

UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

Implementación de un sistema domótico basado en
Raspberry Pi

GRADO EN INGENIERÍA DE
SISTEMAS ELECTRÓNICOS

JUAN MANUEL VÁZQUEZ JIMÉNEZ
MÁLAGA, 2021

Implementación de un sistema domótico basado en Raspberry Pi

Autor: Juan Manuel Vázquez Jiménez.

Tutor: José Manuel Cano García.

Cotutora: Eva González Parada.

Departamento: Departamento de Tecnología Electrónica.

Titulación: Grado en Ingeniería de Sistemas Electrónicos.

Palabras clave: Domótica, Raspberry, MQTT, ESP8266, HomeAssistant, web scraping.

Resumen

El proyecto escogido como trabajo de fin de grado, trata del desarrollo y la implementación de un sistema domótico con el fin de incrementar el ahorro energético y comodidad del cliente.

En este proyecto se pretende desarrollar una red de dispositivos actuadores (ESP8266) comunicados a una centralita (Raspberry) mediante MQTT. Para el control del sistema, se ha optado por la integración de Home Assistant en Raspberry ya que ofrece numerosas funcionalidades e interfaces.

Para la implementación del sistema se propone el reacadicionamiento de elementos existentes, conformando varios módulos del sistema y permitiendo así el abaratamiento del coste para el cliente sin perjuicio de la customización. Este proyecto ha sido pensado para que cualquier tipo de usuario (con o sin conocimientos previos) pueda realizar una configuración del sistema a través de módulos configurables que permiten una mayor adaptación a las necesidades de este.

En este trabajo, se destaca el uso de web scraping como técnica principal para la obtención de información del entorno a través de páginas web. Esta información es posteriormente procesada y puede ser utilizada para determinar el comportamiento del sistema.

Implementation of a home automation system based on Raspberry Pi

Author: Juan Manuel Vázquez Jiménez.

Supervisor: José Manuel Cano García.

Cosupervisor: Eva González Parada.

Department: Departamento de Tecnología Electrónica.

Degree: Grado en Ingeniería de Sistemas Electrónicos.

Keywords: Home automation, Raspberry, MQTT, ESP8266, HomeAssistant.

Abstract

The topic chosen for this paper as the final degree project is about the development and implementation of a home automation system in order to increase energy savings and customer comfort.

This project aims to develop a network of actuator devices (ESP8266) communicated to a central (Raspberry) through MQTT. For the system control, it has been selected the integration of Home Assistant in Raspberry since it offers numerous functionalities and interfaces.

The main solution that is proposed is a reconditioning of elements consisting of various system modules. Therefore, it reduces client costs without limiting customization. This project has been designed for any type of user (with or without previous knowledge) helping to configure the system through customizable modules that allows a greater adaptation to users needs.

In this work, the use of web scraping is highlighted as the main technique to get information from web pages. This information is subsequently processed and can be used to determine the system behavior.

Agradecimientos

En especial, a mis padres por brindarme la oportunidad de realizar mis estudios y por el apoyo que me han dado a lo largo de estos años. A mi hermana, por sus consejos y apoyo incondicional. A mis tías, por mostrar ser siempre otro pilar donde apoyarme.

Agradecer a Eva González Parada por la dedicación y a José Manuel Cano García por la atención que han prestado durante la elaboración de esta memoria.

Contenido

Capítulo 1. Introducción.....	1
1.1. Contexto tecnológico.....	1
1.2. Objetivos del Trabajo Fin de Grado.....	5
1.3. Estructura de la memoria.....	7
Capítulo 2. Especificaciones del sistema.....	8
2.1. Requisitos.....	8
Capítulo 3. Desarrollo del sistema.....	11
3.1. Tecnologías implicadas.....	12
3.2. Desarrollo hardware.....	13
3.2.1. Módulo RGB.....	14
3.2.2. Módulo Relé.....	15
3.2.3. Módulo meteo-ropa.....	18
3.3. Desarrollo software.....	21
3.3.1. Raspberry.....	21
3.3.2. NodeMCU.....	25
3.3.3. Python.....	28
3.4. Configuración fuera de LAN.....	30
3.4.1. Configuración del router.....	30
3.4.2 Configuración DNS.....	31
Capítulo 4. Verificación y pruebas.....	33
4.1. Sistema de pruebas.....	33
4.2. Pruebas realizadas.....	33
Capítulo 5. Manual de instalación y uso.....	35
5.1. Requisitos de instalación.....	35

5.2. Manual de instalación.....	36
5.2.1. Raspberry.....	36
5.2.2. Conexión VNC Raspberry.....	38
5.2.3. NodeMCU.....	39
5.2.4. Home Assistant.....	41
5.2.5. Nuevas integraciones.....	43
Capítulo 6. Conclusiones y trabajo futuro.....	47
Apéndice A. Presupuesto de elaboración.....	49
Apéndice B. Encuesta de uso del sistema.....	51
Referencias.....	52
Bibliografía.....	53

Capítulo 1. Introducción

1.1. Contexto tecnológico

Según la asociación española de domótica e inmótica (CEDOM) [1] “La domótica es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de comunicación entre el usuario y el sistema”.

En abril de 1966 fue implementado el primer producto domótico (a pesar de no haber sido comercializado nunca) el aparato de aproximadamente 360kg, conocido como “ECHO IV” [2], podía desempeñar tareas como la monitorización de la temperatura de una vivienda, hacer una lista de la compra, así como controlar el encendido y/o apagado de electrodomésticos. Sin saberlo, Jim Sutherland (creador del invento), abrió paso a una evolución en los hogares de todo el mundo.

Más tarde, en 1969, fue forjado el corazón de la domótica moderna con la llegada del proyecto ARPANET (Advanced research proyect agency network) propulsado por el gobierno de EEUU, dando lugar a una red de comunicación que precede a la red actual de internet y que posteriormente desencadenó en el concepto de IoT (Internet of things).

La llegada de los microcontroladores en 1971 y su posterior comercialización en 1974, conformaría otra de las partes fundamentales en la domótica moderna. Dotando así de sistemas con una mayor capacidad de integración y mucho más compactos al sustituir numerosos circuitos analógicos por uno (o varios) microcontroladores.

En mayo de 1999, emerge un nuevo estándar para la automatización y gestión de viviendas y edificios con la fundación KNX Association cvba. Este nuevo

estándar, es considerado uno de los sistemas más importantes empleados en domótica a nivel internacional. Los modos de funcionamiento estándar son:

- S.mode(System mode): Instalación y configuración realizada por profesionales.
- E.mode(Easy mode): Es compuesto por un sistema de control central con ajustes predefinidos por el fabricante y una parte personalizable por el usuario mediante el sistema de control central. Utilizado para instalaciones de pequeño tamaño. En esta filosofía se basa la implementación del sistema domótico de este TFG.
- A.mode(Automatic mode): Dispositivo con instalación automática.

Los sistemas KNX se caracterizan por ser un sistema descentralizado, integración mediante dispositivos KNX inteligentes, acceso mediante CSMA/CA y no es un sistema propietario, aunque sus fabricantes (pertenecientes a EIBA) son los únicos autorizados a desarrollar los dispositivos.

En la actualidad, la domótica constituye un volumen de facturación estimado según la CEDOM [3] de 91.000.000€ a lo largo del año 2019. En la figura 1 se puede observar la evolución del mercado en el sector de la domótica en España, la cual refleja un importante incremento en la domotización de los hogares españoles.

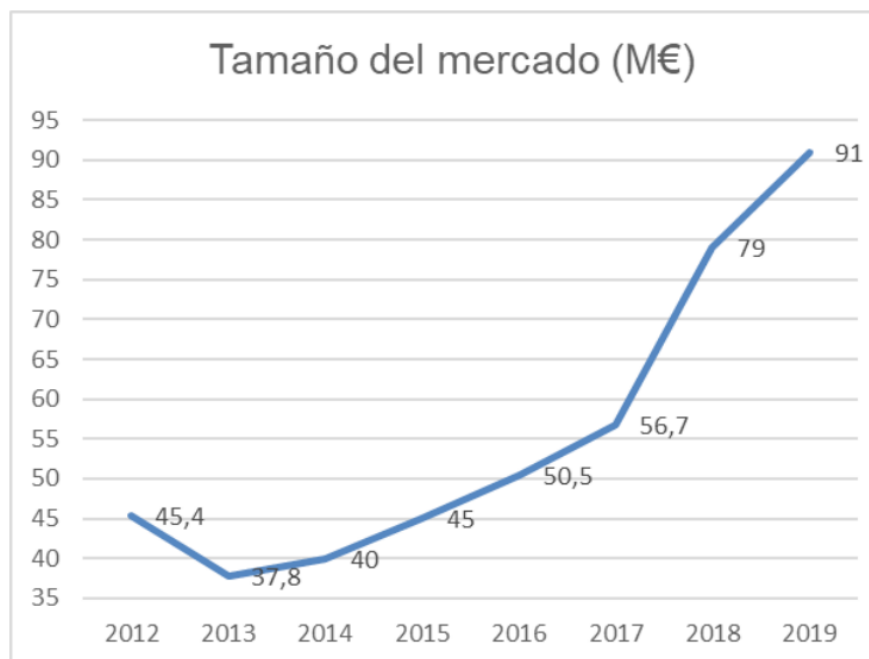


Figura 1. Facturación de los fabricantes de sistemas de control y automatización. Años 2012-2019
(datos en M€) (Fuente: CEDOM).

El abaratamiento de los dispositivos, la fácil integración de productos en los sistemas domóticos centralizados como “Amazon Echo” (60€ aprox.) o “Google Home” (90€ aprox.) la gestión eficiente de los recursos, la llegada de los smartphones y el confort que proporciona al usuario de una vivienda inteligente; son los principales motivos por los que la domótica está cada vez más presente en los hogares de los países más pudientes.

En los sistemas mencionados anteriormente (Amazon Echo y Google Home) destaca el gran número de productos totalmente preparados para que el comprador lo integre en su hogar con unos sencillos pasos; se puede observar que, bombillas, interruptores, sistemas de seguridad y altavoces son los productos inteligentes más asequibles y comunes en cuanto a la domótica no profesional, aunque existe una amplia gama de productos que van más allá de las funciones que ofrecen los mencionados anteriormente. Este abaratamiento en los dispositivos. ha llegado a abarcar un mayor número de usuario que optan por estas opciones frente a las que proporciona una empresa profesional, que aunque son por lo general, implementaciones más sólidas, también provocan un encarecimiento del precio.

Por otro lado, la integración de ciertos sistemas, pueden proporcionar una gestión eficiente en cuanto a los recursos energéticos, este aprovechamiento se puede observar principalmente en las siguientes áreas:

- Climatización: Con elementos que van desde la opción más simple (termostatos inteligentes) hasta la opción más eficiente, como la de un control de las persianas, temperatura del hogar y luminosidad que, en conjunto, permite una climatización óptima.
- Iluminación: La integración de luces inteligentes que adaptan la intensidad lumínica que estas ofrecen y sensores que detectan la presencia de personas, son las opciones más usuales.
- Electrodomésticos: Mediante un control del encendido y apagado de los dispositivos durante las horas donde el precio sea inferior y superior respectivamente. Además, existe la posibilidad de programar la desconexión automática de dispositivos no prioritarios cuando el consumo del hogar es elevado y se prevé alcanzar el umbral contratado.

Según el artículo del diario elPeriódico [4], el uso de smartphones se ha incrementado hasta tal punto, que el 90% de la población mundial adulta, dispone de un teléfono móvil el cual es la herramienta del 91% de internautas para acceder a la red. La gran acogida que estos dispositivos han experimentado, ha propiciado que su uso en la domótica sea casi un requisito indispensable para el control de los sistemas mediante una aplicación.

1.2. Objetivos del Trabajo Fin de Grado

En este Trabajo de Fin de Grado se pretende poner en práctica los conocimientos obtenidos a lo largo del grado así como ampliarlos mediante el aprendizaje de herramientas que no se han impartido en el grado. Dividiendo así los objetivos en tres partes:

- Software: Implementando el sistema en lenguajes de programación tanto con entornos gráficos (NodeRed) como una programación basada en Python y C. Además, se introduce al web-scraping, técnica muy útil para la obtención de información a través de páginas webs.
- Hardware: Desarrollando y simulando circuitos necesarios (OrCAD) junto al diseño PCB de cada uno de los módulos que son objeto de estudio.
- Comunicación: El desarrollo de un sistema comunicado a través de “topics” en el protocolo de red MQTT. Guiado por la filosofía de proporcionar la comodidad al usuario, el control será dirigido con cualquier dispositivo con acceso a Internet (smartphone, tablet, ordenador...).

En este proyecto se procede a implementar un sistema domótico central basado en un pequeño sistema embebido de bajo coste (Raspberry); este hardware ha sido escogido por su versatilidad así como de disponer software de código abierto, lo que implica una facilidad añadida por su gran comunidad. El modelo utilizado para el desarrollo del sistema es: Raspberry Pi 3 Model B. También, se procede a la implementación de módulos basados en el microcontrolador ESP8266, concretamente en la versión NodeMCU 1.0 v3; este microcontrolador ofrece una fácil programación a alto nivel, un bajo coste y gran capacidad de implementación. Entre los módulos que son objeto de estudio en este TFG se encuentran:

- Módulo Relé: El microcontrolador controla el funcionamiento de un relé que activa o desactiva cualquier dispositivo eléctrico que se conecte a este módulo.
- Módulo RGB: El usuario puede controlar una tira LED, permitiéndole modificar la mezcla de colores y/o su intensidad.
- Módulo meteo-ropa: La Raspberry ordena al microcontrolador NodeMCU, cuando debe desplegar un toldo para mantener seca la colada en caso de lluvia.

