

TRABAJO FIN DE GRADO INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Diseño e Implementación de Dispositivos IoT para la Monitorización del Consumo Energético en Entornos Domésticos

Autor

Álvaro Mayas Becerro

Directores

Jorge Navarro Ortiz



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Granada, Septiembre de 2017

Diseño e Implementación de Dispositivos IoT para la Monitorización del Consumo Energético en Entornos Domésticos

Autor

Álvaro Mayas Becerro

Directores

Jorge Navarro Ortiz



DEPARTAMENTO DE TEORÍA DE LA SEÑAL, TELEMÁTICA Y COMUNICACIONES

Granada, Septiembre de 2017

Diseño e Implementación de dispositivos Iot para la monitorización del consumo energético en entornos domésticos

Álvaro Mayas Becerro

Palabras clave: Monitorización Energética, Entornos Domésticos, Internet de las Cosas, ESP8266, Thingspeak, Sensores, Arduino, WiFi Manager

Resumen

En la actualidad, existe un término que está cobrando especial importancia conocido como IoT (Internet of Things).

IoT se puede definir como la interconexión digital de objetos cotidianos a Internet. Esto significa que no sólo los humanos tienen la capacidad de conectarse a Internet, sino que se está caminando hacia una nueva era donde prácticamente cualquier cosa podría conectarse.

El presente proyecto se ha centrado en la incorporación de soluciones IoT para entornos domésticos. Tras un estudio del mercado actual, se ha detectado una carencia en la monitorización del consumo energético debida a la inexistencia de dispositivos de bajo coste que sean capaces de medir las principales fuentes de suministro energético en entornos domésticos.

Por este motivo, el objetivo principal del proyecto será llevar a cabo el diseño e implementación de un dispositivo capaz de monitorizar el consumo energético en este tipo de entornos. Para ello, se utilizará una serie de sensores, encargados de obtener la información necesaria (como vatios/min, litros/min, dm^3 /min, etcétera), y un microcontrolador para, posteriormente, gestionar y procesar los datos en una plataforma web para IoT.

Para cumplir estos objetivos, la solución realizada se ha basado en el microcontrolador ESP8266, de muy bajo coste y con conectividad Wi-Fi. Se ha utilizado la plataforma ThingSpeak, que permite el procesado y visualización de los datos.

Design and implementation of devices for the monitoring of energy consumption in domestic environments

Álvaro Mayas Becerro

Keywords: Energy Monitoring, Domestic Environments, Internet of Things, ESP8266, Thingspeak, Sensors, Arduino, WiFi Manager.

Abstract

Nowadays, there is a concept which is gaining special importance known as IoT (Internet of Things).

IoT can be defined as the digital interconnection of daily life objects to Internet. This means that not only humans have the ability to connect to Internet, but we are moving towards a new era where almost anything could be connected.

This thesis is focused on the incorporation of IoT solutions for residential environments. After a survey of the current market, it has been noted a lack of low-cost solutions for energy monitoring since there is no availability of low-cost devices that can monitor the main energy utilities in domestic environments.

For this reason, the main goal of this thesis will be the design and implementation of a device to monitor the energy consumption in home environments. For that purpose, a set of sensors will be used to obtain the required information (such as Watt/min, liters/min, $dm^3/\text{min...}$) and a microcontroller to, subsequently, manage and process the collected data in a web platform.

To achieve the goal, the proposed solution is based on the ESP8266 chip, a low-cost microcontroller with Wi-Fi connectivity. It has been use the ThingSpeak platform, which allows us to process and visualize using MATLAB.



D. **Jorge Navarro Ortiz**, Profesor del Área de Ingeniería Telemática del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado *Diseño e Implementación de dispositivos Iot para la monitorización del consumo energético en entornos domésticos*, ha sido realizado bajo su supervisión por **Álvaro Mayas Becerro**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 12 de Septiembre de 2017.

Los directores:

Jorge Navarro Ortiz

Agradecimientos

Quisiera mostrar mi agradecimiento a todas aquellas personas que de una manera u otra han formado parte de este proyecto.

Primero, quisiera darle las gracias a mi tutor, Jorge Navarro Ortiz por darme su ayuda en todo momento.

También me gustaría mostrar todo mi agradecimiento a mi hermano y Mª Jesús por todas esas palabras de ánimo que me han brindado a lo largo de estos años. A Nuria por su apoyo incondicional en todo momento. Y a toda mi familia y amigos que me han acompañado en esta dura etapa.

Por último, quisiera mostrar mi mayor agradecimiento a mi madre y a mi padre, pilares fundamentales durante esta etapa y toda mi vida. Gracias por vuestro inmenso sacrificio, apoyo, comprensión, cariño, por haber confiado y creído siempre en mí. Sin vosotros nada de esto hubiera sido posible.

Índice general

1.	Intr	oducci	ión	1
	1.1.	Motiva	ación y Objetivos	2
	1.2.	Estruc	etura Memoria	4
2.	Pla	nificac	ción y Costes	5
	2.1.	Planifi	icación	5
	2.2.	Recurs	sos y Costes	8
		2.2.1.	Recursos Hardware	8
		2.2.2.	Recursos Software	8
		2.2.3.	Recursos Humanos	9
		2.2.4.	Coste de los Dispositivos	9
		2.2.5.	Coste del Proyecto	10
3.	Est	ado de	el Arte	13
	3.1.	Antece	edentes	13
	3.2.	Micro	controladores	14
		3.2.1.	Antecedentes	14
		3.2.2.	Tecnología Actual	16
	3.3.	Sistem	nas de Monitorización de la Energía	18
		3.3.1.	Engage Hub Kit de Efergy	19
		3.3.2.	Pack Wattio Energy de Wattio	20
		3.3.3.	Medidor de Consumo de Agua	21
		3.3.4.	GaSmart	22
4.	\mathbf{Re}	quisito	os S	25
	4.1.	Requis	sitos	25
		4.1.1.	Requisitos Funcionales	25
		4.1.2.	Requisitos No Funcionales	26

5.	Tec	cnología Empleada	27
	5.1.	Especificaciones del Hardware Empleado	. 27
		5.1.1. WeMos D1 Mini	. 27
		5.1.2. Microcontrolador ESP8266	. 30
		5.1.3. Sensor de Corriente	. 30
		5.1.4. Sensor Infrarrojo Reflexivo	. 32
		5.1.5. USB Tester	. 34
		5.1.6. Baterías	. 35
	5.2.	Especificaciones del IDE y Software Empleados	. 36
		5.2.1. Entorno de Programación de Arduino (IDE) $\ \ . \ \ . \ \ .$. 36
		5.2.2. ThingSpeak	. 38
		5.2.3. Project Libre	. 41
		5.2.4. Fritzing	. 42
6.	Dis	seño e Implementación	43
	6.1.	Diseño Software	. 43
		6.1.1. WiFiManager	
	6.2.	Diseño Hardware	. 47
		6.2.1. Sensor de Corriente SCT-013-000	. 47
		6.2.2. Sensor infrarrojo reflexivo TCRT5000L	. 49
	6.3.	Implementación	. 50
		6.3.1. Sensor de corriente	. 50
		6.3.2. Sensor Óptico Reflexivo	. 52
7.	Pri	uebas y Resultados	5 5
	7.1.	Pruebas de calibración	. 55
		7.1.1. Sensor de Corriente	. 55
		7.1.2. Sensor Infrarrojo Reflexivo	. 56
	7.2.	Pruebas del Sensor Infrarrojo Reflexivo	. 56
		7.2.1. Prueba de consumo de agua	. 57
		7.2.2. Prueba de Consumo de Gas	. 59
		7.2.3. Prueba de Consumo de Electricidad	. 60
	7.3.	Pruebas del Sensor de Corriente	. 63
	7.4.	Pruebas de Consumo	. 67
		7.4.1. Sensor de Corriente	. 68
		7.4.2. Sensor Infrarrojo Reflexivo	. 69

ÍNDICE GENERAL

8. Conclusiones							
	8.1. Trabajo y Conclusiones						
		8.1.1.	Trabajo Realizado	71			
		8.1.2.	Conclusiones \dots	72			
	8.2.	Lineas	Futuras	73			
		8.2.1.	Autonomía	73			
		8.2.2.	Tecnologías para IoT	74			
\mathbf{Bi}	bliog	rafía		7 9			

Índice de figuras

2.1.	Planificación
2.2.	Diagrama de Gantt
3.1.	Plataforma de hardware libre Arduino Uno.[6] 16
3.2.	Raspberry Pi. [7]
3.3.	Wemos D1 Mini. [8]
3.4.	Engage Hub Kit [9]
3.5.	Esquema del dispositivo Engage Hug Kit. [9] 20
3.6.	Pack Wattio Energy [10]
3.7.	Medidor de consumo de agua $[13]$
3.8.	Medidor de consumo de gas. [13]
5.1.	WeMos D1 Mini. [15]
5.2.	Pines de la placa We Mos D1 Mini. [15]
5.3.	Esquema de funcionamiento del STC-013-000. [23] 31
5.4.	SCT-013-000. [23]
5.5.	Esquema de funcionamiento del TCRT5000L. [18] 33 $$
5.6.	Sensor TCRT5000L. [23]
5.7.	Funcionamiento del USB Tester
5.8.	IDE de Arduino
5.9.	Sketch de Arduino
5.10.	Nuevo canal en ThingSpeak
5.11.	API Keys
5.12.	Project Libre
5.13.	Fritzing
6.1.	Diagrama de flujo del microcontrolador
6.2.	Interfaz web de WiFiManager

ÍNDICE DE FIGURAS

6.3. Interfaz web de Configure Wi-Fi (No Scan)
6.4. Transformación del voltaje. [23]
6.5. Divisor de tensión. [23]
6.6. Esquema de conexiones del sensor de corriente 48
6.7. Esquema del conector Jack 3.5mm. [23]
6.8. Esquema de conexiones del sensor reflexivo TCRT5000L 49
6.9. Medidor de energía eléctrica
6.10. Dispositivo de monitorización de agua y gas
7.1. Calibración del sensor TCRT5000L
7.2. Monitorización del consumo de agua
7.3. Gráfica del consumo de agua
7.4. Gráfica de los litros de agua consumidos
7.5. Monitorización del consumo de gas
7.6. Gráfica del consumo de gas en kWh 60
7.7. Monitorización del consumo eléctrico 61
7.8. Gráfica del consumo eléctrico en Watt 62
7.9. Gráfica del consumo total de Watt 62
7.10. Medidor de consumo eléctrico
7.11. Prueba de 400 Wattios. Thingspeak 64
7.12. Prueba de 400 Wattios. Arduino
7.13. Prueba de 800 Wattios. Thingspeak 65
7.14. Prueba de 800 Wattios. Arduino
7.15. Prueba de 1200 Wattios. Thingspeak
7.16. Prueba de 1200 Wattios. Arduino
7.17. Prueba de consumo. Medidor de corriente
7.18. Segunda prueba de consumo. Sensor Reflexivo 69
8.1 Módulo de batería para WeMos D1 Mini [8] 73

Índice de tablas

2.1.	Presupuesto Dispositivos	1(
2.2.	Presupuesto Dispositivos	10
2.3.	Coste de los recursos software	11
2.4.	Presupuesto Laboral	11
2.5.	Presupuesto Total	11
5.1.	Comparativa de los diferentes microcontroladores	28
5.2.	Especificaciones Técnicas de WeMos D1 mini	29
5.3.	Especificaciones Técnicas del ESP8266	30
5.4.	Especificaciones Técnicas del SCT-013-000	31
5.5.	Especificaciones Técnicas del TCRT5000L (emisor)	33
5.6.	Especificaciones Técnicas del TCRT5000L (detector)	33
6.1.	Cuadro-resumen de las conexiones del sensor SCT-013-000	51
	Cuadro-resumen de las conexiones del sensor TCRT5000L	
7 1	Tabla-resumen de consumo de agua	50
7.2.	Especificaciones Técnicas	63

Acrónimos

API: Application Programming Interface

BMS: Building Management System

EEPROM: Electrically-Erasable Programmable Read-Only Memory

EPROM: Erasable Programmable Read-Only Memory

GUI: Graphical User Interface

HMI: Human-Machine Interface

HTTP: Hypertext Transfer Protocol

IDE: Integrated Development Environment

IoT: Internet of Things

IVA: Impuesto sobre el Valor Añadido

MQTT: Message Queue Telemetry Transport

PROM: Programmable Read-Only Memory

RAM: Random Access Memory

ROM: Read Only Memory

SBC: Single Board Computer

 $\mathbf{TFG} \text{:}\ \operatorname{Trabajo}\ \operatorname{Fin}\ \operatorname{de}\ \operatorname{Grado}$

Capítulo 1

Introducción

El mercado energético actual se encuentra en un proceso de cambio muy importante. La facturación del consumo de la electricidad, agua y gas se establece ahora con el valor del suministro en cada instante, siendo fundamental tanto para las empresas como para los propios consumidores conocer el consumo real en cada momento.

Un buen conocimiento del mismo puede hacer adecuar el mayor consumo a las franjas horarias donde su precio sea menor y así poder redundar su planificación en un consumo más eficaz. De esta forma, el control y la monitorización del consumo del hogar puede suponer una serie de ventajas tanto para los usuarios como para los propietarios en aspectos de seguridad, confort y ahorro energético.

Durante el discurso inaugural de Smart Buildings Now and in the Future (edificios inteligentes de ahora y en el futuro), Jim Sinopoli aseguró que los edificios contabilizan un 41% de energía primaria consumida, más que los sectores del transporte y la industria. Por ello, subrayó la importancia de potenciar el uso de las nuevas tecnologías para mejorar la eficiencia de los inmuebles [1]. Así, haciendo uso de un sistema que permita la monitorización y el control remoto, es posible visualizar en tiempo real parámetros como consumos eléctricos, consumos de agua o consumos de gas. Posteriormente, el análisis de la información obtenida se puede utilizar para poner en marcha medidas de ahorro energético y comprobar, de forma inmediata, su efectividad.

1.1. Motivación y Objetivos

El presente proyecto surge por la necesidad de hacer un uso eficiente de los diferentes tipos de energías en el entorno doméstico. Dado que la mayoría de las fuentes utilizadas en la actualidad son fuentes de energía no renovables y tienen una duración finita, resulta necesario que se utilicen correctamente en el presente para poder seguir disfrutando de ellas en un futuro.

Tras realizar un estudio sobre el mercado de los dispositivos IoT (*Internet of Things*) y más en concreto, sobre los dispositivos de monitorización del consumo energético, se llegó a la conclusión de que la mayoría de ellos eran invasivos, es decir, necesitaban de una instalación previa para su funcionamiento y un mantenimiento posterior. A causa de ello, el precio de estos dispositivos es elevado.

La necesidad de conocer en tiempo real y de manera remota el consumo energético en nuestro hogares y, sobre todo, que pueda ser accesible para cualquier persona o familia, impulsó a este proyecto a la implementación de dos dispositivos. El primero, para la monitorización del consumo energético de un hogar como puede ser electricidad, agua, gas, etcétera. El segundo, para medir el consumo eléctrico de cualquier electrodoméstico y/o dispositivo que el usuario desee.

