD 2017</i>	50	200
<i>Microsoft Office Word</i>	150	500
<i>Microsoft Office Excel</i>	10	150
<i>Microsoft Office Powerpoint</i>	20	200

Tabla 4: Relación de uso de software

<i>Mano de obra directa</i>	
TAREA	HORAS
<i>Búsqueda de información</i>	100
<i>Diseño de PCB controladora</i>	70
<i>Montaje de PCB</i>	10
<i>Programación</i>	250
<i>Pruebas de funcionamiento</i>	200
<i>Redacción de documentación</i>	150

Tabla 5: Relación de horas de mano de obra directa

Capítulo 2 PRECIO UNITARIO

En este capítulo se detallan los precios unitarios del material listado, así como de precio por hora de cada tarea realizada.

<i>Equipo y herramientas</i>	
MATERIAL	PRECIO (€/ud.)
<i>Ordenador</i>	600
<i>PCB Controladora</i>	150
<i>Osciloscopio</i>	400
<i>Polímetro digital</i>	120
<i>Soldador</i>	43
<i>Fuente de alimentación</i>	230
<i>Generador de señales</i>	570
<i>Protoboard</i>	15
<i>Componentes varios</i>	50

Tabla 6: Precio unitario de equipo y herramientas

<i>Software</i>	
PROGRAMA	Precio (€/licencia.año)
<i>MPLAB X IDE v4.05</i>	0 (Freeware)
<i>KiCAD r9359</i>	0 (Freeware)
<i>AutoCAD 2017</i>	2075.15
<i>Microsoft Office Word</i>	69*
<i>Microsoft Office Excel</i>	
<i>Microsoft Office Powerpoint</i>	

Tabla 7: Precio unitario del software

*Incluye todos los programas de Microsoft Office.

<i>Mano de obra directa</i>	
TAREA	COSTE (€/hora)
<i>Búsqueda de información</i>	25
<i>Diseño de PCB controladora</i>	60
<i>Montaje de PCB</i>	35
<i>Programación</i>	45
<i>Pruebas de funcionamiento</i>	30
<i>Redacción de documentación</i>	55

Tabla 8: Coste unitario de mano de obra directa

Capítulo 3 PRESUPUESTOS PARCIALES

En este capítulo se calculan los presupuestos parciales calculados a partir de las mediciones anteriores y teniendo en cuenta el grado de amortización anual.

<i>Equipo y herramientas</i>						
MATERIAL	CANTIDAD	HORAS DE PROYECTO	HORAS DE USO AL AÑO	COSTE (€/ud.)	AMORTIZACIÓN ANUAL	COSTE TOTAL (€)
<i>Ordenador</i>	2	600	3000	600	20%	48.00
<i>PCB Controladora</i>	3	300	700	150	20%	38.57
<i>Osciloscopio</i>	1	150	400	400	10%	15.00
<i>Polímetro digital</i>	1	20	70	120	20%	6.86
<i>Soldador</i>	1	15	50	43	15%	1.94
<i>Fuente de alimentación</i>	2	150	350	230	10%	19.71
<i>Generador de señales</i>	1	10	100	570	10%	5.70
<i>Protoboard</i>	2	150	400	15	10%	1.13
<i>Componentes varios</i>	10	130	300	50	15%	32.50
TOTAL EQUIPO Y HERRAMIENTAS:						169.41

Tabla 9: Coste total de equipo y herramientas

Presupuestos Parciales

<i>Software</i>					
PROGRAMA	HORAS DE PROYECTO	HORAS DE USO AL AÑO	PRECIO (€)	AMORTIZACIÓN ANUAL	COSTE TOTAL (€)
<i>MPLAB X IDE v4.05</i>	300	700	0	-	0
<i>KiCAD r9359</i>	150	500	0	-	0
<i>AutoCAD 2017</i>	50	200	2075.15	20%	103.76
<i>Microsoft Office Word</i>	150	500			
<i>Microsoft Office Excel</i>	10	150	69	20%	6.44
<i>Microsoft Office Powerpoint</i>	20	200			
TOTAL SOFTWARE:					110.20

Tabla 10: Coste total de Software

<i>Mano de obra directa</i>			
TAREA	HORAS	COSTE (€/hora)	COSTE TOTAL (€)
<i>Búsqueda de información</i>	100	25	2500
<i>Diseño de PCB controladora</i>	70	60	4200
<i>Montaje de PCB</i>	10	35	350
<i>Programación</i>	250	45	11250
<i>Pruebas de funcionamiento</i>	200	30	6000
<i>Redacción de documentación</i>	150	55	8250
TOTAL MANO DE OBRA DIRECTA:			32550

Tabla 11: Coste total de mano de obra directa

Capítulo 4 PRESUPUESTO GENERAL

El presupuesto general se calcula como la suma de todas las sumas parciales anteriores.

<i>Presupuesto general</i>	
CONCEPTO	SUBTOTAL (€)
<i>Equipo y herramientas</i>	169.41
<i>Software</i>	110.20
<i>Mano de obra</i>	32550.00
TOTAL	IMPORTE (€) 32829.61€

Tabla 12: Presupuesto general