Además, motivado por filosofía de la gestión eficiente de la energía, se ha implementado un programa para la monitorización del precio actual de la luz, así como el precio máximo y mínimo a lo largo del día, dotando en todo momento una información relevante para controlar el consumo energético. Otra medida para la gestión de la energía, ha sido la posibilidad de programar el encendido y apagado del módulo relé cuando el precio actual de la luz eléctrica sea inferior a un valor establecido por el usuario.

Home Assistant ha sido la herramienta de apoyo para el control de los dispositivos de los que cuenta el instrumento domótico. Esta herramienta proporciona un gran número de interfaces amigables e intuitivas para el usuario. Home Assistant es un software con licencia open-source lanzado en 2013 basado en Python y de la cual dispone una gran comunidad por lo que el número de ejemplos para integraciones y personalizaciones del sistema, es bastante grande, de manera que con una búsqueda en la página oficial de la plataforma pueden ser observados numerosos ejemplos que sirven como patrón.

Como ya se ha comentado, la principal finalidad es la creación de un sistema domótico de bajo coste, alta capacidad de integración y escalabilidad. En todo momento se ha perseguido una filosofía promovida por el ahorro energético y por una interfaz amigable y sencilla para que los usuarios puedan incrementar las funcionalidades propuestas.

1.3. Estructura de la memoria

En el capítulo 1 se ha realizado un estudio del arte donde se ha recapitulado la evolución de la domótica y las tecnologías que la conforman en la actualidad. Posteriormente se ha procedido a una introducción al sistema desarrollado y las herramientas utilizadas para la elaboración del proyecto, así como los objetivos que se persigue en el TFG.

A continuación, se procede al capítulo 2 donde se hace un análisis más detallado de las especificaciones del sistema para cumplir los objetivos detallados en el capítulo anterior. Con ello, se pretende dar una visión general del proyecto.

En el capítulo 3, se puede observar de manera más detalla el desarrollo del sistema tanto en su parte software como hardware y comunicaciones. Se ha elaborado un diagrama para una mejor comprensión del funcionamiento del sistema así como varias figuras que justifican la utilización de algunos componentes. Además, en el último apartado del capítulo, se realiza una explicación de las configuraciones necesarias para poder controlar el sistema desde fuera de la red de área local.

Seguidamente, en el capítulo 4, se procede a una verificación de los requisitos detallados en el punto 2.1 donde se describen tanto las pruebas realizadas para su verificación, como el resultado obtenido de dicha verificación.

El manual de instalación y uso se detalla en el capítulo 5, donde se especifica todos los detalles y pasos a seguir para configurar una Raspberry totalmente nueva a como se presenta en este trabajo de fin de grado. Además se realiza una explicación de los parámetros que se deben cambiar en los programas correspondientes a los microcontroladores así como de las herramientas y hardware necesario.

En último lugar se encuentran las conclusiones que se han obtenido en el desarrollo del proyecto, donde se realiza un pequeño análisis de posibles desarrollos y mejoras del sistema para futuros desarrolladores.

Capítulo 2. Especificaciones del sistema

2.1. Requisitos

En este capítulo se procede a presentar los requisitos que se han considerado oportunos para el desarrollo del proyecto, donde se describen las funcionalidades que se desean desarrollar. En la figura 2, se puede observar un diagrama general del prototipo desarrollado; este diagrama se detalla en capítulo 3.

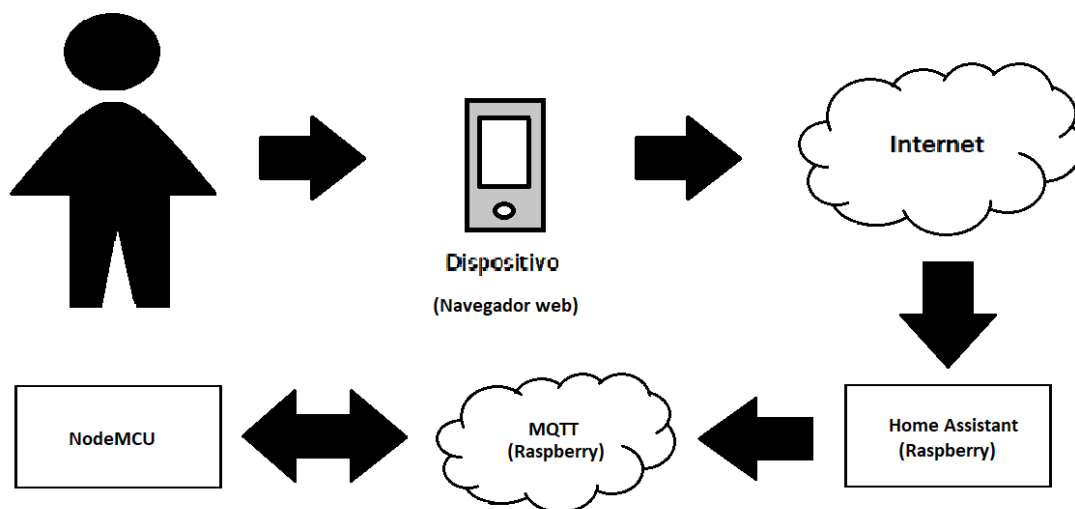


Figura 2. Diagrama general del prototipado.

Id.	Nombre	Descripción	Tipo
RF1	Fuera de LAN	El sistema puede ser controlado desde cualquier dispositivo sin necesidad de que este se encuentre conectado a la misma red a la que se encuentra la central conectada.	Comunicación
RF2	MQTT	Los diferentes módulos son comunicados con la central a través de MQTT.	Comunicación
RF3	Arranque automático	El sistema ejecuta los servicios necesarios para poder lanzar tanto la aplicación de control como su gestión.	Software
RF4	Autenticación	Habilitar un panel de autenticación basado en un usuario y contraseña.	Software
RF5	Panel de Usuario	Desarrollo de un panel de control que permita al usuario utilizar el sistema.	Software
RF6	Uso de arquitectura software que permita la utilización del bajo consumo	Aquellos módulos que son automatizados y que no requiere de una manipulación del usuario, disponen de un modo de bajo consumo tras realizar las tareas para las que haya sido concebido.	Hardware
RF7	PCB	Diseño PCB de los circuitos necesarios para la modularidad del sistema.	Hardware
RF8	Web Scraping	Integración en el diseño de un sistema de captación de información de datos procedentes de páginas WEB	Software
RF9	Circuito de protección	Diseño e implementación de circuitos electrónicos de protección y acondicionamiento en aquellos donde se precise para evitar rupturas por excesos de tensión.	Hardware
RF10	Diseño RGB HW	Realización de un circuito capaz de controlar una tira LED RGB mediante el microcontrolador ESP8266.	Hardware

RF11	Diseño Relé HW	Realización de un circuito capaz de controlar un relay mediante el microcontrolador ESP8266.	Hardware
RF12	Diseño motor paso a paso HW	Realización de un circuito capaz de controlar un motor paso a paso mediante el microcontrolador ESP8266.	Hardware
RF13	Control RGB SW	Implementación de un programa basado en C, capaz de realizar las operaciones necesarias para el control de una tira LED a través del microcontrolador ESP8266.	Software
RF14	Control Relé SW	Implementación de un programa basado en C, capaz de realizar las operaciones necesarias para el control de un relay a través del microcontrolador ESP8266.	Software
RF15	Control motor paso a paso SW	Implementación de un programa basado en C, capaz de realizar las operaciones necesarias para el control de un motor paso a paso a través del microcontrolador ESP8266.	Software

Tabla 2.1. Tabla de descripción de requisitos.

RF5: Como consideración de diseño, se debe conseguir que el panel de control sea intuitivo en la medida de lo posible.

Capítulo 3. Desarrollo del sistema

En el siguiente capítulo se procede a describir tanto la programación software, como el hardware que te ha elegido para la implementación de este proyecto.

En la figura 3, se presenta un diagrama general del sistema desarrollado, donde se puede visualizar la relación entre los distintos elementos.

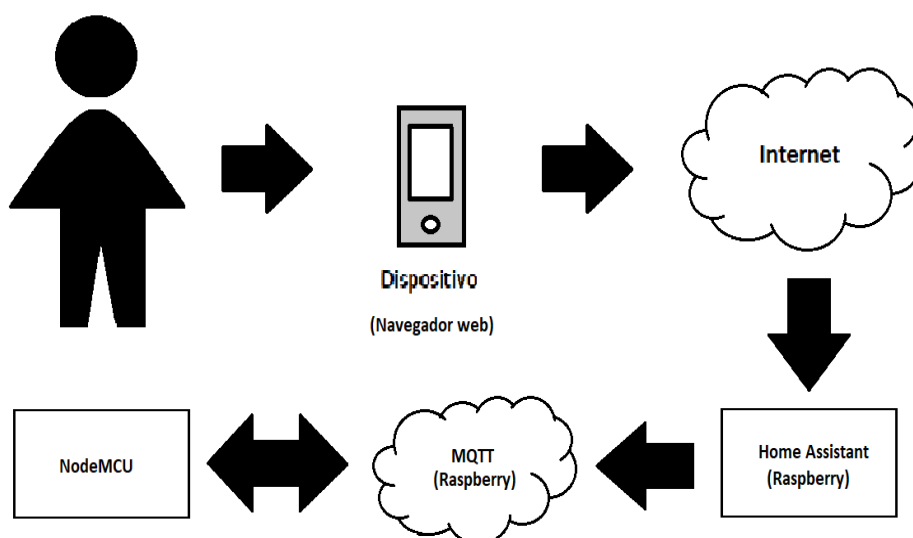


Figura 3. Diagrama general del prototipo.

Como se puede ver en el diagrama de la figura 3, el usuario accede al panel de control de Home Assistant a través del navegador web de cualquier dispositivo (para ello se debe agregar la IP de la Raspberry seguida de “:8123”).

El sistema consta de una central (Raspberry) la cual aloja el servidor de Home Assistant que sirve como herramienta de comunicación entre el usuario y el sistema. Los módulos, compuestos por el microcontrolador NodeMCU, son comunicados a la centralita mediante la suscripción de topics al broker MQTT (residente en Raspberry). Los mensajes transmitidos por el flujo de comunicación, determina el comportamiento de los actuadores de cada uno de los módulos desarrollados.

3.1. Tecnologías implicadas

A continuación se realiza un breve desarrollo de las principales tecnologías que se han utilizado para el desarrollo de este proyecto.

Node-RED es una herramienta de programación “open-source” de alto nivel basada en Node.js que facilita el conexionado de nodos. Sacada a la luz en 2013 por Nick O’Leary y Dave Conway-Jones, fue desarrollada para la visualización y manipulación de la comunicación MQTT en el prototipado de código, sin embargo, poco a poco se ha incrementado su funcionalidad mediante el desarrollo de paletas de nodos que actúan como librerías (proporcionando una mayor sencillez en la programación). Home Assistant es una de las plataformas “open-source” más conocidas en el mundo de la domótica, con abundantes módulos ya desarrollados que posibilitan el control dispositivos de distintas tecnologías y dispositivos. La herramienta otorga el control de los dispositivos desde un interfaz web, así como la programación de scripts que modelan la automatización de comportamientos.

Raspberry es un pequeño sistema empujado de bajo coste. Este hardware ha sido escogido por su versatilidad así como de disponer software de código abierto, lo que implica una facilidad añadida por su gran comunidad. El modelo utilizado para el desarrollo del sistema es: Raspberry Pi 3 Model B; sacado a la luz en febrero del año 2016, renueva procesador, una vez más un Quad-Core, pero pasa de 900MHz a 1.20GHz con una arquitectura de 64bit y una memoria RAM de 1GB.

NodeMCU 1.0 v3 es una plataforma basada en el SoC ESP8266 sacada en 2014 que cuenta con una frecuencia de procesamiento de 80MHz, WiFi 802.11.

3.2. Desarrollo hardware

En este apartado se procede a explicar los circuitos hardware necesarios, así como simulaciones pertinentes.

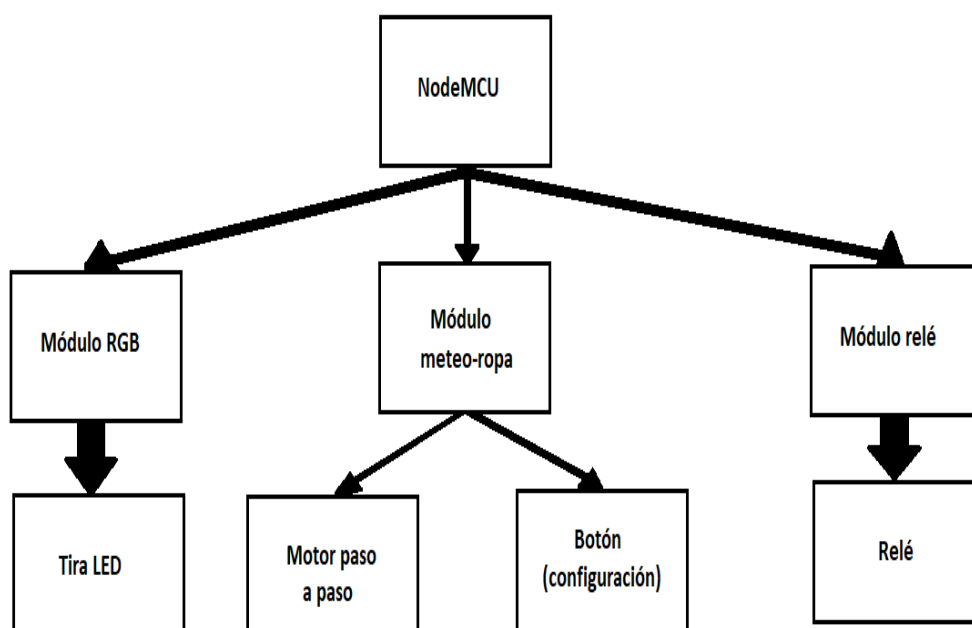


Figura 4. Esquema módulos NodeMCU.