Ambos dispositivos utilizarán métodos no invasivos, de manera que el dispositivo no necesite ningún tipo de instalación. Por tanto, el usuario podrá retirar el dispositivo con total autonomía cuando lo desee. Todos los datos quedarán almacenados en un servidor web donde el usuario podrá ver en tiempo real su consumo.

Así pues, este proyecto se centrará en construir dos dispositivos de monitorización del consumo energético capaces de competir en precio y en prestaciones con los ya existentes en el mercado. El primer tipo de dispositivo deberá ser compatible con cualquier tipo de contador de los suministros de energía habituales en un hogar (electricidad, agua y gas). El segundo tipo de dispositivo deberá poder medir el consumo de cualquier electrodoméstico habitual en hogares.

Tal y como recoge el título del proyecto, el principal objetivo será el diseño y la implementación de dispositivos IoT de bajo coste para la monitorización del consumo energético. Es posible derivar otros objetivos secundarios necesarios para el cumplimiento de nuestra tarea. Los objetivos del proyecto que se va a desarrollar son:

Desarrollar dos dispositivo capaces de monitorizar el consumo energéti-

Introducción 3

co en entornos domésticos basándose en el módulo ESP8266 y que permita la posterior visualización de los datos en remoto.

- Investigación, búsqueda bibliográfica y estudio del arte de los sistemas Arduino para una posterior comparación de las distintas placas.
- Aprendizaje del lenguaje y forma de programación mediante el IDE (Integrated Development Environment) de Arduino.
- Desarrollar un sistema de almacenamiento de datos basado en una plataforma IoT que permita la integración de diversos sensores y su posterior visualización en remoto.
- Diseñar y configurar un dispositivo capaz de tener un consumo mínimo para prorrogar lo máximo posible el tiempo de vida de la batería.
- Conocer el consumo de energía, ya que esto permitirá averiguar qué sistemas consumen más o cuál es la franja horaria de mayor consumo.
- Probar los dispositivos finales para distintos tipos de contadores y electrodomésticos.
- Realizar un estudio sobre el consumo de un electrodoméstico comparando el consumo medido (por el dispositivo implementado) con el consumo según sus especificaciones.

1.2. Estructura Memoria

Este proyecto está organizado en los siguientes capítulos

- Capítulo 1. Introducción: En esta primera parte se encuentra la motivación que impulsó a la realización del proyecto y los objetivos planteados para conseguir llevarlo a cabo.
- Capítulo 2. Planificación y Costes: En este segundo capítulo se plantea la planificación de todo el proyecto, así como el presupuesto del mismo.
- Capítulo 3. Estado del Arte: En esta parte se revisará la historia de los microcontroladores, como han evolucionado los IoT, los tipos de placas más importantes y los dispositivos del mercado actual que compiten con los que se realizarán en el proyecto.
- Capítulo 4. Requisitos: En este capítulo se exponen todos los requisitos tanto funcionales como no funcionales.
- Capítulo 5. Tecnología Empleada: Este capítulo describe todas las características de todos los componentes que usaremos en el proyecto, tanto componentes hardware como componentes software. Además se hace una justificación del por qué se ha utilizado cada uno de ellos.
- Capítulo 6. Diseño e Implementación: En esta parte se describe el desarrollo del trabajo. Se describirá el montaje físico de cada dispositivo y el algoritmo del microcontrolador. Además se explicará el uso de WifiManager,
- Capítulo 7. Pruebas y Resultados: En este capítulo, se llevarán a cabo las diferentes pruebas con los dispositivos, tanto pruebas de calibración como pruebas finales. Posteriormente, se llevará a cabo el análisis de los resultados.
- Capítulo 8. Conclusiones: Por último, se verán las conclusiones acerca del trabajo realizado y posibles mejoras.

Capítulo 2

Planificación y Costes

Una vez fijados los objetivos del proyecto, es necesario la realización de una planificación y de esta manera ir controlando el desarrollo del proyecto para poder alcanzar con éxito los objetivos.

En este capítulo se va a llevar a cabo la planificación del proyecto así como los costes de producción de los dispositivos.

2.1. Planificación

En este proyecto se ha seguido la planificación que observamos en la figura 2.1.

- En primer lugar se encuentra el punto de Búsqueda y proposición del TFG (Trabajo Fin de Grado). En este apartado se puede ver la Fase de Definición, en la cual se define el tipo de proyecto que se va a realizar con el tutor a lo largo del curso. La fase de Proyecto Pre-Asignado consiste en llegar a un acuerdo con el tutor acerca del TFG seleccionado y, por último, la fase de Proyecto Asignado consiste en la asignación propiamente dicha del proyecto por parte de la ETSIIT (Escuela Técnica Superior de Ingenierías Informática y de Telecomunicaciones. De finales de Octubre a mitad Noviembre.
- En el segundo punto se lleva a cabo el *Proceso de Investigación*, que consta de 3 fases: *Definición de los requisitos* donde se definen todos los objetivos del proyecto para posteriormente buscar la placa y sensores que puedan cumplir con los objetivos. En el *Estudio de las Placas* se investiga sobre las diferentes placas existentes en el mercado para

6

escoger la más adecuada para este proyecto. Finalmente, se realiza el *Estudio de los sensores* donde se estudiarán los distintos sensores que cumplan con los requisitos.

- En el tercer punto se encuentra el Proceso de Implementación, que es la parte más compleja y extensa del proyecto. Consta de distintas fases: una primera fase de Estudio del Lenguaje de Programación para la elaboración del código del proyecto. Una segunda fase de Implementación del Hardware y Base de Datos, en paralelo con la fase anterior. En esta fase se lleva a cabo la implementación de los dispositivos y la conexión a la plataforma IoT, Thinkspeak. Y por último, la Fase de Pruebas en la que se llevarán a cabo las pruebas iniciales y finales con los diferentes dispositivos.
- El último punto será la *Elaboración de la Documentación*. Esta última fase se dedicará a redactar la memoria del proyecto y a la elaboración de la presentación para la posterior defensa del mismo.

Nombre	Duracion	Inicio	Terminado
Trabajo Fin de Grado	233 days?	20/10/16 8:00	11/09/17 17:00
⊟Búsqueda y proposición del Trabaj	23 days?	20/10/16 8:00	21/11/16 17:00
Fase de definición	13 days?	20/10/16 8:00	7/11/16 17:00
Proyecto Pre-Asignado	6 days?	8/11/16 8:00	15/11/16 17:00
Proyecto Asignado	4 days?	16/11/16 8:00	21/11/16 17:00
⊟Proceso de Investigación	62 days?	22/11/16 8:00	15/02/17 17:00
Definición de requisitos	5 days?	22/11/16 8:00	28/11/16 17:00
Estudio de las placas	19 days?	12/12/16 8:00	5/01/17 17:00
Estudio de los sensores	47 days?	13/12/16 8:00	15/02/17 17:00
□Proceso de Implementación	88 days?	17/02/17 8:00	20/06/17 17:00
Estudiar Lenguajes de Programaciói	9 days?	17/02/17 8:00	1/03/17 17:00
Implementación del Software	57 days?	2/03/17 8:00	19/05/17 17:00
Implementación del Hardware	70 days?	15/03/17 8:00	20/06/17 17:00
Implementación de Thingspeak	15 days?	1/05/17 8:00	19/05/17 17:00
Fase de Pruebas	19 days?	25/05/17 8:00	20/06/17 17:00
⊟Elaboración Documentos	73 days?	1/06/17 8:00	11/09/17 17:00
Memoria	45 days?	1/06/17 8:00	2/08/17 17:00
Presentación	7 days?	1/09/17 8:00	11/09/17 17:00

Figura 2.1: Planificación.

La planificación se puede ver de una manera más visual en el diagrama de Gantt, figura 2.2. Existen etapas que serán realizadas simultáneamente.

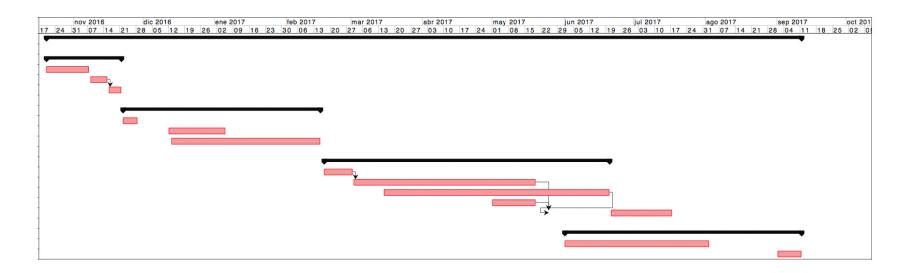


Figura 2.2: Diagrama de Gantt.

2.2. Recursos y Costes

Cuando se está frente algún problema, siempre es necesario realizar una estimación de los recursos de los que se disponen así como el coste que tienen asociado.

En la siguiente sección se explicará todos los recursos de los que disponen así como sus costes.

2.2.1. Recursos Hardware

Como soporte fundamental de todas las actividades desarrolladas durante el proyecto se ha utilizado un ordenador portátil con un sistemas operativo Mac OS, cuyo valor es de $1530 \in$. En él, se han instalado todos los programas y herramientas software necesarias para la implementación del código en Arduino. Si fijamos el tiempo de vida del portátil en 5 años y teniendo en cuenta que solo ha sido utilizado durante 10 meses, contar con este elemento nos supondrá un coste de $255 \in$.

También deberemos adquirir una placa de desarrollo y los sensores necesarios para la monitorización del consumo energético. En capítulos posteriores se desarrollará con mayor profundidad el conjunto de elementos hardware necesarios para implementar el dispositivo de monitorización.

2.2.2. Recursos Software

Los recursos software será utilizados para toda la parte de programación del proyecto, así como para la elaboración de los documentos.

- Arduino: Es una compañía que desarrolla placas de hardware y software libre, compuestos por un circuito impreso que integran un microcontrolador y un IDE. Todas la plataformas de Arduino son liberadas con licencia de código abierto por lo que se permite total libertad de acceso a ellas.
- ThingSpeak: Plataforma online empleada para almacenar los datos.
 Ofrece una versión gratuita que será empleada en el proyecto, ya que cumple con los requisitos planteados anteriormente.
- ProjectLibre: ProjectLibre es un software de administración de proyectos de código abierto, similar a Microsoft Project. ProjectLibre corresobre la plataforma Java lo que permite ejecutarlo en varios sistemas

operativos. Este programa será utilizado para la creación del diagrama de Gantt y de la planificación. Tiene una versión gratuita.

- Fritzing: Es un programa de software libre para el diseño electrónico.
 Permite realizar diseños utilizando una gran variedad de componentes de diferentes fabricantes.
- Overleaf: Plataforma web que permite desarrollar documentos en latex de forma gratuita. Será empleada para redactar la memoria.
- Smartdraw: Plataforma online que se utilizará para realizar un diagrama de flujo que refleje el comportamiento del dispositivo. Disponible de forma gratuita.

2.2.3. Recursos Humanos

En esta sección se hablará de todas las personas que han participado en el desarrollo del trabajo. En primer lugar, los tutores que son los encargados de ofertar y guiar el proyecto. También serán los encargados de la verificación del cumplimiento de los objetivos. Por otro lado, el alumno llevará a cabo todos los estudios y búsquedas necesarias para el desarrollo del mismo, contado siempre con la ayuda de los tutores para solucionar cualquier tipo de problema.

Como se ha podido ver en la planificación el proyecto ha sido desarrollado en 233 días. El trabajo realizado durante esos días ha ido variando dependiendo de la tarea en ese momento. Estimando las horas de trabajo por día y haciendo una media, se han obtenido entorno a unas dos horas y media por día. Por tanto, el desarrollo total del proyecto ocupará un total de 583 horas.

El salario del alumno se ha calculado teniendo en cuenta que es un trabajador con poca experiencia, por lo que se atribuye $25 \in /$ hora. El tutor, se le atribuirá el doble que al alumno ya que tiene mucha más experiencia y además es el encargado de la dirección del proyecto.

2.2.4. Coste de los Dispositivos

En sección se va a detallar el coste total de los dispositivos teniendo en cuenta el principal requisito, el bajo coste.

En la tabla 2.1 se puede ver el coste total de los dispositivos. Como se puede apreciar, se ha conseguido cumplir unos de los objetivos principales,

la obtención de dispositivos de monitorización del consumo energético a bajo coste.

Materiales	Cantidad	Precio total €
WeMos D1 Mini	1	2.76
SCT-013-000	1	6.03
Sensor TCRT5000L	1	0.17
Placa protoboard	2	5.72
Cables	10	1
Resistencias	4	0.12
Condensador Electrolítico	1	0.08
TOTAL	-	15.88

Tabla 2.1: Presupuesto Dispositivos.

2.2.5. Coste del Proyecto

Para terminar este capítulo, se va hacer una recopilación de todos los costes del proyecto y así obtener el coste total del mismo.

Todos los presupuestos que se han realizado en el proyecto se han hecho con el IVA (*Impuesto sobre el valor añadido*) incluido. Se ha realizado así, porque es difícil saber que impuestos se pagan en China cuando compramos en páginas como Aliexpress, Amazon, etcétera.

En cuanto al coste en recursos hardware es el mostrado en la tabla 2.2.

Componente	Coste (€)	
Portátil	255	
Dispositivos	15.88	
TOTAL	270.88	

Tabla 2.2: Presupuesto Dispositivos.

En segundo lugar, se tienen todos los elementos software descritos en la sección anterior. El coste total de estos se muestra en la tabla 2.3.

Elemento	Coste (€)
Arduino	0
ThingSpeak	0
ProjectLibre	0
Fritzing	0
Overleaf	0
Smartdraw	0
TOTAL	0

Tabla 2.3: Coste de los recursos software.

En la tabla 2.4 se muestra el coste total de las personas encargadas del desarrollo del proyecto.

Profesión	€/hora	Horas Totales	Total €
Alumno	25	583	11660
Tutor/Profesor	50	25	1250
TOTAL	-	-	12910

Tabla 2.4: Presupuesto Laboral.

Finalmente, en la tabla 2.5 se muestran el coste total del proyecto.

Tipo de Presupuesto	Total €
Coste Hardware	270.88
Coste Software	0
Coste Laboral	12910
Presupuesto Total	13180.88

Tabla 2.5: Presupuesto Total.

Capítulo 3

Estado del Arte

3.1. Antecedentes

La automatización y telecontrol no es un concepto nuevo. Estas tecnologías se pueden aplicar a diversos ámbitos: si nos referimos a una vivienda estaríamos hablando de domótica, si se aplica a edificios se conoce como inmótica, y si abarca a toda una ciudad podría definirse como urbótica. En este trabajo nos hemos centrado en aplicaciones para entornos doméstico como puede ser comprobar el consumo de un dispositivo eléctrico o monitorizar los consumos de electricidad, agua y gas de una vivienda particular.

En los años 70 ya existían soluciones basadas en protocolos X-10. Permitían la comunicación entre sensores y actuadores de una vivienda que se comunicaban a través de la línea eléctrica. Con el paso de los años han ido apareciendo protocolos y sistemas cada vez más complejos y a su vez más robustos, que cuentan con unas características más adecuadas a cada campo. Por nombrar algunos, en automatización de edificios y viviendas podemos encontrar KNK y LonWorks, y en el ámbito industrial podríamos destacar Profibus, CAN y Modbus, entre otros.

Este apartado no se centrará en profundizar en los procesos de automatización y los buses de campo, ya que el objetivo de este proyecto es el uso de otras tecnologías como sensores de bajo coste, no invasores, con una interfaz web simple y sencilla capaz de ser manejada por cualquier usuario.

Los sistemas de gestión de edificios conocidos como BMS (Building Management System) suelen estar formados por un conjuntos de sensores, actuadores, controladores, pasarelas, buses de campo y una aplicación o programa que sirva de interfaz hombre-máquina (HMI) para facilitar la gestión.

En este proyecto, solo utilizaremos los sensores y la interfaz web para la monitorización del consumo energético. Con esto, lo que conseguiremos será tener dispositivos de bajo coste capaces de monitorizar en tiempo real el consumo energético.

3.2. Microcontroladores

3.2.1. Antecedentes

Un microcontrolador es un circuito integrado programable capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

El inicio de los microcontroladores tiene sus raíces en el desarrollo de la tecnología de los circuitos integrados. Este desarrollo ha hecho posible contener cientos de miles de transistores en un solo chip, requisito previo para la producción de microprocesadores. Las primeras computadoras se construían agregando elementos externos como memorias, contadores, dispositivos de entrada/salida, etcétera. Sin embargo, el incremento posterior de la densidad de integración permitió crear circuitos integrados que contenían tanto al procesador como periféricos. Así es cómo fue desarrollada la primera microcomputadora en un solo chip, denominada más tarde microcontrolador.

El Instituto Smithsoniano afirma que los ingenieros de Texas Instruments Gary Boone y Michael Cochran lograron crear el primer microcontrolador, TMS 1000, en 1971, siendo comercializado en 1974. Este microcontrolador combinaba memoria ROM (Read Only Memory), memoria RAM (Random Access Memory), microprocesador y reloj en un chip que estaba destinado a sistemas embebidos [2].

La mayoría de los microcontroladores en ese momento tenían dos variantes. Unos tenían una memoria EPROM (*Erasable Programmable Read-Only Memory*) reprogramable pero eran significativamente más caros que la variante PROM (*programmable read-only memory*), que sólo era programable una vez. Para borrar la EPROM se necesita exponer la tapa de cuarzo transparente a la luz ultravioleta.

En 1993 tuvo lugar el lanzamiento de la memoria EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) en los microcontroladores (comenzando con el Microchip PIC16x84). Estas pueden borrarse de manera eléctrica y rápidamente sin necesidad de asumir un coste elevado, permitiendo la creación rápida de prototipos y una sencilla programación de los microcontroladores. El mismo año, Atmel lanzó el primer microcontrolador

Estado del Arte 15

con memoria flash. Otras compañías rápidamente siguieron su ejemplo, con los dos tipos de memoria [3].

Desde un punto de vista de arquitectura de computadores, existen dos arquitecturas básicas que, por supuesto, están presentes en el mundo de los microcontroladores: Von Neumann y Harvard. Ambas se diferencian en la forma de conexión de la memoria del procesador y en los buses que cada uno necesita [4].

Arquitectura Von Neumann

La arquitectura Von Neumann utiliza el mismo dispositivo de almacenamiento tanto para instrucciones como para datos. En el caso de los microcontroladores, existen dos tipos de memoria bien definidas: memoria de datos (típicamente de tipo SRAM) y memoria de programa (PROM, EEPROM, flash u otro tipo de memoria no volátil). En este caso, la memoria está segregada y el acceso a cada tipo de memoria depende de las instrucciones del procesador.

Las familias de microcontroladores como Intel 8051 y Z80 implementan este tipo de arquitectura, fundamentalmente porque era la utilizada cuando aparecieron los primeros microcontroladores.

Arquitectura Harvard

Esta arquitectura es la utilizada en los microcontroladores y sistemas integrados en general. En este caso, además de la memoria, el procesador tiene los buses segregados de modo que cada tipo de memoria tiene un bus de datos, uno de direcciones y uno de control.

La ventaja fundamental de esta arquitectura es que permite adecuar el tamaño de los buses a las características de cada tipo de memoria.

Además, el procesador puede acceder a cada una de ellas de forma simultánea, lo que se traduce en un aumento significativo de la velocidad de procesamiento. Los sistemas con este tipo de procesamiento pueden ser típicamente dos veces más rápidos que sistemas similares con arquitectura Von Neumann.

El coste de los microcontroladores se ha ido desplomando con el tiempo, siendo en la actualidad muy baratos y disponibles para desarrolladores aficionados. Además, muchos de ellos cuentan con grandes comunidades en línea con multitud de librerías y ejemplos.

3.2.2. Tecnología Actual

En el año 2005, en el instituto de IVREA (Italia) nace el proyecto Arduino, una plataforma de hardware libre basada en una placa con un microcontrolador y un entorno de desarrollo. Fue diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.

El hardware está formado por un microcontrolador, generalmente Atmel AVR, y puertos de entrada/salida tanto analógicos como digitales. El microcontrolador en la placase programa mediante el lenguaje de programación Arduino (basado en *Wiring*) y el entorno de desarrollo Arduino (basado en *Processing*) [6].



Figura 3.1: Plataforma de hardware libre Arduino Uno.[6]

A partir de la fecha en la que nace esta plataforma surgen otras muchas, como por ejemplo Raspberry Pi, un ordenador de placa reducida o placa única que revoluciona el mundo de los SBC (Single Board Computer) por su pequeño tamaño y su bajo coste. Fue desarrollada en Reino Unido por la Fundación Raspberry Pi con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El hardware está formado por un "System On Chip Broadcom" con procesador ARM 1176JZF-S a 700 MHz, un procesador gráfico Videocore IV y 512 MB de memoria RAM. El sistema operativo es cargado desde una tarjeta de memoria SD que usa como dispositivo de almacenamiento. Raspberry Pi funciona mayoritariamente con sistemas operativos basados en el núcleo Linux [7].

Estado del Arte 17



Figura 3.2: Raspberry Pi. [7]

Otra de las plataformas existentes y la seleccionada para la implementación de este proyecto es WeMos D1 mini, un SOC (System On Chip) de 32 bits a 160 MHz con memoria RAM y EEPROM más el stack Wi-Fi, disponible por un coste muy reducido.

La gran virtud de esta placa es que está gobernada por el chip ESP8266. Esta MCU (*MicroController Unit*) es la encargada de las tareas de procesamiento y control de Wi-Fi. Además, es totalmente compatible con entornos de programación como Arduino, LUA, NodeMCU, etcétera [8].

Wemos D1 mini fue elegido teniendo en cuenta diversos aspectos. Primero, se buscaba un microcontrolador (MCU) ya que inicia más rápidamente que un miniPC tipo Raspberry Pi, que tiene que cargar un sistema operativo completo. En segundo lugar, se necesitaba una placa que cumpliera los requisitos de bajo coste. En este caso los sistemas basados en ESP8266 son muy baratos y además tienen Wifi integrado por lo que no necesitaba un módulo wifi adicional como lo necesita arduino uno.

Wemos D1 mini cumplía con estos requisitos pero además presentaba otras ventajas frente a sus competidores.

- Tamaño: Este modelo de placa ofrece un tamaño muy reducido frente a Arduino uno, Raspberry Pi, entre otros.
- Entradas y salidas disponibles: Esta placa cuenta con multitud de entradas y salidas. Cuenta con dos salidas, una de 5V y otras de 3,3V, para distintos tipos de sensores. Además tiene una salida analógica.

- Sistema abierto: Se basa en el chip ESP8266, que es programable con el IDE Arduino.
- Sistema Integrado: Es un sistema que está preparado para programarse directamente a diferencia de ESP12.
- Muy extendido y estandarizado: Existen infinidad de librerías de libre distribución para poder comunicarse con hardware y software de terceros.



Figura 3.3: Wemos D1 Mini. [8]

3.3. Sistemas de Monitorización de la Energía

El avance de la tecnología hace que cada vez se desarrollen nuevos sistemas para la monitorización del consumo energético. Por ejemplo, *Open Domo, Current Cost, Circutor, Cliensol Energy, Wattio Energy y OpenEnergyMonitor* son algunos de los productos más conocidos en el mercado.

En esta sección se analizarán algunos productos con prestaciones similares a los requisitos de este proyecto. Aunque el mercado de estos dispositivos de monitorización es muy amplio, se han analizado cuatro productos representativos. Los dos primeros permiten la monitorización del consumo eléctrico, el tercero está pensado para monitorizar el consumo de agua y, por último, se ha analizado un producto para monitorizar el consumo de gas.

Estado del Arte

3.3.1. Engage Hub Kit de Efergy

Efergy es uno de los fabricantes globales de productos de ahorro energético cuyo objetivo es ayudar a reducir el consumo eléctrico, ahorrar dinero y conservar el medio ambiente.

Su producto más básico es el llamado "Engage Hub Kit". Este producto se compone de un sensor no invasivo que se conecta al cable de fase del cuadro eléctrico y, en su otro extremo, al transmisor de datos. Este transmisor, a su vez, está conectado de forma inalámbrica a un receptor llamado "hub", que se conecta a Internet a través de un cable Ethernet. Los datos pueden ser recogidos en un intervalo de tiempo de 6, 12 ó 18 segundos, según la configuración que quiera tener el cliente. Los datos recogidos se suben a una plataforma web y pueden ser consultados y gestionados desde cualquier dispositivo.



Figura 3.4: Engage Hub Kit [9]

El precio de este dispositivo es de 79,90 euros. Sus ventajas e inconvenientes son las siguientes:

Ventajas: Su atractiva web, sus aplicaciones para los sistemas operativos móviles Android e IOS y un sistema social media que permite compartir el consumo con amigos, siendo este uno de los más económicos del mercado.

• Inconvenientes: La cantidad de elementos del sistema (sensor, emisor, receptor), necesita emitir la señal a otro aparato, utiliza baterías para el emisor, sus conexión a Internet es por cable Ethernet, solo se puede monitorizar una señal de consumo en todo el sistema, y la plataforma web y las aplicaciones están adaptadas a esta única señal.

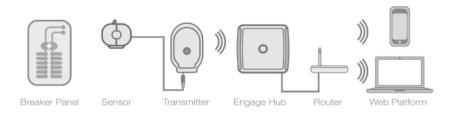


Figura 3.5: Esquema del dispositivo Engage Hug Kit. [9]

3.3.2. Pack Wattio Energy de Wattio

Otro de los productos que se han analizado para la monitorización de la corriente eléctrica es *Pack Wattio Energy*. Este producto es uno de los sistemas más completos del mercado. Se compone de una serie de dispositivos de recogida de datos que se gestionan desde una consola domótica con pantalla táctil, o simplemente accediendo a la aplicación de *Wattio* desde un ordenador o cualquier dispositivo móvil con conexión a Internet.



Figura 3.6: Pack Wattio Energy [10]

Su precio es de 299,90 euros y el pack incluye: centralita, enchufe inteli-

Estado del Arte 21

gente, medidor de electricidad y termostato.

■ Ventajas: Es un sistema muy completo, con enchufe inteligente que mide el consumo de dispositivos individuales, medidor de electricidad de 3 circuitos para el cuadro eléctrico, centralita con pantalla táctil, aplicaciones para dispositivos móviles y se le puede agregar tantos dispositivos como se necesiten.

 Inconvenientes: Necesita una centralita, la comunicación entre dispositivos es a lo sumo de 30 metros y utiliza baterías para los sistemas enlazados. Es un sistema costoso para el cliente.

3.3.3. Medidor de Consumo de Agua

Cliensol es uno de los fabricantes que más apoya el ahorro del consumo energético y una de las empresas que más trabaja en las nuevas tecnologías para intentar aplicarla en la eficiencia energética.

Uno de sus productos es el medidor de consumo de agua. Este producto está compuesto por un transmisor digital, un sensor de agua y un contador de agua.



Figura 3.7: Medidor de consumo de agua [13]

El precio de este dispositivo ronda los 100 euros. Este dispositivo presenta

las siguientes ventajas e inconvenientes.

- Ventajas: Una de sus principales ventajas es que todos los datos recogidos y almacenados puede ser tratados por las distintas plataformas de gestión de energía disponibles.
- Inconvenientes: Este sensor de agua debe ser colocado en el tubo de entrada general, por lo que es un sensor invasivo y necesita de una instalación previa, haciendo al producto más costoso. Además es necesario contar con el monitor de energía EnviR+Módulo de Internet, o bien, con un módulo gateway. También, necesita la licencia correspondiente a la plataforma de gestión de energía online escogida.

3.3.4. GaSmart

Este último producto es capaz de monitorizar el consumo de gas. GaS-mart, pertenece al mismo fabricante que el sensor de agua mencionado antes, Cliensol Energy.

GaSmart permite supervisar los consumos de gas del hogar al instante. Esto permite la creación de múltiples comparativas y optimización del uso y gestión del consumo de gas. El precio de este dispositivo es de 70 euros.



Figura 3.8: Medidor de consumo de gas. [13]

Estado del Arte 23

El medidor de consumo de gas está compuesto por un transmisor digital y un lector (ya sea de tipo V o G). El lector se debe ubicar directamente en el contador de gas de tu compañía, conectando previamente el transmisor digital. Esto permite visualizar el consumo de gas, además de poder ver los consumos en tiempo real.

GaSmart presenta la mismas ventajas e inconvenientes que el medidor de consumo de agua. Como se aprecia, este sensor es invasivo por lo que requiere una instalación. Una de sus mayores ventajas es la diversidad de plataformas de gestión que puede utilizar el usuario para gestionar el consumo.

Capítulo 4

Requisitos

Los requisitos del proyecto representan una visión clara de las necesidades, deseos y expectativas contenidas en el proyecto. En este capítulo se va a exponer todos los requisitos necesarios para llevar a cabo el proyecto.

4.1. Requisitos

En esta sección se va a explicar los requisitos necesario para llevar a cabo la ejecución del proyecto. Se abordarán los requisitos funcionales, que son aquellos que definen una función del sistema de software o sus componentes. Los requisitos no funcionales, son aquellos que definen un atributo de calidad, es decir, características del funcionamiento.

4.1.1. Requisitos Funcionales

- 1. Lectura del consumo energético: El usuario podrá colocar un sensor en su cuadro eléctrico, contador de agua, contador de gas o dispositivo eléctrico que permite medir, en tiempo real, el consumo.
- 2. Configuración de la comunicación: El usuario deberá configurar una conexión Wi-Fi en el dispositivo para enviar los datos.
- 3. Creación de una cuenta en una plataforma web IoT: El usuario deberá crear una cuenta en la plataforma web IoT utilizada para la posterior monitorización del consumo energético.
- 4. Registro de los dispositivos en la plataforma: El usuario podrá registrar y monitorizar tantos dispositivos como quiera, sin límite de

capacidad.

- Envío de datos a la plataforma: La plataforma IoT elegida permitirá el envío de un número suficiente de mensajes para la funcionalidad requerida.
- 6. Consulta de consumo: El usuario podrá consultar el consumo instantáneo en la plataforma web IoT cuando lo desee. Además, se proporcionará un histórico y gráficos en la plataforma para un control más visual del consumo.