En la figura 4, se puede observar los módulos desarrollados mediante el microcontrolador NodeMCU y los actuadores que este controla en cada uno de los módulos descritos en el apartado 1.2.

En el caso del módulo “meteo-ropa”, dispone de dos elementos principales, el primero es el motor que controla el toldo y el segundo es un botón que configura las vueltas que el motor debe dar para dicha regulación, una vez que se obtienen los parámetros de la configuración, son mandados a través de MQTT para almacenarlos en un fichero alojado en la Raspberry.

A continuación, en los siguientes apartados, se procede a realizar un estudio del conexionado de los elementos básicos considerados para el sistema implementado, al nodo microcontrolador, efectuando un análisis de la problemática y las soluciones adoptadas en el desarrollo del hardware del proyecto.

3.2.1. Módulo RGB

Este módulo consta del control de una tira LED mediante los pines GPIO del microcontrolador. Para cada color que conforma la tira, le ha sido asignado un pin GPIO.

Por otro lado, cada pin del microcontrolador es capaz de entregar una corriente de 12mA (insuficiente para aprovechar la intensidad lumínica que ofrece la tira LED la cual necesita unos 20mA en corriente directa) por lo que se ha optado por la integración de un driver de corriente, concretamente el ULN2003 el cual conforma una configuración Darlington con los transistores internos que lo conforman. Se ha escogido dicha solución para incrementar la compactación del módulo. Una de las principales características del driver escogido, es que sus salidas son inversoras, por lo que cuando se transmita un nivel alto en su entrada, un nivel bajo es registrado en su salida. Esta cualidad es beneficiosa para la programación tanto del módulo RGB como del módulo meteo-ropa, ya que estos disponen de su linea común conectada a Vcc. A continuación, en la figura 5, se puede observar el conexionado del módulo RGB, donde el LED representa la tira LED y el conector jack, la alimentación que proviene del transformador que el fabricante de la tira de LED incluye.

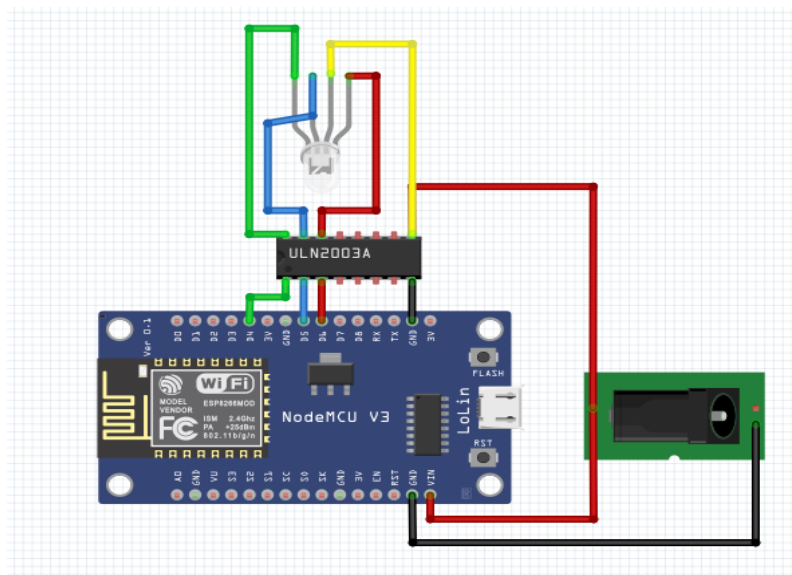


Figura 5. Esquema del conexionado RGB.

3.2.2. Módulo Relé

Este módulo dota al usuario del control de un relé que conmuta el encendido/apagado del dispositivo conectado a este módulo.

El relé está compuesto de una bobina que se opone a los cambios de flanco, es decir, cuando se produce un cambio de estado (HIGH→LOW) la bobina entrega un pico de tensión que puede ser peligroso debido a la posibilidad de ruptura del microcontrolador, por lo que se ha diseñado un circuito de protección que consta de un transistor 2n2222(NPN) y un diodo en antiparalelo.

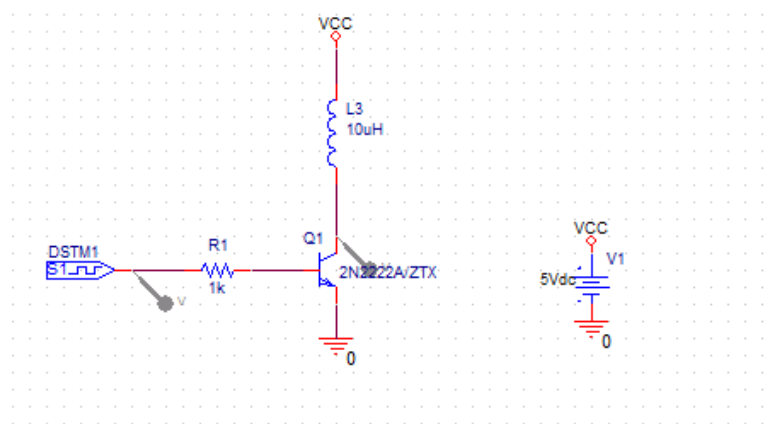


Figura 6. Circuito de simulación conmutación del relé sin protección.

A continuación se procede a simular el circuito de la figura 6 donde se realiza un cambio de flanco de 5V (alto) a 0V (bajo) que se produce a los 50ms; simulando así el cambio de flancos en el microcontrolador.

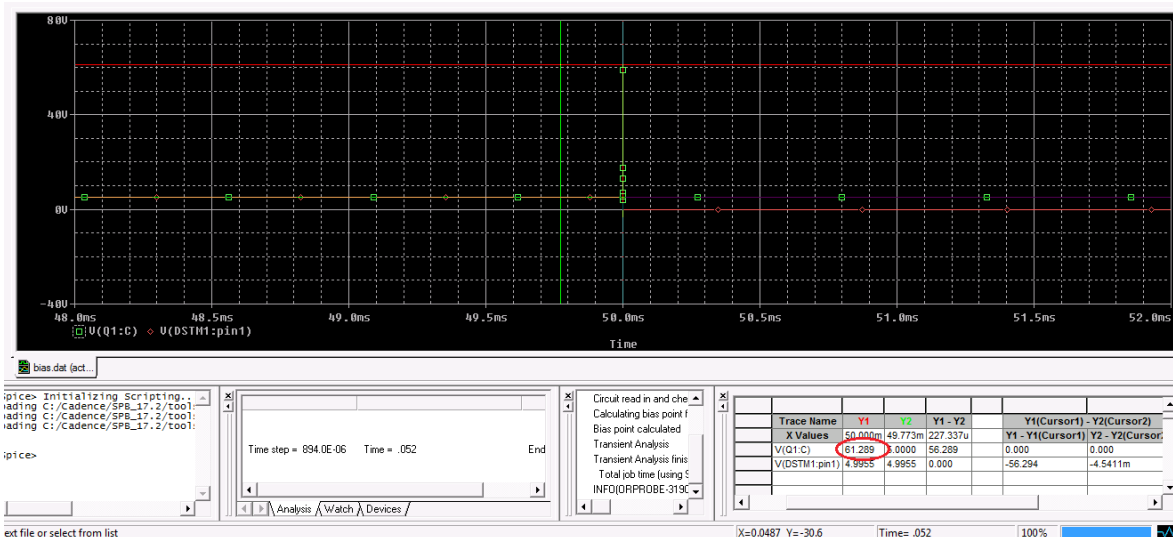


Figura 7. Resultados de la simulación del circuito relé sin elementos de protección.

Como se puede observar en los resultados de la simulación (figura 7) tras una transición, la bobina que conforma el relé, libera un pico de tensión oponiéndose al cambio del estado alto a bajo. Este pico llega a los 61.3V, tensión demasiado alta para que el microcontrolador sea capaz de soportar, pudiendo llegar a provocar una ruptura de sus componentes.

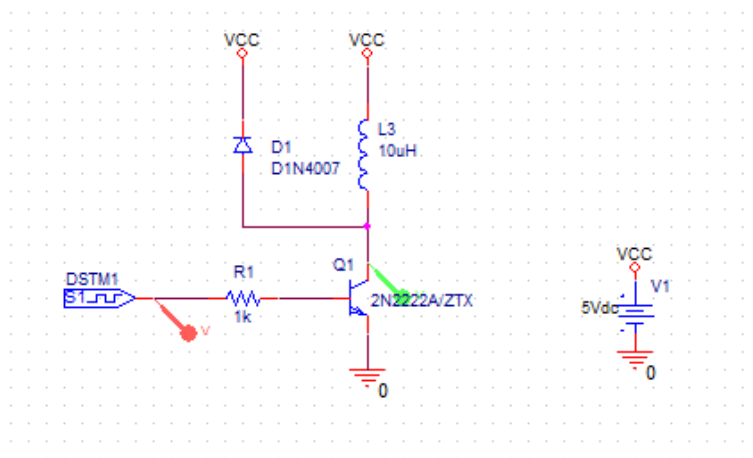


Figura 8. Circuito de simulación conmutación del relé con protección.

Es por esto, que se ha optado por la inserción de un diodo en configuración antiparalelo. Este diodo permite que cuando se produzca los picos de tensión en los cambios de flancos, se descargue este exceso de tensión a través de la fuente y no del microcontrolador.

A continuación se procede a simular el circuito de la figura 8 bajo las mismas condiciones de la simulación del circuito correspondiente a la figura 6.

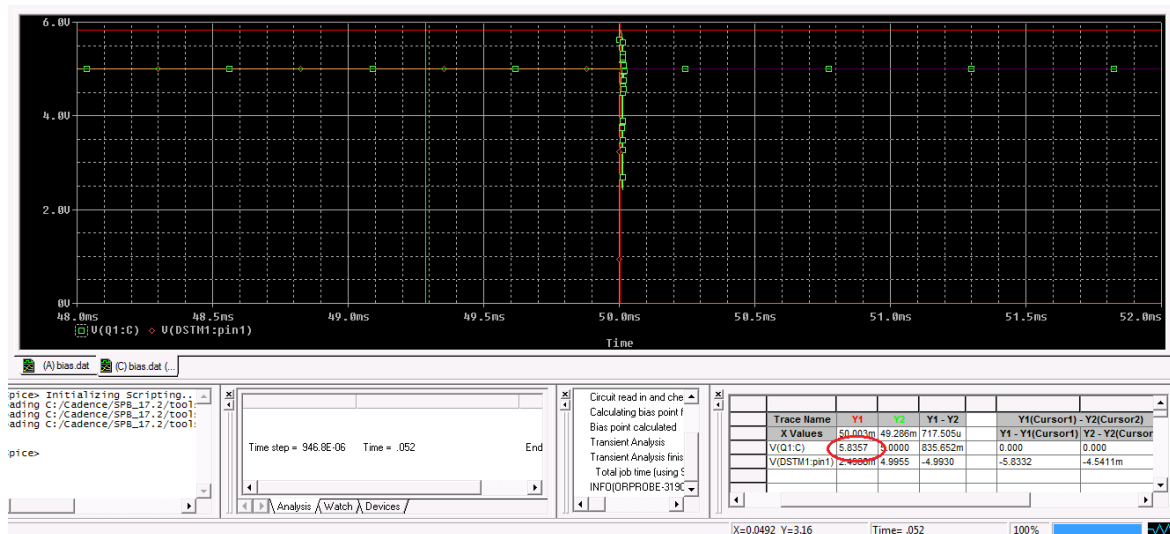


Figura 9. Resultados de la simulación del circuito relé con elementos de protección.

Como se puede observar en la figura 9, gracias a la inserción del diodo, el pico de tensión no supera apenas el voltio por lo que se ha reducido 55.5V extras en los pines del microcontrolador. En la figura 10 se puede observar el conexionado de los elementos que conforman el módulo relé.

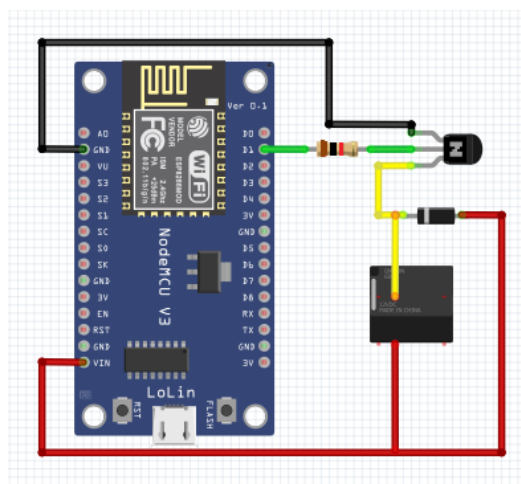


Figura 10. Conexionado del módulo relé.

3.2.3. Módulo meteo-ropa

Para el control de este, se basa en la utilización de un motor paso a paso el cual gira hacia un lado u otro según el tiempo meteorológico del lugar donde se encuentra el usuario. Este módulo dispone de un botón para configurar las vueltas que debe de dar el motor; una vez se ha obtenido esta medida, se manda el resultado a través de MQTT y lo almacena en un fichero de la centralita (Raspberry). Cada 15 minutos, la Raspberry, revisa el tiempo meteorológico a través del web-scraping; cuando el tiempo meteorológico es lluvioso, la central, manda un mensaje a través de MQTT al microcontrolador para que este, despliegue el toldo; de lo contrario, el toldo procede a recogerse.

El microcontrolador ha sido programado para que entre en modo bajo consumo y este no consuma energía innecesariamente mientras no realiza la acción, por lo que cada 14min, despierta a través de una interrupción mediante el pin D0, revisa si le ha llegado un mensaje que cambie el estado del motor y una vez realizadas las operaciones, vuelve a entrar en bajo consumo.