4.1.2. Requisitos No Funcionales

- 1. **Seguridad en la comunicación:** El sistema debe tener un procedimiento de cifrado para la comunicación con la base de datos.
- 2. Seguridad en el acceso de la plataforma: El sistema debe tener un procedimiento de seguridad para el acceso a la plataforma.
- 3. **Usabilidad:** Tanto los dispositivos como la plataforma web serán sencillos e intuitivos para el posterior uso de cualquier usuario.
- 4. **Bajo coste:** Este es un requisito fundamental del proyecto. El dispositivo desarrollado en el proyecto, incluyendo microcontrolador, sensores, etcétera, deberá ser de bajo coste.
- 5. **Bajo consumo:** Es importante que el consumo del dispositivo sea mínimo para aumentar la duración de la batería.
- 6. **Fiabilidad y disponibilidad:** La base de datos deberá ser fiable y deberá estar disponible en el momento que desee el usuario.

Capítulo 5

Tecnología Empleada

En este capítulo se abordará toda las herramientas utilizadas en el proyecto y se explicará el funcionamiento de cada una de ellas. Además, se justificará por qué se han usado en el proyecto.

5.1. Especificaciones del Hardware Empleado

Para el diseño e implementación de estos dispositivos se han utilizado cuatro elementos fundamentales: la placa WeMos, cuyo microcontrolador se encarga de leer y comunicar los datos con la plataforma *Thingspeak*; el sensor de corriente, que mide la corriente eléctrica que atraviesa el conductor; el sensor óptico reflexivo, capaz de monitorizar los consumos energéticos básicos en el entorno del hogar; un USB Tester, utilizado en las pruebas de consumo para poder optimizar la batería que alimentará al sistema.

5.1.1. WeMos D1 Mini

Como se ha dicho en los anteriores capítulos, la elección de WeMos D1 Mini fue debido a su sistema integrado, a su tamaño y, sobre todo, a su bajo coste. WeMos D1 mini trabaja a 3.3V, lo que es adecuado para los sensores usados en el proyecto. Otro aspecto muy importante a la hora de su elección fue el consumo, ya que es muy bajo comparado con otras placas basadas también en el chip ESP8266.

Wemos D1 Mini contiene entradas y salidas tanto analógicas como digitales que permite leer señales desde los sensores, procesarlas y comunicarlas a la plataforma web para su posterior visualización. Además, posee un bajo

consumo y una capacidad de cómputo que cubre todas las necesidades del proyecto.

En la tabla 5.1, se puede ver algunos de los modelos oficiales de microcontroladores de Arduino y WeMos con sus respectivas especificaciones. En la segunda posición podemos ver la placa utilizada en el proyecto.

Name	Proce.	Oper./In. Vol	CPU Speed
WeMos D1	ATSAM3XSE	3.3V/5V	84MHz
WeMos D1 Mini	ATSAM3XSE	3.3V/5V	84MHz
101	Intel Curie	3.3V/7-12V	32 MHz
Gemma	ATtiny85	3.3V/4.16V	8 MHz
LilyPad	ATmega 168P	2.7/2.7V-5.5V	8 MHz
LilyPad USB	ATmega32U4	3.3V/3.8-5V	8 MHz
Mega 2560	Atmega2560	5V/7-12V	16 MHz
Micro	ATmega32U4	5V/7-12V	16MHz
MKR1000	SAMD21	3.3V/5V	48 MHz
Pro Mini	ATmega328P	3.3V/ 3.35-12V	8 MHz
Uno	ATmega328P	5V/7-12	16 MHz
Zero	ATSAMD21G18	3.3V/7-12	48 MHz
Due	ATSAM3X8E	3.3V/7-12	84 MHz
Esplora	ATmega32U4	5V/7-12V	16 MHz
Leonardo	ATmega32U4	5V/7-12V	16 MHz
Mega ADK	ATmega2560	5V/7-12	16 MHz
Mini	ATmega328P	5V/7-9V	16 MHz
Nano	ATmega168	5V-7-9V	16 MHz
Yùn	ARp331 Linux	5V	16MHz
Arduino Robot	ATmega32u4	5V	16 MHz

Tabla 5.1: Comparativa de los diferentes microcontroladores

Como se puede comprobar, la variedad de microcontroladores en el mercado es muy amplia. Solo se han mencionado los microcontroladores más importantes de Arduino y WeMos pero hay muchos más fabricantes de microcontroladores, como pueden ser Tessel, Picaxe o Wiring.

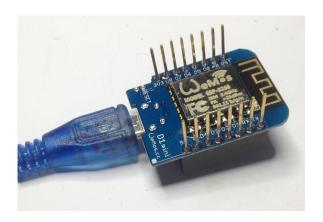


Figura 5.1: WeMos D1 Mini. [15]

El WeMos D1 mini tiene las siguientes especificaciones técnicas:

Características	Valor
Modelo	D1 mini
Microcontrolador	ESP8266
Voltaje de funcionamiento	3.3V
Voltaje de entrada	5V
Entradas/salidas digitales	11
Entradas analógicas	1
Memoria Flash	4Mb
SRAM	1Kb
EEPROM	512 bytes
Velocidad de reloj	$80 \mathrm{MHz} / 160 \mathrm{MHz}$

Tabla 5.2: Especificaciones Técnicas de WeMos D1 mini

Todas las características técnicas han sido obtenidas de la hoja de características de WeMos D1 Mini [8].

En la figura 5.12 se puede ver la descripción de los pines (pinout) de la placa WeMos D1 Mini:

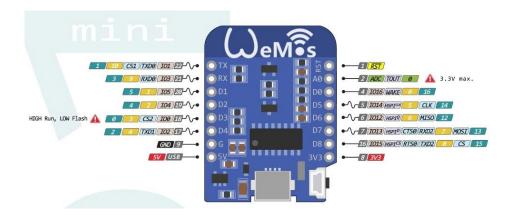


Figura 5.2: Pines de la placa WeMos D1 Mini. [15]

5.1.2. Microcontrolador ESP8266

Como se ha comentando en el apartado anterior, la placa WeMos D1 mini está gobernada por un microcontrolador ESP8266.

Este microcontrolador se encarga del procesamiento y control Wi-Fi. También se puede utilizar para ejecutar una aplicación o para otorgar conectividad a otro procesador. Es fabricado por la empresa Espressif. Algunas de sus principales características las podemos encontrar en la tabla 5.3.

Características	Valor
Voltaje de operación	3.3 V - 3.6 V
Corriente de operación	80 mA
Protocolos Wi-Fi	802.11 b/g/n
Rango de frecuencia	2.4 GHz - 2.5 GHz
Consumo Deep Sleep	$\sim 10 \ \mu A$

Tabla 5.3: Especificaciones Técnicas del ESP8266.

Todos las características y valores han sido obtenidos de la hoja de características del microcontrolador ESP8266 [19].

5.1.3. Sensor de Corriente

En el caso del dispositivo que mide el consumo de electrodomésticos, para medir la corriente se ha utilizado un transformador de corriente de núcleo dividido. Este sensor puede ser abierto y puesto alrededor de los cables existentes sin tener que interrumpir el circuito. De esta manera, se

cumple uno de los principales objetivos del proyecto, hacer que el dispositivo fuese no invasivo.

El sensor actúa como inductor y responde al campo magnético alrededor de un cable con corriente. Al leer la cantidad de corriente que se produce por la bobina, se puede calcular la cantidad de corriente que pasa por el cable.

La pinza amperímetra sigue el principio de un transformador. El transformador está formado por dos devanados (primario y secundario) y un núcleo ferromagnético. Por el efecto de la inducción electromagnética conseguiremos una intensidad de corriente en el devanado secundario proporcional a la corriente que pasa por el primario.

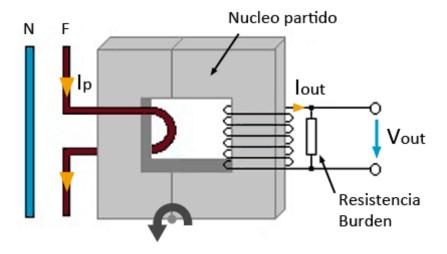


Figura 5.3: Esquema de funcionamiento del STC-013-000. [23]

El sensor tiene una terminación con conector jack de 3.5mm. figura 5.4, que se usará como estándar de conexión de sensores al dispositivo. El transformador de corriente usa su inductancia para medir la corriente, por lo que el circuito debe pasar al menos una vez a través del sensor.

Este sensor tiene las especificaciones técnicas vistas en la tabla 5.4

Características	Valor
Modelo	SCT-013-000
Tensión de entrada	0-100A
Voltaje de salida	0-50mV
No linealidad	±1 %

Tabla 5.4: Especificaciones Técnicas del SCT-013-000

Todas las especificaciones del sensor SCT-013-000 han sido obtenidas de su hoja de características [17].



Figura 5.4: SCT-013-000. [23]

Durante la búsqueda del medidor de corriente, se disponía de un amplio mercado de productos. En la rama de SCT, se encontraba el SCT-013-030. Este medidor de corriente puede medir hasta un rango de 3A, el cual no era suficiente para el proyecto. Esta fue la primera ventaja que tomó el SCT-013-000 ya que este puede medir hasta 100A. Además el SCT-013-030 proporciona una salida de voltaje, lo que obliga hacer una pequeña transformación en el circuito para obtener una salida de corriente. El SCT-013-000 proporciona directamente una salida de corriente.

Otro medidor de corriente de bajo coste es el ACS712. Aunque el precio de este es bajo, se tuvo que descartar debido a que es un sensor invasivo, es decir, es necesario insertarlo en el conductor lo que puede obligar a cortar el cable.

Por tanto, SCT-013-000 fue el elegido para el proyecto ya que cumplía con todos los requisitos de potencia, coste y no-invasivo.

5.1.4. Sensor Infrarrojo Reflexivo

El sensor infrarrojo reflexivo es un sensor que detecta la diferencia de color de un objeto mediante la reflexión de la luz en el mismo. Debido a sus características, se ha utilizado el TCRT5000L.

Este sensor está compuesto por una luz infrarroja y un receptor. Este receptor está constantemente recibiendo luz infrarroja del emisor pero en el

momento de pasar el objeto (banda magnética, aguja del contador, etc), este reflejará más o menos luz, dependiendo si el objeto es más reflector o menos, obteniendo así un pulso. Previamente habremos de determinar el valor del pulso, es decir, cada giro de la aguja o de la banda magnética equivale a un valor fijo de consumo para posteriormente enviarlo a nuestra plataforma web.

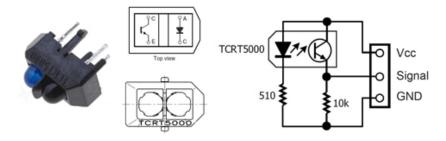


Figura 5.5: Esquema de funcionamiento del TCRT5000L. [18]

Las características básicas del sensor TCRT5000L son:

• Para el emisor

Características	Valor
Modelo	TCRT5000L
Corriente Directa	$60 \mathrm{mA}$
Corriente Máxima	3A
Disipación de Potencia	$100 \mathrm{mW}$

Tabla 5.5: Especificaciones Técnicas del TCRT5000L (emisor).

• Para el detector

Características	Valor
Modelo	TCRT5000L
Voltaje del emisor del colector	70V
Voltaje del colector del emisor	5V
Corriente del Colector	100mA

Tabla 5.6: Especificaciones Técnicas del TCRT5000L (detector).

Todos los datos utilizados en dicho proyecto, han sido obtenidos de la hoja de características del sensor [18].

El TCRT5000L ha sido elegido, principalmente, por su sistema integrado. Es decir, contenía el detector y el led infrarrojo en el mismo sensor. Este sensor tiene una pequeña barrera entre el detector y el led para evitar el ruido, la cual no tiene su principal competidor en este proyecto, CNY70.

Otro de los candidatos para este proyecto era el Módulo Arduino Sensor IR infrarrojo (reflexivo) pero este tenia un precio muy superior al TCRT5000L.

Por tanto, la relación calidad-precio de TCRT5000L y unido a que es un sensor no-invasivo inclinó la balanza hacia este sensor.



Figura 5.6: Sensor TCRT5000L. [23]

5.1.5. USB Tester

El USB Tester es un dispositivo capaz de medir el consumo eléctrico de una conexión USB. Se decidió utilizar este dispositivo porque es muy fácil de usar y por tenerlo disponible. Tan solo se tiene que conectar entre la fuente de alimentación y el dispositivo que se quiera medir el consumo.

Este dispositivo es capaz de hacer lecturas a partir de 0.01 Amperios. En cuanto detecta alguna corriente, el tiempo empieza a correr. Una vez que la corriente cae por debajo del umbral, el tiempo se detiene. El último valor se queda almacenado en el dispositivo, a menos que se restablezca con el botón de reset.

En cuanto a sus especificaciones, puede funcionar en un rango de 3V-9V con un error en torno a 1%. Otra característica del dispositivo es su bajo auto-consumo, ya que cualquier cantidad consumida puede hacer variar los resultados.



Figura 5.7: Funcionamiento del USB Tester.

5.1.6. Baterías

Para la alimentación de los circuitos y los dispositivos puedan ser autosuficientes conectaremos una *power bank*. Se trata de una batería externa con potencia suficiente para alimentar los dispositivos durante un cierto periodo de tiempo.

La power bank básicamente es una batería de litio controlada por un chip, de manera que detecta el dispositivo conectado y le suministra la corriente necesaria. Dispone de circuitos de protección contra sobrecarga y protección frente a cortocircuitos, de manera que el usuario no se tiene que preocupar de si el dispositivo sufrirá daños.

La power bank consta como mínimo de un puerto de entrada microUSB que se utiliza para recargarla. Además, como mínimo, tiene una salida USB para alimentar los dispositivos.

5.2. Especificaciones del IDE y Software Empleados

Para llevar a cabo el proyecto y que cumpla todos los objetivos establecidos, se necesitarán dos herramientas software. La primera herramienta es el entorno de desarrollo Arduino, que será la herramienta de programación. La segunda herramienta software será la plataforma *Thingspeak*, plataforma web para IoT encargada de recibir los datos y mostrarlos al usuario.

Además se explicarán los software necesarios para la elaboración de la memoria.

5.2.1. Entorno de Programación de Arduino (IDE)

Un IDE es un entorno de programación que ha sido empaquetado como una aplicación. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además, en el caso de Arduino, incorpora las herramientas necesarias para cargar el programa en la placa a través del puerto serie.

Los programas de Arduino están compuestos por un solo fichero con extensión "ino", aunque es posible organizarlo en varios ficheros para incluir, por ejemplo, librerías. El fichero principal siempre debe estar en una carpeta con el mismo nombre que el fichero.

Como se ha comentado anteriormente, Arduino es un entorno liviano que proporciona las herramientas básicas que se necesitan para subir, depurar y comunicarnos con la placa. Describiendo de forma breve los menús más importantes, dentro del menú *Archivo* podemos encontrar las siguientes opciones:

- **Nuevo y abrir:** Permite crear y abrir una aplicación para el dispositivo (*sketch*), respectivamente.
- Proyecto: Permite abrir un conjunto de sketch que conforma un proyecto determinado.
- **Ejemplos:** Son sketch que proporcionan ejemplos sobre el uso de la placa de Arduino. Hay ejemplos básicos que van desde hacer parpadear un led hasta hacer un servidor web básico.