El ESP8266 dispone de 3 modos de bajo consumo:

- Modem-Sleep: Donde la CPU sigue funcionando así como PWM o I2S. El ahorro de energía se obtiene deshabilitando la transmisión de datos a través de la red WiFi. El consumo total se encuentra alrededor de los 15mA.
- During Light-Sleep: La CPU puede ser suspendida. Al igual que el modo anterior, este no permite la transmisión de datos mediante la WiFi. El consumo se ve disminuido a 0.9mA.
- Deep-Sleep: Se desactiva cualquier tipo de conexión WiFi. Su consumo se encuentra al rededor de los 10uA.

Ante la previsión de una actividad periódica, se ha optado por el modo más restrictivo (Deep-Sleep) ya que esto reduce considerablemente el consumo. A pesar de que el consumo del microcontrolador se haya reducido, el consumo total no equivale a 10uA en el modo "Deep-Sleep", debido a que la placa posee otros componentes como reguladores de tensión entre otros que consumen una cierta corriente.

Para visualizar el consumo real del microcontrolador, se ha utilizado una fuente de alimentación “Keysight E36313A” con la cual se puede observar el consumo tanto gráficamente como cuantitativamente. Para ello, se ha desarrollado un programa simple en el que la placa NodeMCU entra en un estado cíclico de activa-bajo consumo para así poder observar estas variaciones de corriente.

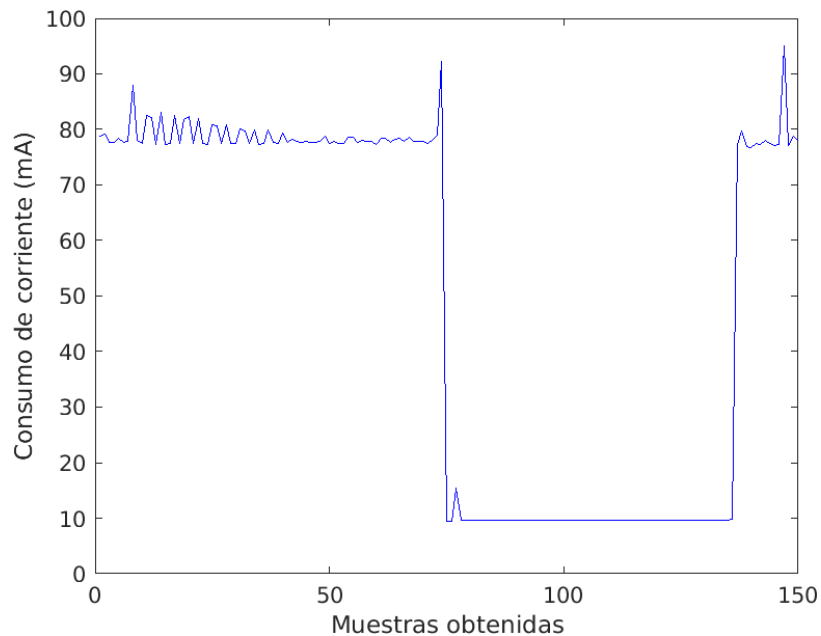


Figura 11. Resultados obtenidos con Keysight E36313A.

El resultado obtenido es una nube de puntos que puede ser exportada y representada mediante MATLAB. Como se puede observar en la figura 11, el consumo oscila en torno a los 80mA cuando la placa se encuentra activa, tras el inicio en bajo consumo, este decae hasta los 10mA aproximadamente; una vez que el microcontrolador sale del bajo consumo, se observa un pico de corriente correspondiente al reseteo del mismo (conexión D0-RST).

En la función que se configura el microcontrolador en bajo consumo, `ESP.deepSleep(TIEMPO_DeepSleep, WAKE_NO_RFCAL)`, se puede observar un segundo parámetro (modo de reinicio) el cual dispone de 4 configuraciones posibles:

- WAKE_RF_DEFAULT: Se procede a una calibración de la señal si esta detecta un error.
- WAKE_RFCAL: Siempre se realiza un calibrado de la señal.
- WAKE_NO_RFCAL: No se realiza ningún calibrado de la señal.
- WAKE_RF_DISABLED: Tras el reinicio, se deshabilita la señal WiFi.

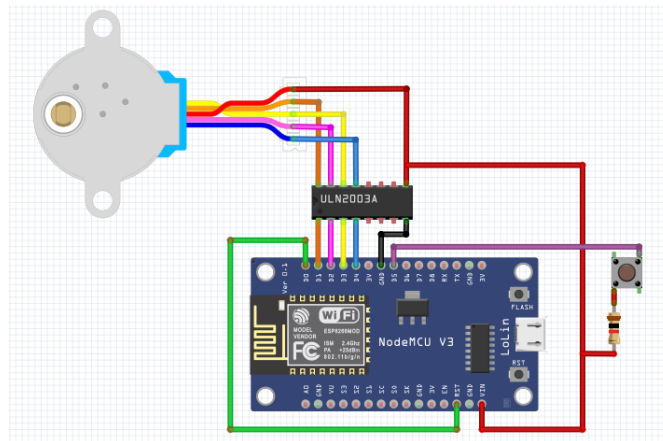


Figura 12. Conexión del módulo meteo-ropa.

En la figura 12 se puede observar el conexionado de los elementos hardware que conforman el módulo “meteo-ropa”.

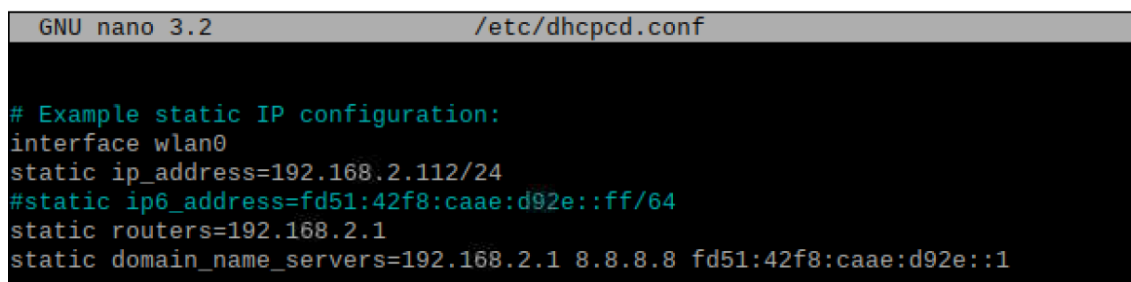
3.3. Desarrollo software

A continuación se procede a realizar un estudio de los programas realizados en el proyecto y las configuraciones software necesarias para la elaboración de este.

3.3.1. Raspberry

En este apartado se realiza las explicaciones pertinentes de los programas alojados en la Raspberry tanto de la herramienta Node-RED, como de Home Assistant o de la configuración propia de la centralita.

Para evadir futuros problemas con la IP del broker MQTT, se recomienda configurar la IP de este como estática; para ello se debe modificar el archivo “dhcpcd.conf” mediante el comando `sudo nano /etc/dhcpcd.conf` tal y como se muestra en la figura 13, continuando con el guardado del archivo (Ctrl+o) y el posterior reinicio de la Raspberry.



```
GNU nano 3.2 /etc/dhcpcd.conf

# Example static IP configuration:
interface wlan0
static ip_address=192.168.2.112/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.2.1
static domain_name_servers=192.168.2.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

Figura 13. Configuración del archivo “dhcpcd.conf”.

El parámetro que aparece en la figura 13 “wlan0” se corresponde con la tarjeta de red, este nombre es variable y se puede visualizar mediante la ejecución del comando `ip a` en una terminal.

Como ya se ha comentado anteriormente, se ha intentado que la programación del proyecto sea lo más simple y amigable para el usuario por lo que se ha perseguido una mínima configuración del sistema por parte del usuario y que esta pueda ser lo más amigable posible. Es por esto que se ha optado por la utilización de dos herramientas fundamentalmente detalladas en los dos siguientes apartados.

3.3.1.1. Node-Red

La programación a través de bloques que esta aplicación ofrece, ha resultado una de las soluciones más factibles en cuanto a la facilidad que ofrece para la modificación de parámetros para el usuario.

Para acceder a la herramienta es necesario introducir en un navegador, la IP de la Raspberry seguida de “:1880”.

-Módulo meteo-ropa: La programación de este módulo se basa en el script “tiempo.py” (basado en Python) el cual realiza el web-scraping. El web-scraping se basa en la obtención de datos de páginas webs a través de la inspección de su código fuente; en este caso, la información se obtiene a través de la página web www.eltiempo.es de la cual se extrae el informe de la precipitación actual en la ubicación. Esta comprobación se realiza de manera periódica cada 15min.

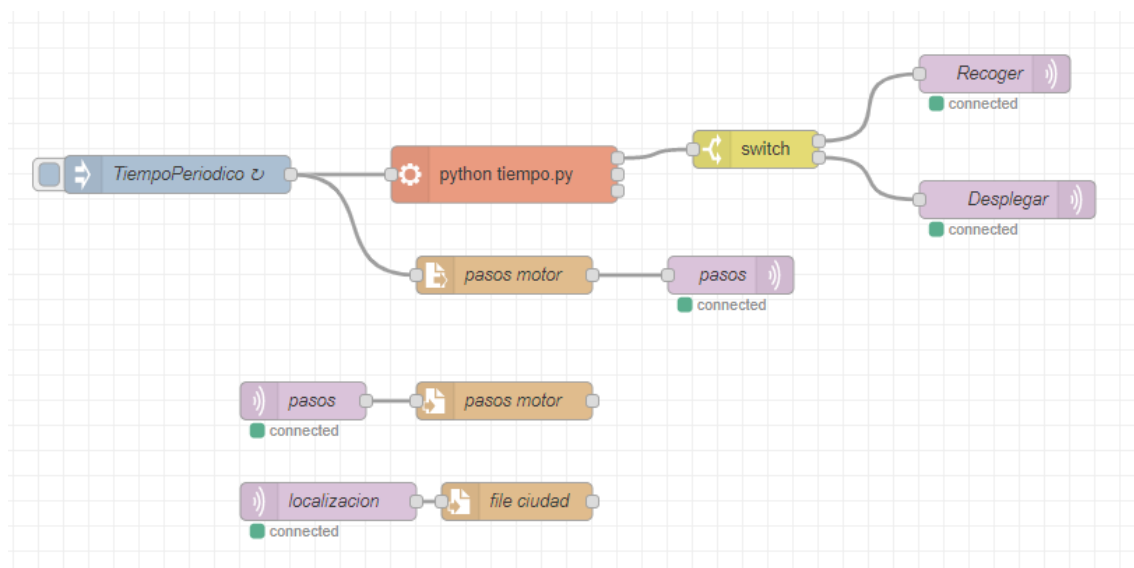


Figura 14. Diagrama de Node-Red para el control del motor paso a paso.

En el diagrama de la figura 14, se encuentra todo predefinido por lo que el usuario solo podrá cambiar el tiempo que transcurre entre cada ejecución del script tiempo.py (en caso de que se desee) el cual dispone de una configuración predefinida de 15min, esta modificación se puede realizar haciendo doble clic en el bloque “TiempoPeriódico” y modificando la casilla “every”.

Por otro lado, la localización del usuario puede variar de dos maneras distintas:

- Modificando el archivo “ciudad.txt” alojado en la Raspberry en el directorio “/home/pi”.
- A través de MQTT con el topic “casa/ubicacion”.

-Actualización y obtención de los precios de la luz eléctrica: Este programa se puede dividir en dos partes; la primera, asociada a la obtención de todos los precios de la red eléctrica a lo largo del día (mediante el script “precioLuz.py”) los cuales son almacenados en un fichero y actualizados a las “00:00” de cada día.

La otra parte del programa procesa los resultados obtenidos en el script anterior, para que se muestren en la interfaz de Home Assistant el valor más alto, el más bajo y el actual (variable con la hora) mediante el script “PrecioLuzActual.py”. En la figura 15 se puede observar el programa desarrollado para la obtención y actualización de los precios de la luz eléctrica.

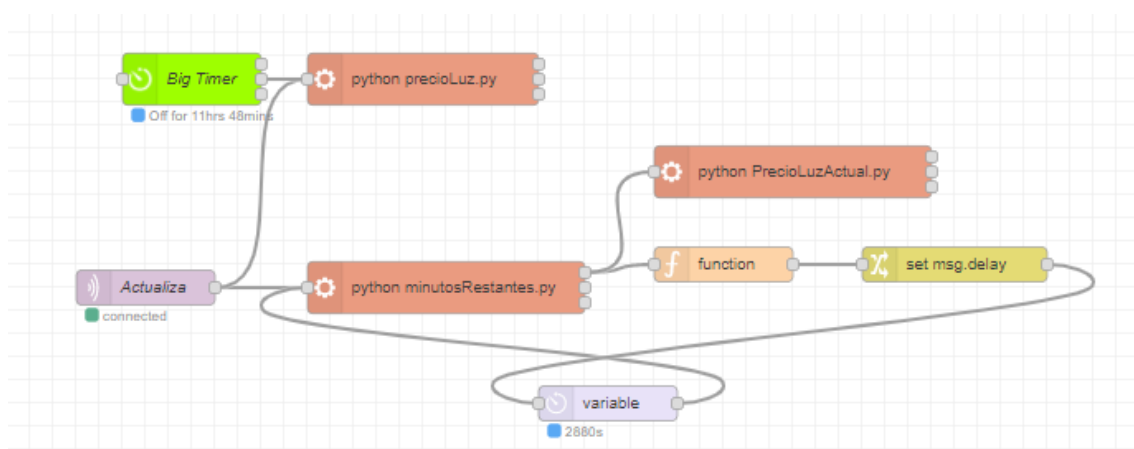


Figura 15. Diagrama de Node-Red para la obtención y actualización de los precios de la red.

-Switch dependiente del precio de la red eléctrica: En este diagrama se compara el valor introducido por el usuario (consigna) en la interfaz de Home Assistant y el precio actual de la luz eléctrica, si el valor introducido por este es superior al precio actual, el programa cierra el switch asociado y de igual manera ocurre en el caso contrario. En el programa que se presenta en la figura 16, realiza la comprobación del precio de la luz eléctrica periódicamente cada hora (cambio en el precio de la luz). El usuario solamente debe configurar el último bloque al que se le asocia la llamada al switch mediante MQTT en el caso de que se disponga de más de un modulo switch. Si el cliente desea incrementar el número de conmutadores, simplemente debe introducir otro bloque de envío MQTT en paralelo del que se dispone (casa/switch1) incluyendo el topic al que este esta asociado.

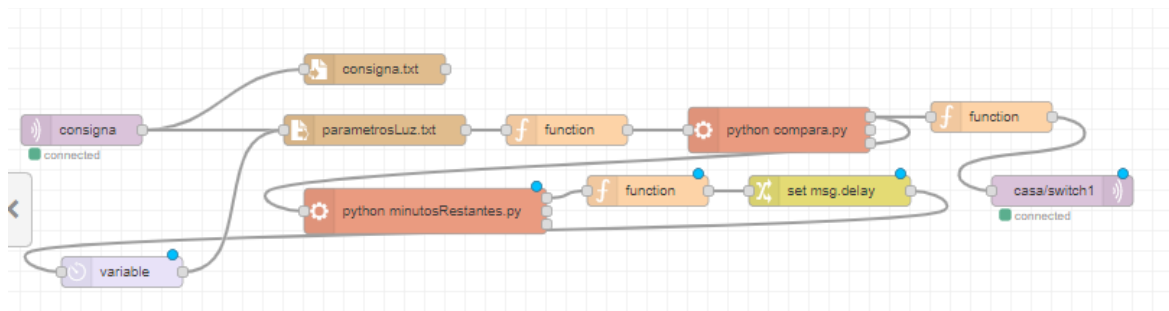


Figura 16. Diagrama de Node-Red para la conmutación de un switch condicionado al precio de la red eléctrica.

3.3.1.2. Home Assistant

La programación de Home Assistant se basa principalmente en el archivo “configuration.yaml” el cual se dispone ya configurado aunque, en caso de incrementar el número de conmutadores, se debe copiar la configuración que ya se ofrece y asignarle otro topic distinto a “casa/switch1”.

Para la visualización de los precios de la luz eléctrica, se debe modificar el fichero “scripts.yaml” y así habilitar un botón “PrecioLuz” para que este mande el mensaje de actualización necesaria en el primer inicio de la máquina, y así lanzar la ejecución del código de Node-Red mostrado anteriormente.

Por otro lado, para habilitar el cuadro donde se ingresa el valor de la variable “consigna”, necesaria para el correcto funcionamiento del programa “compara.py” (explicado en el apartado 3.3.3.4), es necesario la manipulación del archivo “automations.yaml”.

Todos los archivos mencionados anteriormente para el correcto funcionamiento de Home Assistant se encuentran en el repositorio del TFG.

3.3.2. NodeMCU

En este apartado se realiza una explicación del software asociado a la programación del microcontrolador.

Todos los programas que se han implementado, disponen de 3 variables que deben ser modificadas por el usuario; entre ellas se encuentra el nombre de la red (ssid), la contraseña de la misma (password) y la IP que se le ha asignado a la Raspberry para que se realice la comunicación MQTT (mqtt_server).

-Módulo meteo-ropa: El motor escogido para el prototipado, posee 4 bobinas con las cuales, según la excitación de estas, el rotor es movido. Para la manipulación de las bobinas, se ha optado por la utilización de una matriz donde esta manipulación se realiza a través de la conocida “secuencia medio paso” en que para un ciclo completo del rotor, son necesarios la secuenciación de 8 pasos para un mayor torque y precisión de movimiento del rotor (en comparación a la secuencia de paso completo).

Pasos/Bobina	B1	B2	B3	B4
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

Tabla 3.2.2.1. Excitación de las bobinas que conforman el motor paso a paso.

Un nivel alto (1) representa la bobina excitada en el motor. A cada paso que se realiza, el ángulo del rotor es girado 45° (en realidad estos grados de giro son mucho menos ya que este motor dispone de una reducción 1:64 para aumentar el torque de este por lo que son necesarios 64 pasos para un giro completo del rotor pero para la explicación se ha simplificado con 8 pasos). A continuación, en la figura 17, se procede a la representación gráfica de la evolución del giro del rotor en los 3 primeros pasos.

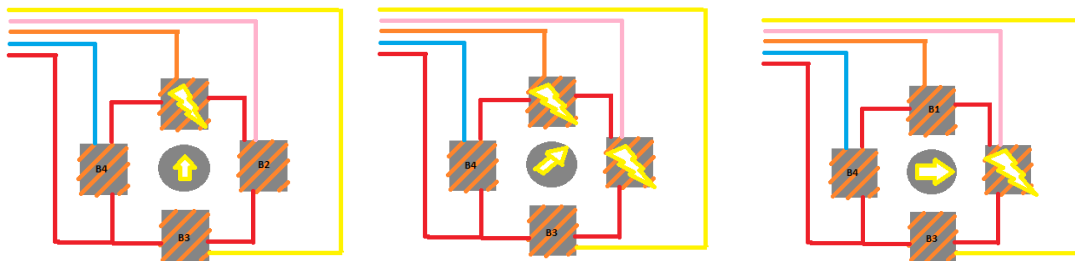


Figura 17. Representación del movimiento del rotor en los 3 primeros pasos.

Además, como ya se había comentado anteriormente, este módulo dispone de un botón que activa el modo “configuración” donde el motor comienza a girar hasta que vuelva a ser pulsado por el usuario, en ese intervalo, el microcontrolador realiza una cuenta de los pasos que han sido necesarios para alcanzar la posición deseada. Una vez obtenido dicho dato, NodeMCU manda un mensaje al topic “casa/motor/estado/pasos/guardar” donde la Raspberry guarda dicho valor en un fichero alojado en la misma.

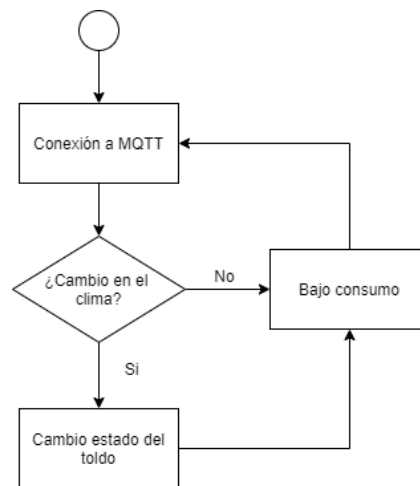


Figura 18. Diagrama de estados del módulo meteo-ropa.

En la figura 18 se puede observar la lógica implementada para el control del motor en función del tiempo meteorológico.

Módulo RGB: El slider que proporciona la integración de Home Assistant solamente dispone de un rango de 0 a 254, aunque NodeMCU precisa de un rango de 0 a 1023 por lo que tras la llegada del mensaje a la placa, se realiza una conversión para que esto sea posible y así poder manipular toda la luminiscencia de la que dispone la tira de LED.

3.3.3. Python

En esta sección se realiza un estudio de los programas que han sido necesarios para el control del sistema.

3.3.3.1. Tiempo.py

Mediante este script, se realiza la obtención de datos necesarios para el control del módulo meteo-ropa a través de la página www.eltiempo.es y se realiza la técnica conocida como web-scraping.

En primer lugar se realiza la lectura del fichero “ciudad.txt” el cual aloja la localización donde se encuentra el dispositivo. Seguidamente, se realiza la comprobación de si esta localización dispone de espacios para sustituir dichos espacios por guiones (esta acción es necesaria para la búsqueda de la información en la página web). Una vez realizadas las acciones mencionadas, se ingresa en la página web del tiempo para obtener el código fuente donde es almacenado en la variable “soup”. Posteriormente se realiza una búsqueda de la variable “precip_amount” que es la variable de la página del tiempo donde se observa si hay precipitaciones en esa zona; dicha variable puede contener varios valores “none” si no hay precipitaciones o en cualquier otro caso, indica la existencia de precipitaciones y su grado (chubascos, precipitaciones fuertes...) por lo que si el valor de esa variable es “none”, el programa devolverá “NO” y en cualquier otro caso, devolverá “SI” para indicar que hay precipitaciones.

3.3.3.2. PrecioLuz.py

En este script también se realiza de forma similar al anterior (tiempo.py) la obtención de los datos de la página web “www.tarifaluzhora.es” donde se captan los datos referentes al precio de la luz a lo largo de todo el día y los almacena en el fichero “luz.txt” en función de la hora. En el guardado de la información referente a los precios de la luz eléctrica, se realiza una conversión para que dichos valores sean almacenados en MW/h en lugar de kW/h.

3.3.3.3. PrecioLuzActual.py

En el script “PrecioLuzActual.py” se encarga de obtener el precio actual de la luz eléctrica consultándolo en el fichero “luz.txt” que ha sido rellenado a través del script mencionado en el anterior apartado. Además, realiza una lectura de los precios máximos y mínimos para obtener una orientación de los precios a lo largo del día. Una vez realizada dicha lectura, los almacena en el fichero “parametrosLuz.txt” para que puedan ser mostrados en la interfaz de Home Assistant.

3.3.3.4. Compara.py

El script, “compara.py”, realiza una comprobación del precio actual de la luz eléctrica y la consigna que se ha ingresado en Home Assistant para que si el valor de la consigna (ingresado por el usuario a través de Home Assistant) es superior al precio de la luz actual, se active un switch que se haya configurado y en caso contrario, lo apague. Esta funcionalidad permite el ahorro energético y económico del usuario ya que puede programar el encendido de un dispositivo que requiera de un alto consumo eléctrico en las horas donde el precio es inferior como por ejemplo un termo eléctrico.

3.3.3.5. MinutosRestantes.py

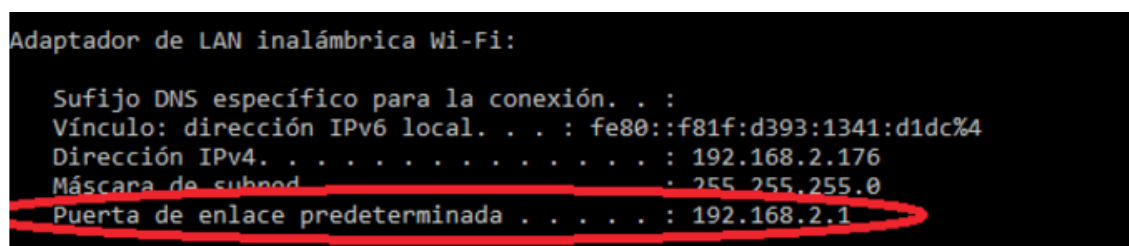
La programación de este script se ha utilizado de manera auxiliar para el primer inicio del sistema ya que hay ciertos programas que necesitan ser lanzados de manera periódica cada hora como lo son por ejemplo “precioLuzActual.py”. Este programa se encarga de calcular los minutos restantes hasta llegar a completar la hora.

3.4. Configuración fuera de LAN

En este apartado, se procede a la configuración necesaria para que el sistema pueda ser controlado fuera de la red LAN, es decir, desde cualquier localización, y no sea restringida al domicilio donde se encuentra el router que proporciona la conexión a la Raspberry y los microcontroladores.

3.4.1. Configuración del router

En cuanto a la configuración del router, a pesar de que la forma de proceder es distinta en función del tipo de router, las tareas a realizar son las mismas para todos los distintos modelos. Para el control del sistema fuera de LAN se debe abrir los puertos correspondientes a Home Assistant y opcionalmente (por si se desea modificar algún parámetro) a NodeRed; estos puertos son los 8123 y 1880 correspondientemente. Para la apertura de estos puertos se debe acceder al router mediante un navegador, introduciendo en la barra de búsqueda la dirección de este. En una terminal se debe ejecutar el comando “ifconfig” para sistemas Linux, o “ipconfig” para sistemas Windows; en la información que este comando ofrece, el dato asociado a “Puerta de enlace predeterminada” es el correspondiente a la dirección del router, como se muestra en la figura 19.