El menú *Programa* proporciona las opciones de verificar y compilar el programa, incluir librerías y mostrar la carpeta del programa. Esta opción

es útil para poder buscar los sketch sin tener que navegar hasta encontrar la carpeta con los ficheros. Esta carpeta está localizada en la carpeta del usuario, dentro de una carpeta llamada ARDUINO. En esta carpeta se deberán copiar todas las librerías utilizadas.

El menú *Herramientas* incluye las siguientes opciones:

- Auto formato: Formatea el código de tal forma que sea legible.
- Archivo de programa: Comprime toda la carpeta del proyecto para guardarla en un fichero ZIP.
- Reparar codificación y recargar: Esta opción es útil para reparar ficheros con diferentes codificaciones, e.g. cuando se ha guardado un fichero con codificación UTF8 usando una codificación diferente.
- Monitor serie: Es un monitor serie que permite la comunicación con la placa. Permite tanto enviar como recibir datos a través de un puerto serie (e.g. a través del USB).
- Placa y puerto: Permite seleccionar con qué placa estamos trabajando (modelo) y en qué puerto está conectada.
- **Programador:** En caso de disponer de un programador de PIC's, se puede seleccionar uno determinado para programar nuestro chip. En el caso de las placas basadas en ESP8266, no se utiliza.
- Quemar bootloader: Permite volver a cargar el bootloader (programa encargado de iniciar el chip y periféricos, así como de iniciar el programa).

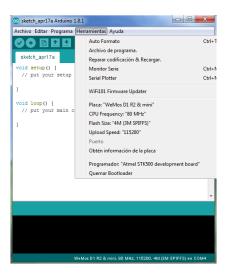


Figura 5.8: IDE de Arduino.

Por último se va a comentar los elementos de un programa en Arduino. Los sketch en Arduino tienen la estructura que se puede ver en la figura 5.9.

Como se puede ver, se tiene dos funciones principales;

- Setup: Esta función solo se ejecuta una vez al inicio del programa, por lo que es el lugar adecuado para tareas de inicialización (de variables, abrir canales de comunicación, etcétera). Es decir, es donde se deberá cargar toda la configuración y estado inicial del proyecto.
- Loop: Como se ha descrito anteriormente, el funcionamiento de Arduino consiste en ejecutar un bucle infinito. Por tanto, es el cuerpo principal del programa que se ejecutará infinitamente.

```
sketch_apr17a

void setup() {
// put your setup code here, to run once:
}

void loop() {
// put your main code here, to run repeatedly:
}
```

Figura 5.9: Sketch de Arduino.

5.2.2. ThingSpeak

ThingSpeak es una aplicación de código abierto para IoT cuya función es el almacenamiento y recuperación de datos a través de Internet utilizando el protocolo HTTP (Hypertext Transfer Protocol) o MQTT (Message Queue Telemetry Transport). ThingSpeak fue originalmente lanzado por ioBridge como un servicio de apoyo a las aplicaciones de IoT.

Como se verá a lo largo de esta sección, ThingSpeak permite la creación de diferentes canales para diferentes dispositivos. Un canal es el concepto que utiliza esta plataforma para enviar o recuperar datos, pudiendo almacenar valores en 8 campos diferentes. Además de almacenar los datos, es capaz de representarlos gráficamente de una forma amigable.

ThingSpeak ha integrado el soporte del software de cálculo numérico MATLAB. Esto permite a los usuarios de ThingSpeak analizar y visualizar datos subidos utilizando scripts de MATLAB. Actualmente, Thingspeak pertenece a MathWorks.

Como se ha dicho anteriormente, ThingSpeak es gratuito (para uso personal) con una API (*Application Programming Interface*) abierta. En el caso de las licencias gratuitas, ThingSpeak solamente permite mandar tres millones de mensajes al año por usuario, suficientes para los requisitos de este proyecto.

La comunicación con la plataforma se basa en canales. Estos almacenan los datos que se envían a través de Internet. Los datos almacenados se componen de 3 elementos:

- Ocho campos para almacenar datos de cualquier tipo.
- Tres campos para almacenar la latitud, altitud, longitud y elevación.
 Por supuesto, necesitaríamos un componente hardware que diera proporcionara esta información.
- Un campo para el almacenamiento del estado.

New Chan	nel		Help
Name Description			Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.
			Channel Settings
Field 1	Field Label 1	×	 Channel Name: Enter a unique name for the ThingSpeak channel.
Field 2		D .	 Description: Enter a description of the ThingSpeak channel.
Field 3		0	 Field#: Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
			 Metadata: Enter information about channel data, including JSON, XML, or CSV data.
Field 4		D	 Tags: Enter keywords that identify the channel. Separate tags with commas.
Field 5		0	 Latitude: Specify the position of the sensor or thing that collects data in decimal degrees. For example, the latitude of the city of London is \$1,5072.
Field 6		0	 Longitude: Specify the position of the sensor or thing that collects data in decimal degrees. For example, the longitude of the city of London is -0.1275.
Field 7		0	 Elevation: Specify the position of the sensor or thing that collects data in meters. For example, the elevation of the city of London is 35.052.
Field 8			 Make Public: If you want to make the channel publicly available, check this box.
11000			 URL: If you have a website that contains information about your ThingSpeak channel, specify the URL.
Metadata			 Video ID: If you have a YouTube ^{1w} or Vineo® video that displays your channel information, specify the full path of the video URL.
Tags			Using the Channel
	(Tags are comma separated)		You can get data into a channel from a device, website, or another ThingsSpeak channel. You
Make Public			can then visualize data and transform it using ThingSpeak Apps.
	_		See Tutorial: ThingSpeak and MATLAB for an example of measuring dew point from a weather station that acquires data from an Arduino® device.
URL			Learn More
Elevation			
Show Location	0		
Latitude	0.0		
Longitude	0.0		
Show Video	0		
	YouTube		
	⊕ Vimeo		
Video ID			
Show Status	0		
	Save Channel		

Figura 5.10: Nuevo canal en ThingSpeak.

Como se comentó, un requisito fundamental, tanto de este proyecto como de las comunicaciones IoT, es la seguridad de los datos en las comunicaciones. Así, se utilizará un canal privado de ThingSpeak. Este canal tendrá un ID Channel que será el identificador del canal. Asimismo, el canal tendrá dos claves, una para escribir datos y otra para leerlos. Así, unos usuarios (o aplicaciones) podrán escribir datos, y otros podrán leerlos sin posibilidad de modificarlos. Thingspeak incorpora una opción de generación de contraseñas, si bien se pueden cambiar en cualquier momento. En el proyecto han sido configurados de manera privada puesto que contienen información sensible, ya que un mayor o menor consumo energético podría indicar que la casa está ocupada o desocupada.

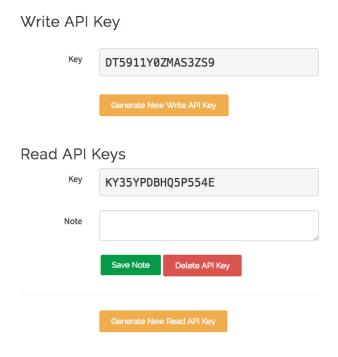


Figura 5.11: API Keys.

Además de poder crear diferentes canales, ThingSpeak tiene otras funcionalidades:

- ThingTweet: Actúa como proxy permitiendo a los dispositivos enviar actualizaciones de status a Twitter.
- TweetControl: Escucha los hashtags de Twitter y luego permite ejecutar servicios web usando ThingHTTP.
- ThingHTTP: Permite a un microcontrolador o dispositivo a bajo

nivel conectarse a un servicio web a través de Internet o la propia red.

- React: Ejecuta una acción cuando se cumple una cierta condición para los datos en el canal.
- TalkBack: Permite que cualquier dispositivo pueda actuar sobre los comandos en cola.
- **TimeControl:** Con TimeControl se puede ejecutar ThingHTTP o ThingTweet a una determinada hora, e incluso añadir un nuevo comando TalkBack.

5.2.3. Project Libre

Otro de los software utilizados para el proyecto ha sido *Project Libre*. Este programa ha sido utilizado para la elaboración de la planificación y del diagrama de Gantt. Project Libre es un software de administración de proyectos de código abierto, basado en Java, por lo que se puede utilizar en varios sistemas operativos.

Las características más importantes de Project Libre son:

- Compatibilidad con Microsoft Project 2010.
- Interfaz de usuario basado en cintas.
- Diagrama de Gantt.
- Gráficos PERT.
- Informes de uso de tareas.
- Diagrama de Estructura de descomposición del trabajo.

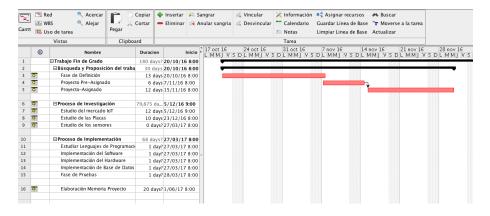


Figura 5.12: Project Libre.

5.2.4. Fritzing

Fritzing ha sido utilizado durante la elaboración de la memoria para la confección de los esquemas de conexiones de los dispositivos. Es un programa muy visual y se ha considerado que puede facilitar la comprensión del diseño e implementación de los dispositivos del proyecto.

Fritzing es un programa libre de automatización de diseño electrónico. Este programa busca ayudar en el proceso de pasar de prototipos (usando placas de prueba, por ejemplo) a productos finales. Creado bajo los principios de Processing y Arduino, permite a los usuarios documentar sus prototipos basados en Arduino y crear esquemas de circuitos impresos para su posterior fabricación. Además, cuenta con un sitio web donde se comparte diseños y experiencias. Gracias a este sitio web, no se tuvieron problemas a la hora de encontrar los esquemas de los sensores del proyecto y de la placa WeMos.

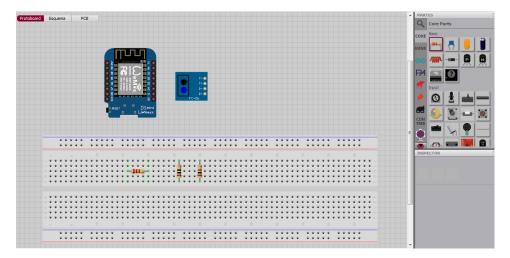


Figura 5.13: Fritzing.

Capítulo 6

Diseño e Implementación

En este capítulo se presentará el camino seguido para el funcionamiento de los dispositivos desarrollados, desde el diseño realizado hasta su programación.

6.1. Diseño Software

Para una mejor compresión de los dispositivos, se va a comenzar creando un diagrama de flujos simulando el comportamiento que tiene el microcontrolador del proyecto. Así, se intentarán esclarecer todos los pasos seguidos por el dispositivo de una manera visual y simple que hará comprender todo el proyecto de mejor forma.

El algoritmo que sigue el microcontrolador para la conexión y envío de datos a la plataforma web se puede ver en el diagrama de flujo de la figura 6.1.

Como podemos ver en el diagrama de flujo, lo primero que hará el microcontrolador será comprobar si existe acceso a Internet a través de una conexión Wi-Fi, la cuál se deberá haber configurando previamente. En el caso de que no exista esta conexión, volveremos al inicio.

Si se establece la conexión, el microcontrolador pasará a comprobar la existencia de cuenta con ThingSpeak, así como un canal creado para el sensor. Este canal deberá haber sido creado previamente. Si no encuentra el canal, volverá a comprobar si hay conexión a Internet ya que puede haberse producido algún corte a la hora de conectarse al canal.

Una vez que se tiene conexión con el canal de ThingSpeak, se le proporcionará la configuración del canal. La configuración del canal será la proporcionada a través de Wifi Manager, es decir, ID del canal, contraseña de escritura, valor del pulso, etcétera. A partir de ese momento el microcontrolador comenzará a leer los datos del sensor.

El sensor estará constantemente leyendo los datos hasta que se produzca un cambio en estos. En caso de no haber cambios, seguiremos leyendo datos, de esta manera se evita el envío de datos innecesarios a la plataforma (ahorrando además energía en el dispositivo). Por el contrario, si se han producido cambios en los datos se procederá al envío de los mismo a Thingspeak.

Llegados a este punto, solo quedará por comprobar si los datos han sido enviados de manera correcta. En ese caso, el microcontrolador tendrá que espera un intervalo de 15 segundos, antes de volver a leer datos. Después de esto, los datos enviados se podrán ver en ThingSpeak. En caso de que los datos no se envíen correctamente, se volverá al inicio del diagrama para comprobar la conexión desde el principio.

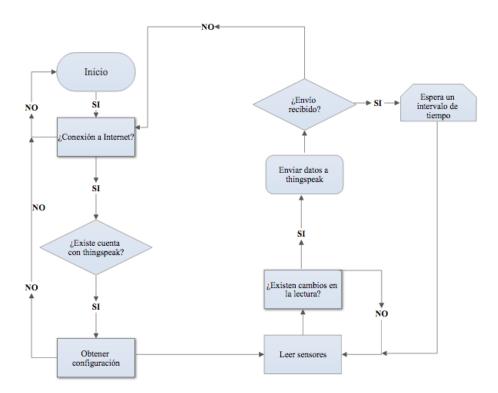


Figura 6.1: Diagrama de flujo del microcontrolador.

6.1.1. WiFiManager

WiFiManager es una librería de Arduino que permite gestionar las conexiones Wi-Fi del ESP8266 mediante un portal web. Cuando el microcontrolador funciona por primera vez (o no consigue conectarse a ningún punto de acceso configurado previamente), este comienza a funcionar como punto de acceso. El usuario se conectará al mismo usando un PC o móvil, y accederá al navegador. El dispositivo redirigirá cualquier petición HTTP a la dirección IP 192.168.4.1, donde tendrá un servidor web con las diferentes opciones para configurar una conexión Wi-Fi.



Figura 6.2: Interfaz web de WiFiManager.

En la figura 6.2 se puede ver la interfaz que proporciona el portal web de WiFiManager. En él se pueden observar las siguientes opciones:

- Configure Wi-Fi: Permite escanear las redes Wi-Fi disponibles y, tras elegir una, el dispositivo se conectará a esta tras introducir las contraseña correctamente.
- Configure Wi-Fi (No Scan): Permite conectar el dispositivo a una red sin realizar un escaneo previo. En este caso será necesario introducir el SSID de la red, la contraseña y los parámetros que se soliciten.
- Info: Proporciona información acerca del módulo Wi-Fi.
- Reset: Reinicia el ESP8266.

Para el proyecto y sus respectivas pruebas se ha utilizado la opción de Configure Wi-Fi (No Scan), que proporciona la interfaz que se muestra en la figura 6.3.

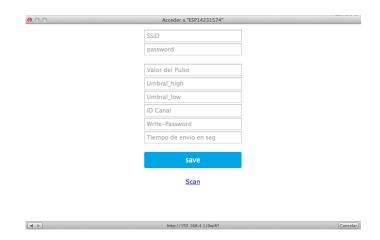


Figura 6.3: Interfaz web de Configure Wi-Fi (No Scan).