```
Adaptador de LAN inalámbrica Wi-Fi:
Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . : fe80::f81f:d393:1341:d1dc%4
Dirección IPv4. . . . . : 192.168.2.176
Máscara de subred. . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.2.1
```

Figura 19. Información del comando ipconfig.

Tras introducir los datos para acceder al router y el correspondiente logueo. Se debe buscar la opción “mapeo de puertos”/“port forwarding” o una opción similar a estas. Una vez encontrada esta opción rellenar los datos tal y como se muestran en la figura 20.

Configuración rápida Básica **Avanzada** Administración Cerrar sesión

Control parental
Firewall
NAT
→ Mapeo de Direcciones
→ **Mapeo de Puertos**
→ Aplicaciones Especiales
→ Tabla de Mapeo NAT
DDNS
USB
UPnP y DLNA
Notificación por Correo
Telefonía

Mapeo de Puertos
Es posible configurar el router como un servidor que redirija todas las peticiones que se realicen desde Internet a servicios determinados (FTP, HTTP, etc.) de tu Red Local a ciertos equipos (direcciones IP locales) y puertos. Es decir, dependiendo de qué tipo de servicio se trate (qué puerto TCP/UDP), el router asignará ese servicio a un servidor concreto de la red local identificada por su dirección IP local y puerto para tratarlo. Es posible configurar rangos de puertos, una lista de ellos, o una combinación de ambos.
Por ejemplo:
Rango de puertos: ej. 100-150
Lista de puertos: ej. 25,110,80
Combinación: ej. 25-100,80

No.	LAN IP	Tipo de Protocolo	Puerto en la LAN	Puerto público	Activar	Configurar
1	192.168.2.58	TCP&UDP	8123	8123	✓	 
2	168.168.2.58	TCP&UDP	1880	1880	✓	 
<div> <div>↑</div> <div>↑</div> <div>↑</div> </div> <div> Ip de Raspberry Puertos asociados a Home Assistant y NodeRed. </div> <div>  </div>						

Cancelar

Figura 20. Apertura de puertos.

3.4.2 Configuración DNS

Una vez realizada la apertura de puertos, se dispone de un control desde fuera de LAN aunque para acceder al sistema se debe introducir la IP pública en el buscador, esta acción puede ser incómoda ya que se debe recordar la dirección IP pública siempre que se quiera acceder a esta. Una solución es el uso de DNS (Domain Name Space) donde a esa IP pública se le “asigna un nombre” permitiendo que se pueda acceder como cualquier dominio de internet. En la elaboración de este TFG se ha utilizado la herramienta No-IP, ya que esta dispone de tres DNS completamente gratis durante un mes (cada una).

Para usar la herramienta descrita, es necesario crear una cuenta. Una vez se ha realizado el registro, se debe rellenar el campo “Hostname” con el nombre que se le desea asignar, el tipo de dominio y click en “Add Hostname”, en este caso se le ha asignado “tfgjuanma2021.ddns.net”. A continuación, se procede a la configuración de esta DNS, para ello, es necesario introducir los datos tal y como aparece en la figura 21.

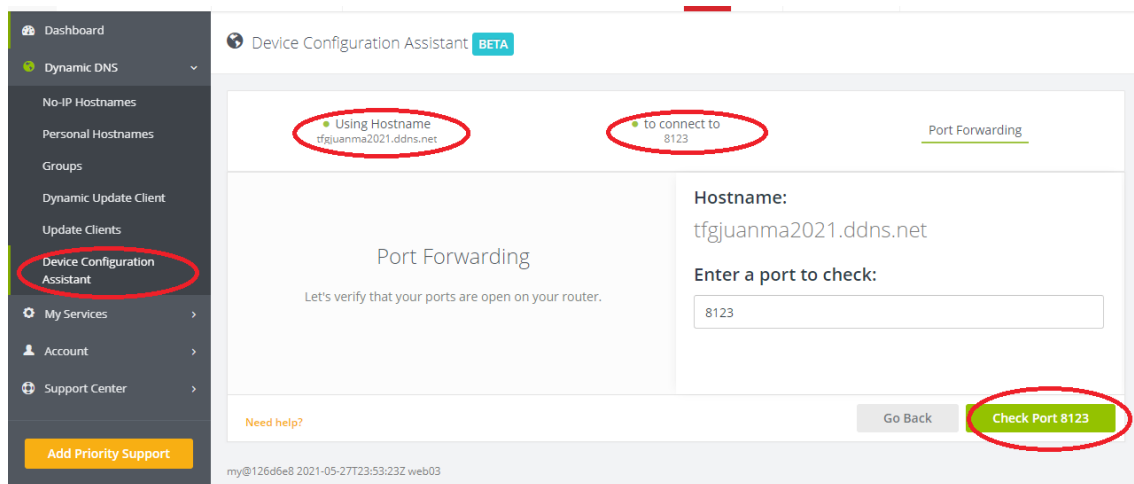


Figura 21. Configuración de la DNS en No-IP.

Una vez que se ha realizado el chequeo del puerto correspondiente a Home Assistant (8123), la DNS queda configurada.

Posteriormente se debe volver a la configuración del router para guardar la DNS generada, esta configuración se puede observar en la figura 22.



Figura 22. Configuración de la DNS en el router.

Ahora sí, queda todo configurado y preparado para acceder al sistema fuera de LAN y con una DNS asociada.

Capítulo 4. Verificación y pruebas

En este capítulo se han descrito las pruebas realizadas al sistema para verificar su correcto funcionamiento.

4.1. Sistema de pruebas

En este apartado se procede a detallar las características tanto software como hardware que se ha utilizado para la realización de pruebas realizadas.

PC: Lenovo Ideapad 310-15 IKB con una partición de Ubuntu.

Smartphone: BQ Aquaris U plus.

Hardware: Keysight E36313A.

4.2. Pruebas realizadas

A continuación se recoge las pruebas realizadas para la comprobación de los requisitos y el resultado obtenido de dichas pruebas.

ID del requisito	Descripción de la prueba	Resultado
RF1	Mediante el teléfono, se ha accedido al panel de Home Assistant a través de los datos móviles.	Favorable.
RF2	A través de una terminal, se ha estudiado el correcto comportamiento de los distintos actuadores de cada módulo con el envío de mensajes a los topics correspondientes.	Favorable.
RF3	Sin ninguna manipulación de la Raspberry, se intenta acceder al panel de Home Assistant y utilizar las distintas funcionalidades.	Favorable.
RF4	A través de un navegador, se accede al sistema y se observa si Home Assistant solicita una autenticación.	Favorable.

RF5	Realización de una encuesta a distintos usuarios de distintos rangos de edad sobre el uso de la aplicación Home Assistant, donde se solicita una valoración tras su uso.	Favorable.
RF6	Medición del consumo mediante la fuente Keysight E36313A.	Favorable.
RF7	Contratar el servicio de fabricación PCB y tras el montaje de los componentes pertinentes, verificar el correcto funcionamiento.	No se ha podido verificar.
RF8	Observar el correcto funcionamiento de los scripts generados donde se requiera web-scraping mediante el mostrado por pantalla de los datos.	Favorable.
RF9	A través de simulaciones, observar los efectos de los circuitos de protección que se han desarrollado.	Favorable.
RF10 y RF13	Mediante el panel de Home Assistant, verificar que el comportamiento de la tira LED se corresponde con lo esperado.	Favorable.
RF11 y RF14	Verificación de la correcta activación y desactivación del relé a través del panel de Home Assistant.	Favorable.
RF12 y RF15	En el archivo "ciudad.txt" incluir una ciudad donde esté lloviendo y comprobar que el motor realiza los movimientos necesarios para desplegar el toldo. Posteriormente, comprobar que realiza el giro inverso tras cambiar la localización a una ciudad donde no haya precipitaciones en ese momento.	Favorable.

Tabla 4.1. Tabla de las pruebas realizadas.

Para la comprobación del RF5, se ha realizado una encuesta a 2 personas de cada rango de edad propuestos (15-25, 26-36, 37-47, 48-58). El resultado final ha sido una puntuación media de 4/5; la encuesta realizada se puede encontrar en el apéndice B.

En el RF7 no se ha podido verificar los diseños PCB que se han desarrollado debido a la falta de tiempo y a los costes que demandaban su fabricación. En el repositorio del TFG se encuentran los gerbers necesarios para visualizar el diseño PCB de cada uno de los módulos.

Capítulo 5. Manual de instalación y uso

5.1. Requisitos de instalación

A continuación se recoge una lista de requisitos mínimos, hardware y software necesario para poder reconstruir la implementación del proyecto que se ha llevado a cabo durante el TFG:

Hardware:

- Raspberry y su correspondiente transformador para la alimentación del microprocesador.
- Módulos NodeMCU (desarrollados en el TFG).
- Tarjeta micro SD de al menos 8GB y clase 10.
- Adaptador de tarjetas micro SD.
- PC con al menos 4 GB libres.
- Conexión a Internet.
- Cable USB-micro USB.
- Cable Ethernet.

Software:

- Arduino IDE (o similar).
- Raspberry Pi Imager.
- Advanced IP Scanner.
- VNC Viewer.

5.2. Manual de instalación

A continuación, se detalla los pasos a seguir para montar el sistema desde cero y recrear el proyecto desarrollado en este TFG.

5.2.1. Raspberry

En primer lugar, es necesario insertar la tarjeta micro SD en el ordenador a través del adaptador correspondiente. Posteriormente, se realiza la ejecución del software oficial de Raspberry (Raspberry Pi Imager); clicar la opción “choose OS” → “use custom” (opción del panel emergente) y seleccionar el archivo “ImagenCompletaRaspberryTFG.img”, pinchar la casilla “choose SD card” para indicar la tarjeta SD insertada anteriormente y pulsar “write” tal y como se muestra en la figura 23.



Figura 23. Software Raspberry Pi Imager.

Tras completar la carga de escritura y verificación, desmontar/expulsar la tarjeta de forma segura del equipo tal y como se muestra en la figura 24.

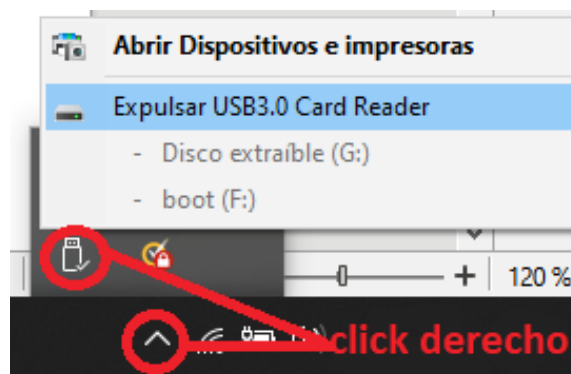


Figura 24. Expulsar SD de forma segura.

Para el inicio de la centralita, es necesario insertar la tarjeta micro SD en la ranura correspondiente, conectar el cable Ethernet y el transformador para alimentar la Raspberry. En la figura 25 se pueden observar la disposición de los puertos que se han mencionado anteriormente.

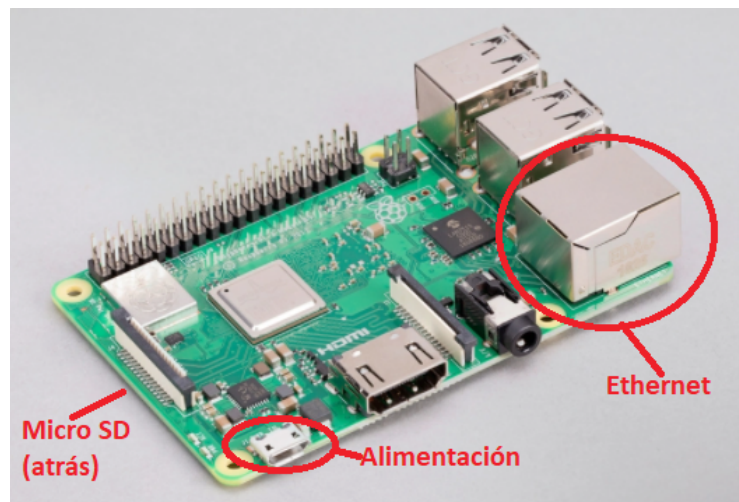


Figura 25. Puertos de Raspberry mencionados (Fuente: raspberrypi.org).

Tras el inicio de la centralita, ejecutar el software “Advanced IP Scanner” para visualizar la dirección IP que el router ha asignado a la centralita. Pulsar el botón “explorar” en el programa mencionado y esperar hasta obtener un resultado similar al que se muestra en la figura 26.

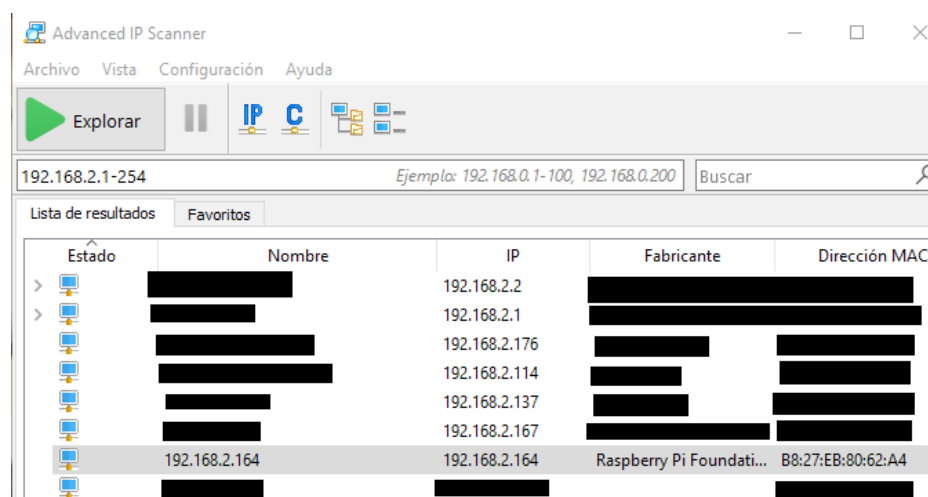


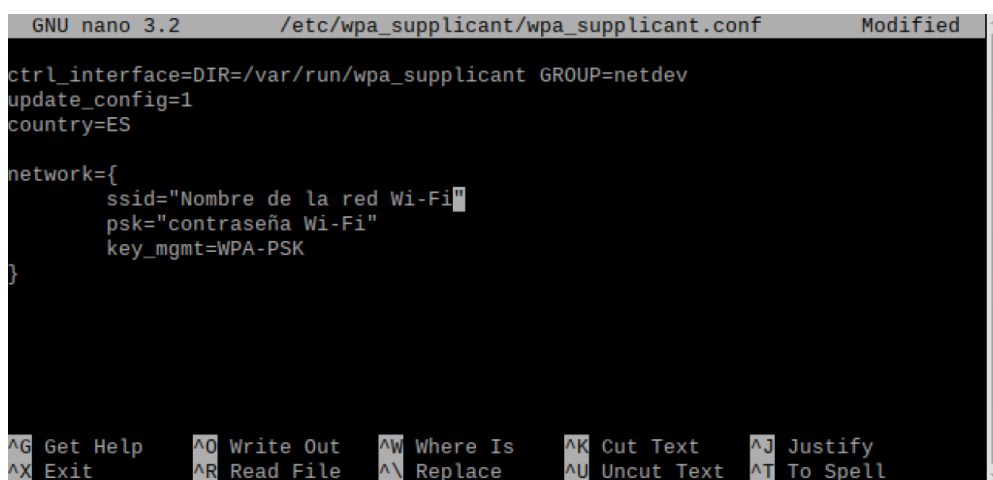
Figura 26. Resultados obtenidos con Advanced IP Scanner.

5.2.2. Conexión VNC Raspberry

En este apartado se realiza una conexión VNC para poder visualizar el entorno de Raspberry y así poder realizar modificaciones de ficheros y conexión Wi-Fi entre otras opciones.

Para poder visualizar el entorno de la centralita, es necesario el software “VNC Viewer”, tras su instalación, pinchar en la pestaña Archivo → nueva conexión y rellenar las 2 primeras casillas que aparecen en la ventana emergente con la IP de la Raspberry y un nombre a escoger (opcional) respectivamente, hacer click en aceptar; con el dispositivo encendido, hacer doble click en el elemento creado y aceptar las ventanas emergentes hasta que aparezca la ventana de autenticación (usuario: pi, contraseña: raspberry) y posteriormente el escritorio, en la ventana de VNC Viewer.

Para realizar la conexión Wi-Fi, abrir una terminal y ejecutar el comando “`sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`” rellenar este archivo tal y como se muestra en la figura 27. Se recomienda establecer una IP estática tal y como se explica en el apartado 3.3.1.