El uso de WiFiManager permite configurar los parámetros de la conexión Wi-Fi en tiempo de ejecución, no siendo así necesario incluirlos en tiempo de compilación. Para ello, es necesario incluir las librerías ESP8266WiFi, DNSServer, ESP8266WebServer y WiFiManager. Tras ello, se inicializa la librería WiFiManager y se incluye la siguiente línea dentro del SETUP:

```
wifiManager.autoConnect()
```

La función *autoConnect()* sólo permite incluir como parámetros el SSID y la clave del punto de acceso creado por el ESP8266. Para incluir otros parámetros, permitiendo así la configuración del dispositivo en tiempo de ejecución, se han utilizado las funciones que se muestran a continuación (proporcionadas por la librería WiFiManager).

Se deberán incluir llamadas a estas dos funciones por cada parámetro que se desee poner en el portal web del ESP8266.

Tras guardar los datos y realizarse con éxito la conexión Wi-Fi, los parámetros indicados serán almacenados en su memoria no volátil. De esta manera, no se necesita ninguna configuración adicional cuando el dispositivo se reinicie. En el caso de que el microcontrolador no detecte la red configurada, volverá a trabajar como punto de acceso para permitir la configuración de una nueva conexión Wi-Fi.

6.2. Diseño Hardware

En esta sección se describirá el diseño de los dispositivos respecto a sus componentes físicos, se justificarán las decisiones relacionadas con los sensores utilizados y se explicará cómo se ha llevado a cabo el montaje físico.

6.2.1. Sensor de Corriente SCT-013-000

Para el medidor de consumo eléctrico se han tenido en cuenta diversos factores para el correcto funcionamiento del mismo.

En primer lugar, se debe tener en cuenta que la pinza amperímetra va conectada a una de las entradas analógicas del WeMos. Tal como indican las especificaciones técnicas, las entradas analógicas del WeMos miden voltajes entre 0V y 5V, teniendo así 5V de margen dinámico. Además, dado que la corriente a medir será alterna, por lo que varía de -2.5V a 2.5V. Por tanto, habrá que ajustarla y elevarla 2.5V para que la tensión en la entrada analógica del WeMos varíe entre los 0V y 5V permitidos.

En la figura 6.4 se puede ver un esquema de esta transformación.

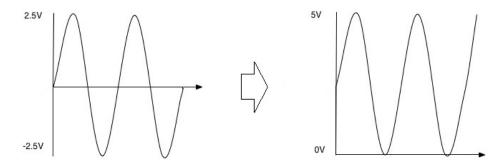


Figura 6.4: Transformación del voltaje. [23]

Para obtener esos 2.5V se va a utilizar un divisor de tensión. En la figura 6.5, se puede ver el divisor donde R1 y R2 serán de $10\mathrm{k}\Omega$, el voltaje de entrada es de 5V ya que es el que nos proporciona el WeMos. De esta manera, se obtendrá el voltaje de salida 2.5 V $((5V\times 10k\Omega)/(10k\Omega+10k\Omega)=2.5V)$, para la alimentación del sensor de corriente.

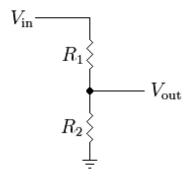


Figura 6.5: Divisor de tensión. [23]

Por último, se ha añadido un condensador electrolítico de 10μ F para filtrar los posibles ruido y conseguir que la medida sea lo más suave posible.

Así, el diseño del medidor de corriente quedará como se puede ver en la figura 6.6.

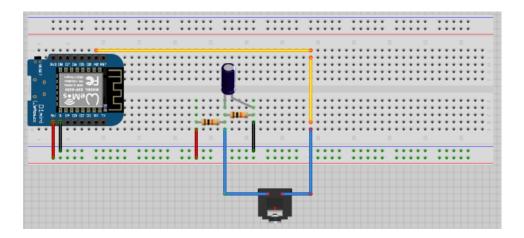


Figura 6.6: Esquema de conexiones del sensor de corriente.

La conexión del sensor de corriente se realiza a través de un conector Jack hembra 3.5 mm. Este conector sigue el esquema de la figura 6.7. Como se puede apreciar, solamente es necesario conectar dos patillas de las tres que tiene el adaptador. Una salida irá conectada directamente al pin A0 del WeMos y la otra se conectará a la salida del divisor de tensión para su alimentación.

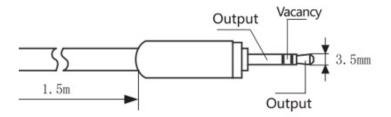


Figura 6.7: Esquema del conector Jack 3.5mm. [23]

6.2.2. Sensor infrarrojo reflexivo TCRT5000L

Para el diseño e implementación del dispositivo que monitoriza el consumo de electricidad, agua y gas se ha seguido el esquema que aparece en la figura 6.8.

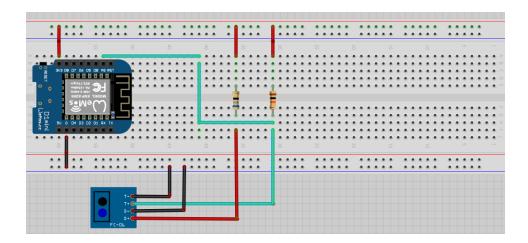


Figura 6.8: Esquema de conexiones del sensor reflexivo TCRT5000L.

Además del WeMos y el sensor infrarrojo reflexivo, se han utilizado dos resistencias; una de 10 k Ω y otra de 68 Ω . Estas resistencias no ha sido escogidas de forma aleatoria sino que han sido calculadas para el buen y seguro funcionamiento del dispositivo.

Para no romper el diodo LED debido a una corriente excesiva, se ha necesitado poner una resistencia en serie, llamada R_{LED} . Esta resistencia estará conectada a 3.3 V. Para hacer los cálculos se ha escogido de la hoja de características del sensor el voltaje y la corriente típica, cuyos valores son 1.25 V y 60 mA, respectivamente. El valor de la resistencia mínima (que

permite una corriente máxima 60 mA) se calculará de la siguiente manera:

$$(3.3V - 1.25V)/0.06A = 34.16\Omega$$

Por seguridad y debido a que los valores de la hoja de características se refieren al máximo absoluto, se tomará una resistencia un poco mayor, de $68~\Omega$.

La otra resistencia escogida será de $10k\Omega$. Por la hoja de características se puede ver que la corriente del colector máxima es de 0.5 mA cuando se tiene 5V y el LED está trabajando a 60 mA. Por tanto, con esta resistencia de 10 k Ω se consigue que la corriente máxima a través del fototransitor sea de 0.5 mA.

Como ya se dijo en el capítulo 5, el sensor TCRT5000L cuenta con 4 patillas: ánodo, cátodo, colector y emisor. Las dos primeras patillas corresponden al diodo LED y las otras dos al receptor.

Como se puede ver, el circuito de implementación del dispositivo no es muy complejo. La resistencia R_{LED} se conectará a 3.3 V y al ánodo del diodo LED, el cuál deberá estar constantemente recibiendo corriente debido a que siempre estará emitiendo luz infrarroja. El colector irá conectado a la resistencia de 10 k Ω y ésta se conectara a 3.3 V. Como pasa con el diodo LED, el detector deberá estar constantemente capturando la luz reflejada en el objeto. Además el colector también irá conectado al pin analógico para que se muestre la salida por pantalla, a través del monitor serie de Arduino o bien enviando los pulsos detectados a Thingspeak.

6.3. Implementación

Una vez hecho el diseño, en esta sección se describirá la implementación realizada.

6.3.1. Sensor de corriente

Como se ha comentado en la sección de diseño, tan solo se ha necesitado dos resistencias y un condensador electrolítico para llevar a cabo el funcionamiento del medidor de corriente. Como se ha podido ver, el montaje físico del dispositivo no es muy complejo. Sin embargo, lo que añade complejidad es el cálculo de las resistencias y la calibración del mismo.

Para la calibración y medida correcta de la corriente eléctrica, en el

código se va a utilizar la librería *Emonlib* que se utilizará para dos funciones, cuyos parámetros deberán ser configurados adecuadamente.

La primera función, emon1.current(pin, calibración), permitirá asignar un pin de entrada y un valor de calibración. El primer parámetro será el pin analógico del WeMos, A0. Para el cálculo del parámetro de calibración, se ha realizado un cálculo teórico y después se ha ajustado de forma empírica. Este parámetro será explicado con más detalle en el capítulo 7.

La segunda función, emon1.calcIrms(muestras), se utiliza para el cálculo de el valor cuadrático medio de la corriente, i.e. I_{RMS} . En esta función el único parámetro que se debe configurar es el número de muestras que se va a utilizar. Este número puede variar ya que no es más que el número de muestras que utilizar para realizar la medida. En este caso se ha escogido 1480 muestras que es número considerable de muestras y no produce retardos, es decir, un aumento de las muestras puede producir retardos entre medida y medida. Al igual que una disminución de las muestras puede provocar que la medida no sea tomada de forma óptima.

Para esclarecer el funcionamiento de esta última función, se va a enumerar la secuencia seguida por la misma para el cálculo de la corriente RMS.

- 1. Muestrea un valor.
- 2. Filtra la muestra por software normalizada.
- 3. Realiza la raíz cuadrada de la muestra filtrada.
- 4. Suma este resultado a los anteriores.
- 5. Tras realizar el proceso, aplica la calibración anterior sobre la lectura que se está realizando.
- 6. Por último, calcula la corriente RMS y la devuelve.

En la figura 6.9, se puede ver el medidor de corriente ya implementado. Además, en la tabla 6.1 se pueden ver todos los pines usados y su función.

PinOut	Conexión	
A0	Salida del sensor	
D1-D8 (pines digitales)	No utilizados	
5V	Alimentación	
G	Masa (0V)	

Tabla 6.1: Cuadro-resumen de las conexiones del sensor SCT-013-000.

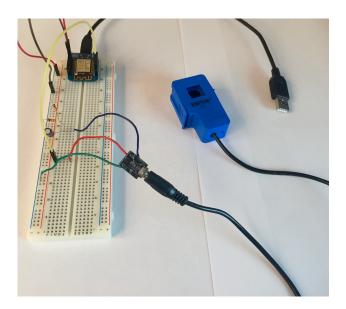


Figura 6.9: Medidor de energía eléctrica.

6.3.2. Sensor Óptico Reflexivo

Tal como se ha visto, el sensor infrarrojo reflexivo TCRT5000L utilizará el pin A0 (lectura de datos), el pin de alimentación de 3.3V y el pin de tierra de la placa WeMos. En la figura 6.10 se muestra la implementación llevada a cabo tal y como se ha explicado en la sección de diseño.

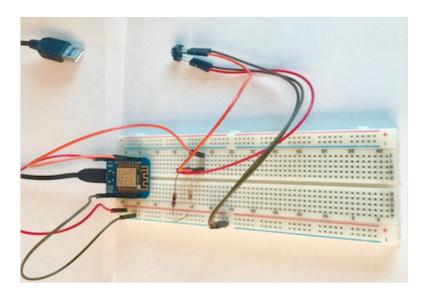


Figura 6.10: Dispositivo de monitorización de agua y gas.

En la tabla 6.2 se muestran los pines utilizados.

PinOut	Conexión
A0	Colector del transistor
D1-D8 (pines digitales)	No utilizados
3.3V	Alimentación
G	Masa (0V)

Tabla 6.2: Cuadro-resumen de las conexiones del sensor TCRT5000L.

A pesar de que la placa WeMos se alimenta con 5V, este sensor utiliza 3.3V para evitar que sature y de esta manera tener unas medidas más precisas. Además, deberá estar siempre alimentado para poder detectar un pulso en cualquier momento, con lo que su consumo será mayor con respecto al medidor de corriente. Todos los detalles del consumo quedan detallados el capítulo 7.

Capítulo 7

Pruebas y Resultados

Una vez llevados a cabo el diseño y la implementación de los dispositivos desarrollados, se realizarán las pruebas con ambos dispositivos para comprobar que cumplen los objetivos iniciales.

7.1. Pruebas de calibración

Antes llevar a cabo las pruebas finales, se realizará las calibraciones necesarias para que el funcionamiento sea el correcto. Estas calibraciones permiten ajustar los parámetros del código comentados en el capítulo 6.

7.1.1. Sensor de Corriente

En el capítulo 6 se comentó que para la implementación del medidor de corriente se ha utilizado la librería *Emonlib*. Para el cálculo de la corriente RMS, primero, se necesita inicializar la librería con la función *emon1.current* (pin, calibración) en la que el parámetro de calibración dependerá del tipo de SCT.

Este parámetro refleja el número de espiras divido entre su resistencia interna. Según la hoja de características, el número de espiras es 2000 y no tiene resistencia interna. A la hora de la implementación, se tuvo que suponer que la resistencia interna sería muy pequeña ya que el medidor de corriente daba datos erróneos. Por tanto, se tuvo que buscar el número de calibración de forma empírica.

Las pruebas de calibración consistieron en medir un dispositivo eléctri-

co del que se conocía su consumo real. Así, cuando el dispositivo estaba apagado, el medidor de corriente debía de marcar cero. Del mismo modo, cuando estaba enchufado debería dar el valor de consumo conocido. De esta manera, se pudo detectar que el SCT tenía una resistencia interna que no era considera en la hoja de características.

7.1.2. Sensor Infrarrojo Reflexivo

Este dispositivo requiere de unos pequeños ajuste cada vez que se cambie de tipo de contador debido a que los umbrales de cada tipo contador son únicos. Es decir, los valores para detectar los pulsos en cada tipo contador son diferentes unos de otros. También se deberá ajustar el valor de consumo asociado a cada pulso en función del tipo de contador. Por ejemplo, en los contadores de agua probados cada pulso corresponde a un consumo de un litro. Las variables que se deberán ajustar son las que vemos en la figura 7.1.

```
const float unidad_C = 0.25;
unsigned int umbral_high = 750;//685
unsigned int umbral low = 740; //673
```

Figura 7.1: Calibración del sensor TCRT5000L.

Para calibrar estos parámetros, se ha construido un programa que leía los valores analógicos del sensor. Una vez construido este programa de prueba, se colocó el dispositivo en el contador a probar y se observaron los valores máximo y mínimo para así poder poner el rango de valores. El rango calculado se incrementa un cierto porcentaje por seguridad. Además, estos valores se han utilizado para evitar la detección incorrecta de pulsos consecutivos. Así, si se detecta un valor mayor que umbral-high significará que hay un nuevo pulso y si el valor encontrado está entre el umbral-high y el umbral-low significará que sigue en el mismo pulso y no deberá tenerlo en cuenta.

7.2. Pruebas del Sensor Infrarrojo Reflexivo

En esta sección se describen las pruebas realizadas en el entorno doméstico para comprobar el funcionamiento del sensor infrarrojo reflexivo.

Este sensor, como se ha comentado durante todo el trabajo, es capaz de monitorizar el consumo de electricidad, agua y gas. Por tanto, se llevará a cabo una monitorización del consumo diario de electricidad, agua y gas.

7.2.1. Prueba de consumo de agua

En esta prueba se ha monitorizado todo el consumo de agua de una casa durante un cierto período de tiempo.

Como se muestra en la gráfica 7.2, lo único que se debe hacer es colocar el dispositivo en el contador de agua. El sensor se colocará en una de terminada cifra, e.g en cero. De esta manera, se obtendrá un pulso cada vez que la aguja del contador de una vuelta completa y pase por cero. Finalmente, se deberá de ajustar de la forma que se indica en la sección de calibración.



Figura 7.2: Monitorización del consumo de agua.

En esta prueba se midió el consumo de agua a lo largo de un día. Además, se estuvo atento a todo el consumo de agua para poder cerciorar que el dispositivo tomaba las medidas de manera correcta.

En las gráficas 7.3 y 7.4 se pueden ver el consumo de agua en litros por minutos y el número de litros totales consumidos durante la prueba, respectivamente.

Como se puede ver en ambos gráficos, hay tres picos de mayor consumo de agua. Estos corresponden al agua utilizada para la ducha. Un dato inesperado fue el consumo tan elevado que puede llegar a tener la cisterna del WC a lo largo del día, ya que en la casa donde se realizó la prueba no había cisterna de bajo consumo. Todos los pequeños consumos a lo largo del día que se pueden ver en la gráfica son causados por este motivo. Cada descarga supone sólo 6 litros, pero utilizada muchas veces a lo largo del día puede provocar un gran consumo.

En el horario de la comida y cena también se aprecia un aumento del consumo debido al uso del agua para cocinar, beber, fregar platos, etcétera.

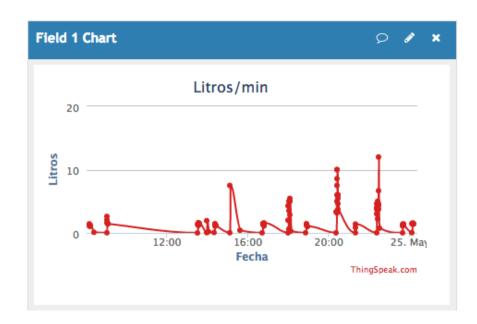


Figura 7.3: Gráfica del consumo de agua.



Figura 7.4: Gráfica de los litros de agua consumidos.

En la tabla 7.1, se pueden ver los datos aproximados obtenidos durante la prueba.

Actividad	Consumo (Litros)
Ducha	70
WC	60
Aseo personal	12
Fregar Platos, cocinar, beber	18
Total	≈ 160

Tabla 7.1: Tabla-resumen de consumo de agua.

7.2.2. Prueba de Consumo de Gas

En la siguiente prueba se va a llevar a cabo una monitorización del consumo del gas durante un cierto período de tiempo.

La prueba se va a realizar con el mismo dispositivo que se ha utilizado para hacer la prueba del consumo del agua. El modo de funcionamiento será el mismo. El dispositivo será colocado de la manera que vemos en la figura 7.5 y calibrado como se dijo en la sección de calibración.

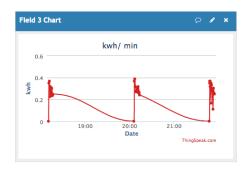


Figura 7.5: Monitorización del consumo de gas.

Al igual que se hizo un seguimiento para cerciorarnos que el consumo de agua era el correcto, el consumo del gas también fue seguido durante todo el tiempo que duró la prueba. Una vez hechas las pruebas, los resultados obtenidos son los que vemos en la figura 7.6.

Dado que el gas solo se usa para el agua caliente, el consumo medido es

bastante bajo.



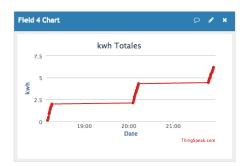


Figura 7.6: Gráfica del consumo de gas en kWh.

En los resultados se puede apreciar que los valores están escalados a kwh. Esto se debe a que en las facturas se escala a esta unidad, aunque el contador mide en dm^3 .

Como muestran los resultados en la figura 7.6, el consumo se produce únicamente en 3 puntos del día, debido a las duchas realizadas.

Para comprobar la precisión de los dispositivos, se comprobó el consumo que había antes y después de realizar la prueba. Una vez realizada la prueba pudimos comprobar que los resultados por los dispositivos eran muy aproximados a los calculados de forma teórica.

De esta manera, se puede decir que los resultados obtenidos en las pruebas finales han sido de un grado satisfactorio.

De este modo, se ha comprobado que el dispositivo es capaz de monitorizar el consumo energético y enviarlo a la plataforma ThingSpeak en tiempo real.

7.2.3. Prueba de Consumo de Electricidad

En esta última prueba se va a llevar a cabo la monitorización del consumo eléctrico.

Para poder realizar esta prueba se ha tenido que poner el móvil como punto de acceso debido a que no se tenia una conexión Wi-Fi en la zona de contadores eléctricos. A causa de esto, la prueba se ha llevado a cabo en un periodo de tiempo menor que las anteriores pruebas.

El dispositivo quedó colocado de la forma que se muestra en la figura

7.7. La calibración se llevó a cabo tal como se describió en la sección de calibraciones.



Figura 7.7: Monitorización del consumo eléctrico.

Un dato importante a tener en cuenta para esta prueba es la rapidez con la que se producen los pulsos, así como el retardo que produce la red al enviar los datos a la plataforma web. Por eso, en esta prueba se mandarán los datos cada cierto tiempo. Este tiempo será variable ya que se puede definir en la interfaz de Wifi Manager.

Cuando el dispositivo está mandado datos, este no detecta pulsos. Así, en el código de este dispositivo se han llevado a cabo diversos ajustes para estimar los pulsos perdidos durante la transmisión, considerando el consumo real justo antes de la misma.

Los resultados obtenidos tras la prueba se pueden ver en la figura $7.8~\mathrm{y}$ 7.9.

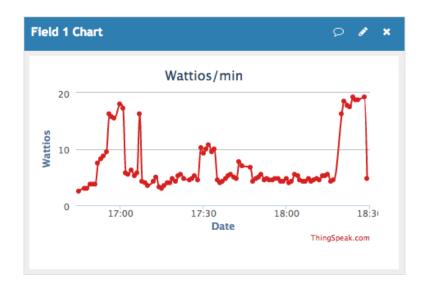


Figura 7.8: Gráfica del consumo eléctrico en Watt.

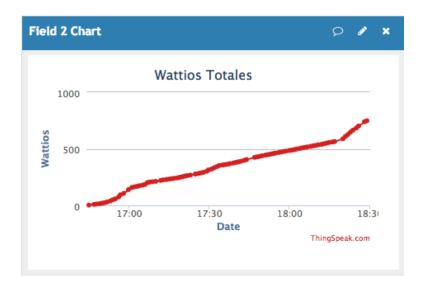


Figura 7.9: Gráfica del consumo total de Watt.

Además del consumo habitual en la casa, se procedió a realizar diferentes consumos controlados. Para comprobar que el dispositivo era capaz de monitorizar tanto altos consumo como bajos consumos, se procedió a enchufar y hacer uso de diferentes dispositivos eléctricos, como un microondas, un secador, una cafetera... En la figura 7.8 se puede ver el flujo de corriente producido durante la prueba.

Siendo consciente del posible consumo que se estaba produciendo y el

momento en que se estaba produciendo, se puede afirmar que los resultados obtenidos han sido de un grado muy satisfactorio. Al igual que en las pruebas anteriores, se miró el consumo antes y después de la prueba.

Finalmente, se puede decir que se tiene un dispositivo capaz de monitorizar el consumo eléctrico, de agua y de gas.

7.3. Pruebas del Sensor de Corriente

En esta sección se va a realizar un estudio sobre el consumo de diferentes dispositivos eléctricos. Se llevará a cabo una comparación sobre el consumo teórico y el medido empíricamente.

Para la prueba se va a utilizar un dispositivo eléctrico con tres potencias diferentes. El dispositivo que se va ha utilizar para la prueba es un calefactor del fabricante fm. Para una mejor comparación de los resultados y poder comprobar que el dispositivo eléctrico consume realmente lo que dice el fabricante, en la tabla 7.2 se pueden ver las especificaciones técnicas del mismo.

Características	Valor
Modelo	H-2000
Voltaje de entrada	230 V
Potencia máxima	1200 W
Potencia	400 ó 800 W
Frecuencia de Entrada	50 Hz

Tabla 7.2: Especificaciones Técnicas.

Toda la información y características técnicas han sido obtenidas de la página oficial del fabricante y hoja de características del mismo [20].

Llegados a este punto, se va a poner en marcha el medidor de consumo eléctrico. Como se ha ido diciendo a lo largo de todo el proyecto, el dispositivo es no invasivo por lo que la puesta en marcha del mismo será muy sencilla, tal como se ilustra en la figura 7.10.

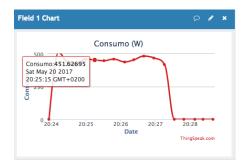


Figura 7.10: Medidor de consumo eléctrico.

Como se puede ver, en este caso el dispositivo está conectado a una batería externa. Para poder ver los valores medidos se utilizará tanto Thingspeak como el monitor serie de Arduino. Una vez hecho esto, los resultado obtenidos son los siguientes:

■ Prueba de 400 W

En esta prueba se ha puesto el dispositivo a trabajar a 400 Wattios y los resultados obtenidos por el medidor de corriente son los que se muestran en las figuras 7.11 y 7.12 en Thingspeak y Arduino, respectivamente.



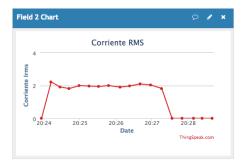


Figura 7.11: Prueba de 400 Wattios. Thingspeak.

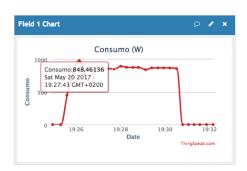
Irms: 2.07
Consumo: 464.82
Irms: 2.02
Consumo: 497.97
Irms: 2.17
Consumo: 476.02
Irms: 2.07
Consumo: 476.02
Irms: 2.07
Consumo: 480.62
Irms: 2.09
Consumo: 494.42
Irms: 2.15
Consumo: 484.28
Irms: 2.11

Figura 7.12: Prueba de 400 Wattios. Arduino.

Como se puede ver, tanto en la gráfica de Thingspeak como en los datos mostrados por el monitor serie de Arduino difieren un poco con los proporcionados por el fabricante. Esta diferencia de medida, aunque no es elevada, puede deberse a diferentes factores del dispositivo construido y debería resolverse mediante una calibración más precisa. También pueden deberse a que los valores de consumo del calefactor no son exactamente los teóricos sino cercanos.

■ Prueba de 800 W

En esta segunda prueba, se va a utilizar el mismo dispositivo pero esta vez con una potencia de 800 W, según especificaciones. Los resultados son los mostrados en las figuras 7.13 y 7.14.



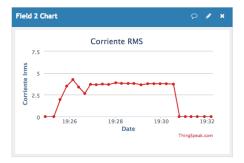


Figura 7.13: Prueba de 800 Wattios. Thingspeak.

```
Irms: 3.66
Consumo: 811.33
Irms: 3.53
Consumo: 804.96
Irms: 3.50
Consumo: 833.98
Irms: 3.63
Consumo: 857.87
Irms: 3.73
Consumo: 849.12
Irms: 3.69
Consumo: 851.01
Irms: 3.70
Consumo: 841.60
Irms: 3.66
```

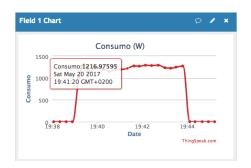
Figura 7.14: Prueba de 800 Wattios. Arduino.

Al igual que en la prueba anterior, los resultados obtenidos son satisfactorios aunque presentan cierto error con respecto a las especificaciones del fabricante.

Al igual que en la prueba anterior, este error puede ser debido a diversos factores del dispositivo construido o al propio calefactor.

■ Prueba de 1200 W

En esta última prueba se va a medir un consumo alto del dispositivo eléctrico. De esta manera, se comprobará que el medidor de corriente funciona tanto para consumos alto como para consumos más pequeños. En las figuras 7.15 y 7.16 se pueden ver los resultados obtenidos para un consumo teórico de 1200 Wattios.



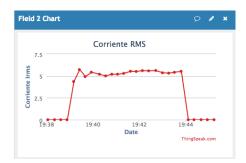


Figura 7.15: Prueba de 1200 Wattios. Thingspeak.

Irms: 5.28
Consumo: 1229.98
Irms: 5.35
Consumo: 1236.34
Irms: 5.38
Consumo: 1237.54
Irms: 5.38
Consumo: 1256.05
Irms: 5.46
Consumo: 1249.43
Irms: 5.43
Consumo: 1237.47

Irms: 5.38

Figura 7.16: Prueba de 1200 Wattios. Arduino.

Finalmente, a la vista de los resultados en las diferentes pruebas, se ha comprobado que el dispositivo construido consigue unos resultados satisfactorios.

Este dispositivo se podría utilizar para comprobar que los electrodomésticos consumen lo que indican las especificaciones del fabricante, además de para medir el consumo que puede ser variable en función de muchas circunstancias (e.g. un frigorífico cambiará su consumo en función de la temperatura seleccionada, si se abre mucho o poco sus puertas, etcétera).

Además de esto, el medidor de corriente se podría utilizar en modo monitorización. Por ejemplo, en caso de que frecuentemente se nos olvide apagar una plancha, un calefactor, el aire acondicionado, ... Se puede saber en cualquier momento a través de la plataforma IoT si el dispositivo está apagado o encendiendo. De esta manera, se podría avisar a alguien para apagarlo.

7.4. Pruebas de Consumo

Uno de los objetivos principales del proyecto era hacer un dispositivo de bajo consumo. De esta manera, el dispositivo podría estar en funcionamiento el máximo tiempo posible.

En esta última sección se van a realizar diferentes pruebas para determinar si los dispositivos construidos en el proyecto son de bajo consumo o no. Con estas pruebas lo que se va a conseguir es optimizar al máximo el consumo del dispositivo sin que se produzcan pérdidas de datos.

Para llevar acabo la ejecución de dichas pruebas se va a utilizar un USB

Tester. Este dispositivo fue descrito en el capitulo 5. Además, para medir el tiempo se ha utilizado un cronómetro para una mayor precisión de los resultados.

7.4.1. Sensor de Corriente

En este apartado se va a llevar a cabo las pruebas de consumo para el medidor de corriente SCT-013.

La prueba consistirá en poner en funcionamiento el medidor de corriente y con el USB Tester comprobar su consumo. De esta manera, se podría estimar el tiempo que puede estar en funcionando el dispositivo con la batería. En la prueba mediremos el tiempo que tardará en consumir 20 mAh.

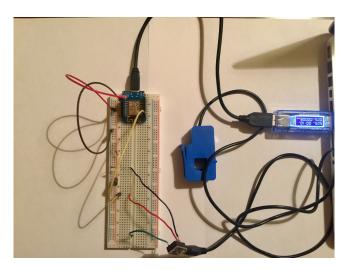


Figura 7.17: Prueba de consumo. Medidor de corriente

Una vez ejecutada la prueba, el sensor tarda 24 minutos y 12 segundos en consumir 20 mAh, lo que da un consumo de 0.82 mAh por minuto. Es decir, el medidor de corriente tendrá un consumo de 49.58 mAh a la hora.

Por tanto, queda comprobado que el medidor de corriente construido en el proyecto es de bajo consumo, cumpliendo así con uno de sus principales objetivos.

7.4.2. Sensor Infrarrojo Reflexivo

• Primera Prueba

La primera prueba que se realizó del dispositivo fue medir el consumo cuando este envía datos constantemente, cada 15 segundos.

La prueba consistió en medir el tiempo que tardaba en consumir 20 mAh. Como se ha dicho anteriormente, para medir el tiempo se ha utilizado un cronómetro que es más preciso que USB Tester, en cuanto a tiempo se refiere.