```
GNU nano 3.2 /etc/wpa_supplicant/wpa_supplicant.conf Modified
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=ES

network={
    ssid="Nombre de la red Wi-Fi"
    psk="contraseña Wi-Fi"
    key_mgmt=WPA-PSK
}

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit          ^R Read File    ^_ Replace      ^U Uncut Text   ^T To Spell
```

Figura 27. Configuración del archivo wpa_supplicant.conf.

Una vez rellenados los campos pertinentes, pulsar la combinación “Ctrl+o” y después, pulsar enter para guardar los cambios realizados en el fichero; combinar “Ctrl+x” para salir del editor del archivo y reiniciar la Raspberry. Una vez que se ha reiniciado la Raspberry, volver a repetir los pasos de utilización de los softwares “Advanced IP Scanner” y “VNC Viewer” ya que al realizar la conexión mediante Wi-Fi, la dirección IP de la Raspberry, es modificada.

5.2.3. NodeMCU

Una vez obtenida la dirección ip de la Raspberry, abrir Arduino IDE y seleccionar en la pestaña Archivo → Preferencias. Tras la apertura de la ventana, copiar “https://arduino.esp8266.com/stable/package_esp8266com_index.json” en la casilla de “Gestor de URLs Adicionales de Tarjetas” así como en la pestaña de red, marcar la casilla “Sin proxy” y pulsar “Ok” tal y como se muestra en la figura 28.

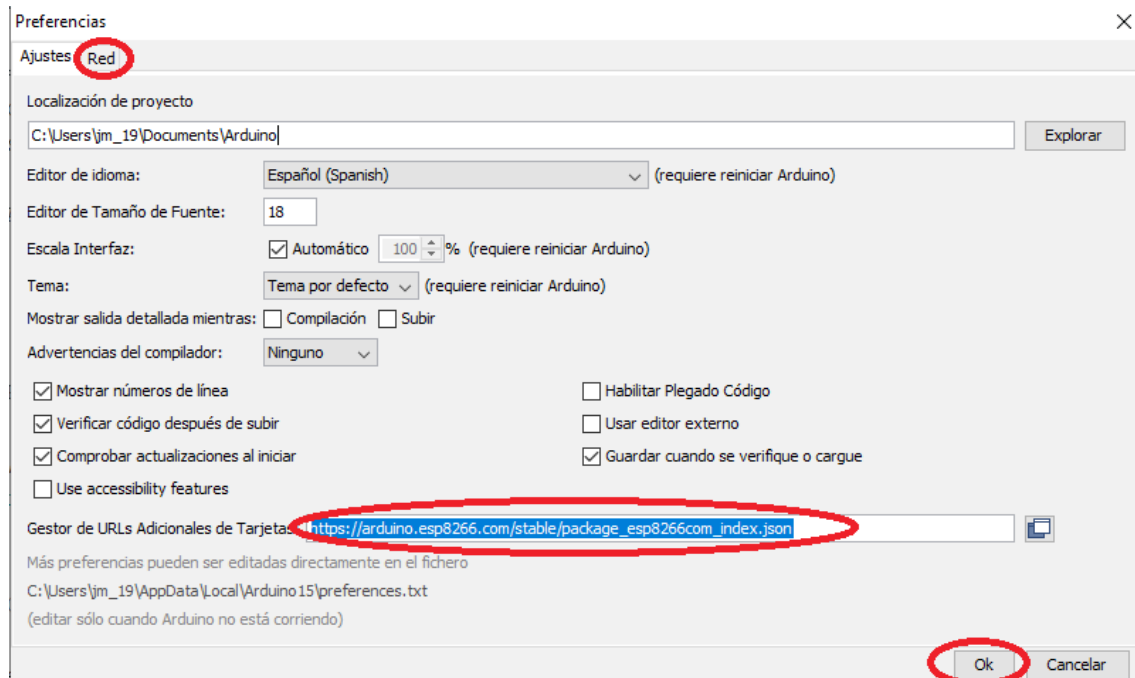


Figura 28. Ventana de preferencias de Arduino IDE.

A continuación, pulsar en la pestaña Herramientas → Placa... → Gestor de tarjetas. Mediante el buscador, instalar la placa “esp8266”. Tras la instalación de dicha tarjeta, verificar los parámetros del desplegable de herramientas después de seleccionar la placa “NodeMCU 1.0 (ESP-12E Module)” como se muestra en la figura 29.

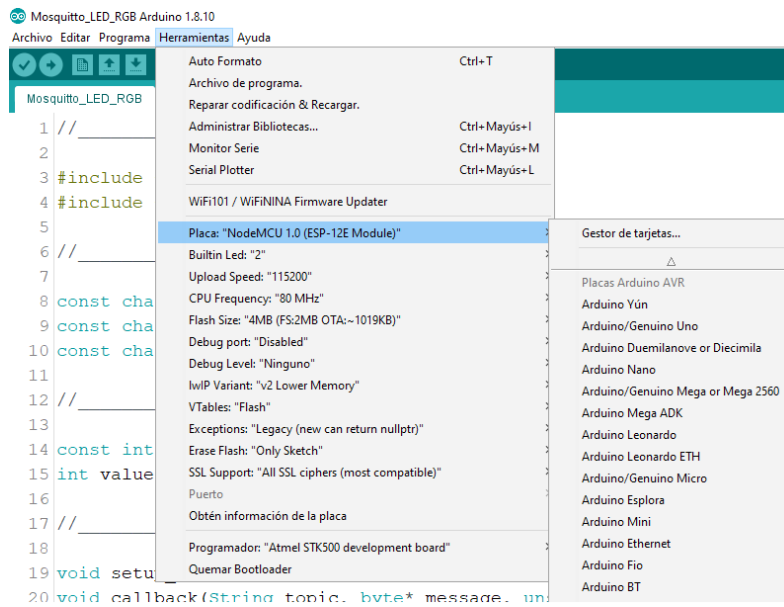


Figura 29. Configuración parámetros NodeMCU.

Seleccionar: Programa → Incluir librería → Administrar bibliotecas e instalar la librería “PubSubClient.h by Nick O’Leary”. En la figura 30 se puede observar la librería detallada.

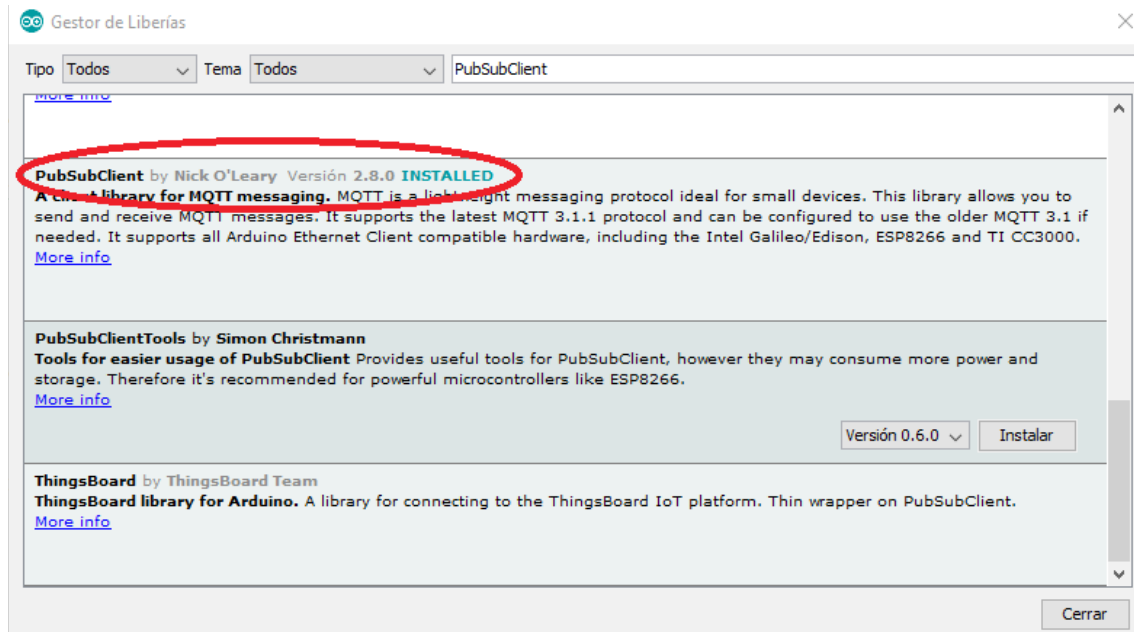


Figura 30. Librería “PubSubClient”.

Una vez realizados todos los procesos anteriores, abrir cualquiera de los programas desarrollados para este proyecto y modificar las líneas 8, 9 y 10 para configurar los parámetros necesarios para la conexión Wi-Fi y con el broker MQTT(Raspberry).

```
1 // _____LIBRERIAS_____//
2
3 #include <ESP8266WiFi.h>
4 #include <PubSubClient.h>
5
6 // _____VALORES RED WIFI_____//
7
8 const char* ssid = "Nombre Red Wi-Fi";
9 const char* password = "Contraseña Wi-Fi";
10 const char* mqtt_server = "IP Raspberry";
11
```

Figura 31. Parámetros a cambiar en los programas de NodeMCU.

Tras modificar los parámetros de la figura 31, conectar el microcontrolador NodeMCU y seleccionar Herramientas → Puerto → COM X y hacer click en la casilla “Subir” (ubicada en la parte superior izquierda con forma de flecha).

5.2.4. Home Assistant

Para acceder a Home Assistant, es necesario introducir la IP de la Raspberry seguida de “:8123” en el navegador web del dispositivo elegido. En la pantalla de inicio, ingresar usuario “juanma” y contraseña “tfg2021”, este es el usuario por defecto como administrador; para agregar nuevos usuarios: configuración → personas → click en “añadir persona” y rellenar los campos pertinentes.

La configuración del panel de control que se puede observar en la figura 32, es la proporcionada por defecto para el control de los 3 módulos diseñados a lo largo del proyecto así como del control de otras funcionalidades.

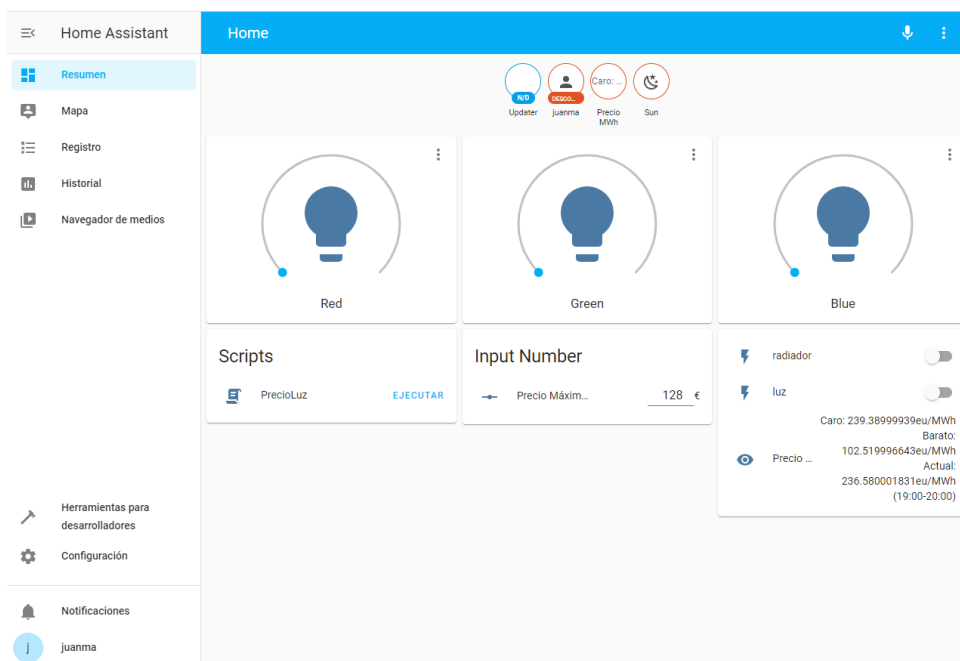


Figura 32. Panel de control de Home Assistant por defecto en este TFG.

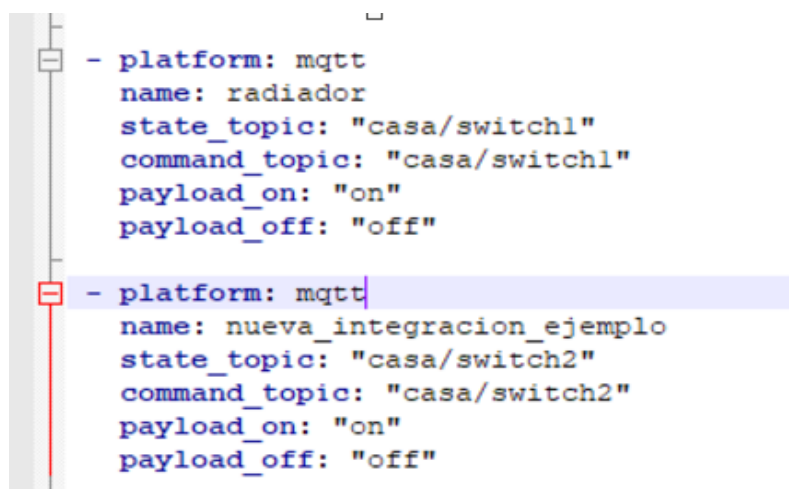
Para una correcta visualización de los parámetros de la luz eléctrica, es necesario hacer click en “EJECUTAR” de la tarjeta Scripts. Una vez actualizado el tramo horario y verificando que se corresponde con el tramo horario actual, no hará falta esta tarjeta, pudiéndose incluso eliminar del panel.

5.2.5. Nuevas integraciones

Si se desea incluir más interruptores inteligentes o cualquier otro módulo es necesaria la modificación del archivo `configuration.yaml`, mediante el comando

```
"sudo geany  
/home/homeassistant/.homeassistant/configuration.yaml"
```

(dicho comando se debe ejecutar a través de una terminal y con la ayuda del software VNC Viewer para poder visualizar y acceder a esta). Una vez ejecutado el comando, se abrirá el editor de textos donde se puede cambiar el nombre de las interfaces y otras opciones. Para incrementar, por ejemplo el número de interruptores inteligentes como se había mencionado, es necesario copiar desde la línea 81 a la 86 (ambas inclusives) y pegar dicha selección posteriormente a esta última línea, es decir, a partir de la línea 87; posteriormente, modificar los parámetros "name", "state_topic" y "command_topic" para asignar un nombre al interruptor y asignar un topic al nuevo interruptor. En la figura 33 se puede observar un ejemplo de la integración de un nuevo interruptor.