En esta ocasión los resultados obtenidos muestran que se tardan 15 minutos y 27 segundos para consumir 20 mAh. Por lo que se obtiene un consumo de 1.294 mAh por minuto. Es decir, un consumo de 77.66 mAh en una hora.

La razón por la que el dispositivo no se implementó de esta manera fue porque el dispositivo siempre está enviando datos haya o no nuevos pulsos. De esta manera, puede que esté enviando datos repetidos como vemos en la figura 7.18. Por tanto, está llenando las gráficas de datos repetidos. Además genera un consumo más elevado.

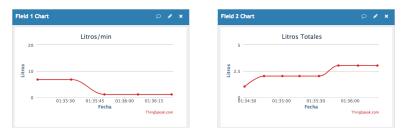


Figura 7.18: Segunda prueba de consumo. Sensor Reflexivo.

• Segunda Prueba

En la segunda prueba de consumo del sensor infrarrojo reflexivo, se dispondrán de las mismas condiciones que en la prueba anterior. La única diferencia es que en esta prueba se añadirá un filtro para no enviar datos repetidos.

El sistema para la prueba de consumo será el mismo, mediremos el tiempo que tardará el dispositivo en consumir 20 mAh.

Los resultados obtenidos en dicha prueba son de un consumo de 20 mAh en un tiempo de 15 minutos y 54 segundos, con lo que tenemos un consumo de 1.257 mAh por minuto. Es decir, un consumo de 75.47 mAh en una hora.

Los resultados en esta prueba son muy similares a la anterior. Solo hay una pequeña diferencia entre ambas, en cuanto a consumo.

Debido al pequeño ahorro de energía obtenido en esta prueba y al envío solo de nuevos datos, este método de implementación del código será el elegido para el proyecto. Los resultados obtenidos con esta opción se pueden ver en las gráficas obtenidas en las pruebas realizadas con los dispositivos en la sección anterior.

Capítulo 8

Conclusiones

En este último capítulo se hará un análisis sobre el trabajo realizado a lo largo del proyecto. Se expondrán las principales conclusiones obtenidas en su desarrollo. Asimismo, se comentarán qué trabajos futuros se podrían realizar para mejorar los dispositivos del proyecto.

8.1. Trabajo y Conclusiones

8.1.1. Trabajo Realizado

En este proyecto se distinguen dos fases claramente diferenciadas. En la primera fase se ha realizado un estudio que ha abarcado los siguientes puntos:

- Estudio del mercado: En esta primera parte se ha visto como ha ido evolucionando el mercado de Iot. El estudio se ha centrado en los dispositivos que pueden competir con los realizados en el proyecto.
- Estudio de las diferentes placas: En segundo lugar se hizo un estudio de las diferentes placas del mercado actual que utilizaban el microcontrolador ESP8266. Se analizaron sus prestaciones para elegir la más adecuada para el proyecto.
- Estudio de los componentes hardware y software: Para terminar con esta primera fase, se llevó a cabo un análisis de los diferentes componentes tanto hardware como software del mercado. De esta manera, se intentaban cumplir los objetivos principales del proyecto.

La segunda fase de este proyecto consta de tres partes. En ellas se han implementado las herramientas necesarias para lograr que los dispositivos funcionen correctamente. Estas partes son:

- Diseño e implementación de las herramientas software: Tras el análisis de la fase anterior, se llevó a cabo la implementación del código del proyecto en el lenguaje que se consideró más óptimo y compatible para la placa.
- Diseño e implementación de los dispositivos hardware: Una vez conocidas todas las especificaciones de los componentes, se llevó a cabo el diseño y el montaje de los diferentes dispositivos.
- Realización de las pruebas finales: Para la finalización de este apartado se han realizado pruebas consistentes en la monitorización de la energía eléctrica, el agua y el gas.

8.1.2. Conclusiones

Una vez terminado el proyecto y echando un vistazo atrás hacia los objetivos propuestos, siendo estos la simplicidad, el bajo coste, bajo consumo y la facilidad de uso, se puede determinar que se han cumplido con éxito dichos objetivos.

Destacamos que a los largo de todo el proyecto se han tenido diferentes dificultades. Todas estas dificultades han sido solventadas por medio de la investigación en el campo que les compete, siendo esto un éxito personal en el avance de la formación académica.

Personalmente, este proyecto ha supuesto todo un reto académico. El desarrollo desde su fase inicial hasta la finalización del proyecto ha implicado poner en práctica las diferentes competencias que componen el conjunto de las asignaturas del Grado de Ingeniería de Tecnologías de Telecomunicación. Además de una satisfacción personal, nos ha hecho pensar en la importancia que tiene el ahorro energético, no solo en la parte económica sino también en la parte que abarca el cuidado del medio y su desarrollo sostenible. Como hemos podido comprobar a lo largo del proyecto, son pequeños detalles los que se deben cuidar para conseguir este ahorro energético.

Y para concluir, hemos de decir que los IoT, en un futuro no muy lejano, estarán a la orden del día. Se estima que para 2020 cada persona cuente con más de 8 dispositivos interconectados. Esto se debe a que la mayoría de los avances tecnológicos necesarios para su implantación ya han sido realizados.

Conclusiones 73

8.2. Lineas Futuras

Hoy en día, la tecnología camina a pasos agigantados. Uno de los principales problemas de todos los dispositivos es la posibilidad de caer en desuso o quedarse obsoletos.

En este capítulo se van proponer posibles mejoras para que los dispositivos del proyecto puedan ser usados durante mucho tiempo y no caigan en desuso. Además, se expondrán algunas nuevas tecnologías que han surgido en estos últimos años para los IoT. Estas tecnologías harán de los dispositivos un mejor producto para los usuarios dotándolos de diferentes ventajas.

8.2.1. Autonomía

Uno de los principales problemas de los dispositivos eléctricos móviles es la batería. En este proyecto los dispositivos han sido alimentados con una batería externa. Como se ha podido ver a lo largo del proyecto, esas baterías se deben cargar o remplazarse por otra cada cierto tiempo.

Una posible mejora para los dispositivos, en cuanto a autonomía se refiere, sería añadirle un módulo (shield) de batería. Como se muestra en la figura 8.1, la placa se compone de dos adaptadores, uno para enchufar la batería y el otro para la conexión USB. Con ello nos permitirá alimentar la placa mediante la batería y cargar la batería.

Con este módulo de batería lo que se conseguiríamos sería aumentar la autonomía de los dispositivos. En cualquier caso, cada cierto tiempo se debería recargar las batería o cambiarla por otra cargada.



Figura 8.1: Módulo de batería para WeMos D1 Mini. [8]

8.2.2. Tecnologías para IoT

Siempre que se habla de redes inalámbricas el primer pensamiento que se viene a la mentes es Wi-Fi. Como se puede ver en la actualidad, el desarrollo a pasos agigantados de los IoT han impulsado el desarrollo de nuevas redes y tecnologías.

En la actualidad, existen multitud de tecnológicas de transmisión inalámbricas de datos como Wi-Fi, Bluetooth, 2G, 3G, 4G..., pero a todo esto se deben añadir nuevas redes emergentes que utilizan la "banda blanca" liberada por la televisión digital terrestre. Estas nuevas redes implementan soluciones de acceso IoT en áreas extensas.

Dependiendo de la aplicación, los factores como el alcance, velocidad de transferencia, seguridad, potencia y autonomía dictarán cuál es la mejor alternativa a la hora de elegir una red inalámbrica u otra.

Como se ha podido ver en algunas de las pruebas realizadas en el proyecto, la tecnología Wi-Fi no era capaz de llegar a ciertos sitios. Por tanto, sería aconsejable estudiar el empleo de algunas de estas tecnologías para una posible mejora y evolución de los dispositivos del proyecto.

Estas son algunas de las principales tecnologías de comunicación que pueden elegir los desarrolladores.

- 1. **ZigBee:** Es una tecnológica inalámbrica más centrada en aplicaciones domóticas e industriales. Se basan en la tecnológica IEEE 802.15.4, una tecnología de red inalámbrica que opera a 2.4 GHz en aplicaciones que requieren comunicaciones con baja tasa de envío de datos dentro de áreas delimitadas. Tiene un alcance de 100 metros.
 - Alguna de las ventajas que presenta esta tecnología son el bajo consumo, seguridad, robustez, alta escalabilidad y gran capacidad para soportar un gran número de nodos.
- 2. **Thread:** En la actualidad, el protocolo de red más innovador basado en IPv6 es Thread. Diseñado para domótica, está basado en 6LowPan. Este no es protocolo de aplicaciones IoT, sino que se diseñó como un complemento Wi-Fi. Es una solución resistente para basada en IP para aplicaciones IoT. Trabaja en la banda de 2.4 GHz. El alcance es de hasta 35 Km para GSM y 200 Km para HSPA.
- 3. Sigfox: Sigfox en términos de alcance está entre la tecnología Wi-Fi y la comunicación móvil. Utiliza bandas ISM, que se pueden utilizar sin necesidad de adquirir licencias.

Conclusiones 75

4. **Neul:** Es similar a Sigfox y funciona en la banda sub-1GHz. Neul aprovecha pequeños fragmentos de la "banda blanca" de las estaciones de TV para ofrecer alta escalabilidad, amplia cobertura y bajos costes. Se pueden alcanzar hasta 10 Km en entornos urbanos.

La tecnología de comunicaciones que utiliza se llama *Weightless*, que es una nueva tecnología de red inalámbrica ampliada diseñada para aplicaciones IoT que compite contra las soluciones GPRS, 3G, CDMA v LTE WAN.

- 5. LoraWAN: Esta tecnología se parece en algunos aspectos a Sigfox y Neul. LoraWAN está diseñada para implementar redes de área amplia (WAN) con características especificas para soportar comunicaciones móviles, bidireccionales, económicas y seguras para aplicaciones de IoT,M2M, ciudades inteligentes... Optimizada para bajo consumo de energía y ofrecer amplias redes con millones de dispositivos. Se pueden alcanzar hasta 5 Km en entornos urbanos.
- 6. Z-Wave: Es una tecnología RF de bajo consumo diseñada inicialmente para productos de domótica. Optimizado para la comunicación fiable de baja latencia de pequeños paquetes de datos, alcanza velocidades de datos de hasta 100 kbit/s. Opera en la banda sub-1 GHz y es robusta frente a interferencias de Wi-Fi y otras tecnologías inalámbricas en la banda de 2.4 GHz. Muy escalable y compatible con redes de topología mallada.

Bibliografía

- [1] Casadomo.com, Smart Building Conference. 3^a ed. ISE: 2010.
- [2] THE MOST WIDELY USED COMPUTER ON A CHIP: THE TMS 1000, Stan Augarten
- [3] Atmel's Self-Programming Flash Microcontrollers, Odd Jostein Svendsli
- [4] A Brief History of Computing: ENIAC and EDVAC, Jack Copeland
- [5] Building Internet of Things with the Arduino, *Charalampos Doukas*. Volumen 1, 2012.
- [6] Arduino. 2017 Disponible en https://www.arduino.cc/, (accedida el 21 de Junio de 2017).
- [7] RASPBERRY PI USER GUIDE, Eben Upton, Gareth Halfacree. Disponible en http://www.cs.unca.edu/~bruce/Fall14/360/RPiUsersGuide.pdf, (accedida el 21 de Junio de 2017).
- [8] D1 MINI [WEMOS ELECTRONICS]. Disponible en https://wiki.wemos.cc/products:d1:d1_mini (accedida el 21 de junio de 2017).
- [9] Efergy.com, Medidores de Consumo (2017). Disponible en http://efergy.com/es/engage-hub, (accedida el 21 de Junio de 2017).
- [10] WATTIO.COM, Pack Energy-8 (2017). Disponible en https://wattio.com/es/tienda/pack-energy-8, (accedida el 21 de Junio de 2017).
- [11] MICROPROCESADORES Y MICROCONTROLADORES, Hector Uriel Villamil González. Universidad Nacional Abierta y a Distancia. 2009.
- [12] IOT ANALYTICS THINGSPEAK, *ThingSpeak. 2017. Disponible en* https://thingspeak.com/, (accedido el 21 de Junio de 2017).

78 BIBLIOGRAFÍA

[13] CLIENSOL.ES, Medidor de consumo de agua, GaSmart.. Disponible en http://www.cliensol.es/medir-consumo-agua.html, http://www.cliensol.es/medidor-de-consumo-de-gas.html, (accedido el 21 de Junio de 2017).

- [14] Arduino Programming Notebook, *Brian W.Evans*, OpenLibra, Ed. Española. 2011.
- [15] PROMETEC.COM. Disponible en https://www.prometec.net/wemos-d1-esp8266-wifi/, (accedida el 21 de Junio de 2017).
- [16] TOP 6 ESP8266 MODULES FOR IOT PROJECTS, Todd Henderson. 2016.
- [17] SCT-013-000.PDF, Hoja de características, Sensor de Corriente. Disponible en http://www.datasheetspdf.com/datasheet/SCT-013-000.html, (accedido el 21 de Junio de 2017).
- [18] TCRT5000L.PDF, Hoja de características, Sensor Infrarrojo Reflexivo. Disponible en, http://www.vishay.com/docs/83760/tcrt5000.pdf, (accedido el 21 de Junio de 2017).
- [19] ESP8266.PDF, Hoja de características, ESP8266. Disponible en http://download.arduino.org/products/UNOWIFI/0A-ESP8266-Datasheet-EN-v4.3.pdf, (accedido el 21 de Junio de 2017).
- [20] FMCALEFACCION.COM Especificaciones técnicas, Modelo H-2000. Disponible en http://www.fmcalefaccion.com/es/estufas/44-estufa-halogena-h1-2000-8427561020001.html, (accedido el 21 de Junio de 2017).
- [21] ARDUINO. CURSO PRÁCTICO DE FORMACIÓN, Óscar Torrente Artero. Primera edición. 2013.
- [22] ESP8266 LOW POWER SOLUTIONS, Espressif IOT Team. Version 1.1. 2016.
- [23] OPEN ENERGY MONITOR, openenergymonitor.org. Disponible en https://openenergymonitor.org/, (accedido el 21 de Junio de 2017).
- [24] Internet de las cosas y el futuro de la economía conectada, $Klaus\ Schwab$
- [25] ADEL S. SEDRA, Microelectronic Circuits
- [26] Guide to AngularJS Documentation, Open-source tools for energy monitoring and analysis
- [27] COLEGIO OFICIAL DE INGENIEROS DE TELECOMUNICACIONES. *Disponible en* https://www.coit.es/, (accedido el 21 de Junio de 2017).

BIBLIOGRAFÍA 79

[28] The Industrial Internet of Things, Alasdair Gilchrist

- [29] ESP8266 LOW POWER SOLUTIONS, Espressif IOT Team. Version 1.3. Aug. 2016. Disponible en https://espressif.com/sites/default/files/documentation/9b-esp8266-low_power_solutions_en.pdf (accedida el 21 de junio de 2017).
- [30] WEB DE LUIS LLAMAS INGENIERÍA, INFORMÁTICA Y DISEÑO (ZARAGOZA), *Luis Llamas*. Disponible en https://www.luisllamas.es/ (accedida el 19 de junio de 2017).