```
- platform: mqtt  
  name: radiador  
  state_topic: "casa/switch1"  
  command_topic: "casa/switch1"  
  payload_on: "on"  
  payload_off: "off"  
  
- platform: mqtt  
  name: nueva_integracion_ejemplo  
  state_topic: "casa/switch2"  
  command_topic: "casa/switch2"  
  payload_on: "on"  
  payload_off: "off"
```

Figura 33. Ejemplo de nueva integración para un interruptor en Home Assistant.

Por otro lado, si se ha realizado la integración de un nuevo interruptor que se desea controlar según el precio de la luz eléctrica, acceder al panel de Node-Red con la dirección IP de la Raspberry, seguido de dos puntos 1880 y copiar el flow "Encendido/apagado según precio" en uno nuevo, modificando los elementos señalados en la figura 34.

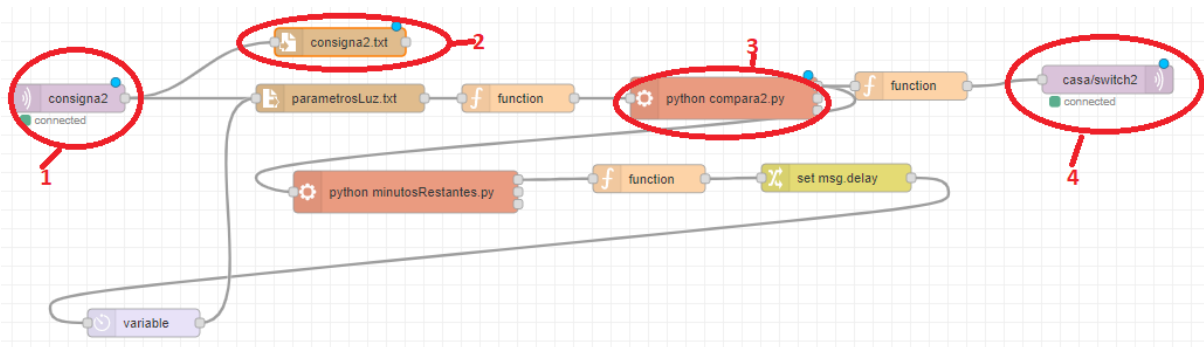


Figura 34. Modificaciones del flow “Encendido/apagado según precio” para una nueva integración.

-Elemento 1 y 4: Modificar el campo “topic” (en el ejemplo se le ha asignado “consigna2” y “casa/switch2”).

-Elemento 2 y 3: Modificar el nombre del archivo.

Clicar en la casilla roja situada en la parte superior derecha “Deploy”.

Seguidamente, se debe crear un archivo de texto vacío en el directorio /home/pi con el nombre que se le haya asignado al elemento 2.

Por otro lado, se debe copiar el programa “compara.py” modificando la línea 3 por el nombre del archivo de texto anteriormente creado y modificando el nombre de este programa por el que se le haya ingresado en el elemento 3.

Por otra parte, copiar el contenido del archivo “automations.yaml” de la misma manera a como se ha realizado anteriormente en la integración de un nuevo interruptor inteligente, modificando los parámetros “alias”, “entity_id”, “topic” y “payload_template” tal y como se muestra en la figura 35.

```

1  - alias: "nuevo valor consigna"
2    trigger:
3      platform: state
4      entity_id: input_number.box1
5    action:
6      service: mqtt.publish
7      data:
8        topic: consigna
9        payload_template: "{{ states('input_number.box1') | int }}"
10
11 - alias: "nuevo valor consigna2"
12   trigger:
13     platform: state
14     entity_id: input_number.box2
15   action:
16     service: mqtt.publish
17     data:
18       topic: consigna2
19       payload_template: "{{ states('input_number.box2') | int }}"
20

```

Figura 33. Resultado del archivo “automations.yaml” tras una nueva integración.

En último lugar, habría que incluir en el fichero “configuration.yaml” la copia de la tarjeta “input_number” (línea 115 a 121) cambiando la variable “name” y “topic”. Posteriormente, se debe reiniciar el servicio de Home Assistant a través del comando “sudo systemctl restart home-assistant@homeassistant.service” e integrar la nueva tarjeta al panel de Home Assistant.

Capítulo 6. Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones que se han obtenido en el desarrollo del proyecto, donde se realiza un pequeño análisis de posibles desarrollos y mejoras del sistema para futuros desarrolladores.

A lo largo de esta memoria se ha desarrollado un sistema domótico con alta capacidad de integración, escalabilidad y customización, donde un sistema central, basado en Raspberry, se encarga de realizar las comunicaciones y procesamiento de información entre el usuario y los actuadores presentados. Así mismo se ha realizado el diseño software y hardware del proyecto, incluyendo el diseño PCB de cada uno de los módulos. Cada módulo es controlado a través del microcontrolador NodeMCU, el cual se comunica a la centralita con el protocolo MQTT a través del envío/recepción de mensajes a topics.

Por otro lado, los conocimientos adquiridos a lo largo de este proyecto han sido ampliamente variados.

En primer lugar se ha realizado el uso de web-scraping que supone un amplio abanico de posibilidades a la hora de realizar integraciones que aprovechen la obtención de información de otras páginas webs; en este proyecto se ha utilizado de 2 maneras diferentes, una ha sido para obtener el tiempo meteorológico de una localización. La otra manera de aprovechar el web-scraping, ha sido consultado los precios de la red eléctrica.

A pesar de haber trabajado anteriormente con el protocolo MQTT, en este trabajo es donde se ha podido verificar la capacidad de este, y su fácil manejo.

El uso de Python, ha supuesto una nueva línea de investigación para así aprender otro lenguaje de programación que se encuentra en auge y que cada vez es más solicitado.

Gracias al manejo del microcontrolador NodeMCU, se han observado las grandes posibilidades que brinda el ESP8266 por su comunicación con otros dispositivos de manera inalámbrica y la fácil programación similar a la de Arduino.

El desarrollo de un sistema basado en Linux, ha supuesto otro reto que se ha convertido en un incremento en el conocimiento de este sistema operativo tanto en su manejo como en su programación. Para este proyecto se ha realizado investigaciones para la comunicación Wi-Fi, establecer la IP estática, desarrollo de servicios, generación de scripts, uso de paquetes específicos, entre otras.

El uso de herramientas como Home Assistant y Node-Red, han supuesto al principio un gran reto que con el transcurso de su utilización, se han convertido en herramientas muy útiles y que han supuesto un gran aporte al conocimiento.

Las búsquedas en Internet y la selección de la información, han supuesto otra mejora notable que han desarrollado una agilidad en cuanto a las investigaciones.

Por otro lado, me gustaría realizar una serie de anotaciones para mejorar el sistema para futuros desarrolladores, como son el uso de algunas integraciones de Home Assistant como la de presencia de usuarios, que puede ser muy interesante a la hora de integrar un sistema domótico (como por ejemplo apagar ciertos dispositivos si no se encuentra ningún usuario en el hogar). El control de climatización, supone otra línea importante en el ahorro energético, por lo que la integración de un termostato puede suponer una gran aportación al sistema. Otra integración interesante de Home Assistant, es la que ofrece información acerca de cuando sale el Sol y cuando se pone, por lo que podría utilizarse también para el ahorro energético donde podría salir algún tipo de aviso en el panel si se ha activado alguna luz en una franja horaria donde se puede aprovechar la luz solar.

Por otro lado, la polivalencia que ofrece un sistema embebido como Raspberry, ofrece una gran capacidad de integración que puede ser aprovechada con el uso de herramientas como Gstreamer para incorporar una cámara y así, conformar una pequeña integración de seguridad.

Apéndice A. Presupuesto de elaboración

A continuación se detalla una serie de tablas donde se puede observar el coste de los materiales y equipos necesarios para el desarrollo de este proyecto así como de las licencias utilizadas (aunque se ha optado siempre por el uso de software gratuito o con licencias gratuitas limitadas).

Componente	Coste unitario (€)	Unidades empleadas	Coste total (€)
NodeMCU	3.68	3	11,04
Raspberry Pi 3 B+	37.08	1	37.08
Fuente de alimentación Raspberry Pi 3 B+	5.39	1	5.39
Disipadores de calor Raspberry Pi 3	1.2	1	1.2
ULN2003	1	2	2
Tira LED RGB	12	1	12
Motor paso a paso	3.75	1	3.75
Transformador AC-DC	0.66	3	3.06
Transistor 2n2222	0.76	1	0.76
Relé	0.87	1	0.86
Diodo 1n4007	0.19	1	0.19
Resistencias	0.13	3	0.39
Placa perforada de puntos	0,44	3	1,32
Conector jack alimentación DC	0.87	1	0.87
Conectores macho y hembra pin header PCB (kit)	7.99	1	7.99
Cable de cobre (10m)	4.95	1	4.95
Sensor DHT11	3.3	1	3.3
Pulsador	0.55	1	0.55
Conector pcb k128	0.3	4	1.2
Protoboard	6	1	6
Total			103,9

Tabla en Apéndice A.1. Coste de componentes y materiales

Equipo/software		Coste unitario (€)	Periodo de amortización (meses)	Periodo de uso (meses)	Coste total (€)
Lenovo	Ideapad	611.35	48	6	74,41875
310-15					
Raspbian		0	6	6	0
Raspberry	Pi	0	6	6	0
Imager					
Home Assistant		0	6	6	0
Node-Red		0	6	6	0
OrCad Lite		0	6	6	0
EasyEDA		0	6	6	0
Arduino IDE		0	6	6	0
No-IP		0	1	1	0
Fritzing		0	6	6	0
Advanced IP Scanner		0	6	6	0
VNC Viewer		0	6	6	0
				Total	74,41875

Tabla A.2. Tabla en Apéndice A.2. Amortización de equipos y licencias de software

Actividad		Horas
Coordinación del TFG		18
Análisis de requisitos		7
Desarrollo hardware/software del prototipo		100
Realización de las pruebas		4
Redacción de la memoria		70
Preparación de la presentación		5
Horas totales		204
Coste por hora	20€	Coste total de la mano de obra 4080€

Tabla en Apéndice A.3. Horas invertidas en la realización del proyecto (mano de obra)

Concepto	Coste (€)
Coste de componentes y material fungible	97,9
PC y Software	74,41875
Mano de obra	4080
Coste total	4252,31875

Tabla en Apéndice A.4. Presupuesto total

Apéndice B. Encuesta de uso del sistema

Redondee con un círculo el rango de edad en el que se encuentra:

[15-25]

[26-36]

[37-47]

[48-58]

A continuación se le solicita una serie de funciones a realizar, tras su ejecución, evalúe en una escala de 0-5 el nivel de complejidad que le ha supuesto, siendo el 0 nada complejo y 5 muy complejo.

1) Incremente la intensidad lumínica roja de la tira LED.

Nota:

2) Encienda el radiador.

Nota:

3) Establezca el encendido del radiador cuando el precio de la luz eléctrica sea inferior a 105 €/MWh.

Nota:

4) ¿Podría indicar cuál es el precio actual de la red eléctrica?

Nota:

5) Establezca una valoración del 0 al 5 sobre lo intuitivo que le parece el panel de control.

Nota:

Observaciones (opcional):

Referencias

- [1] CEDOM, “Qué es domótica”, URL: <http://www.cedom.es/sobre-domotica/que-es-domotica> (última consulta el 5/Junio/2021).
- [2] Dag Spicer, “The ECHO IV home computer: 50 years later”, URL: <https://computerhistory.org/blog/the-echo-iv-home-computer-50-years-later> (última consulta el 5/Junio/2021).
- [3] CEDOM, “Estudio de mercado Sector de la Domótica e Inmótica”, Noviembre 2020, URL: <http://www.cedom.es/sobre-domotica/publicaciones/cedom-estudio-de-mercado-2019> (última consulta el 5/Junio/2021).
- [4] Fran Leal, “El móvil ya acapara mas de la mitad del tráfico web del mundo”, *elPeriódico*, 6 Octubre 2020, URL: <https://www.elperiodico.com/es/activos/innovadores/20201006/movil-acapara-mas-mitad-trafico-web-informe-ditrendia-8144430> (última consulta el 5/Junio/2021).

Bibliografía

- [1] Gunther Gridling, Bettina Weiss , “Introduction to Microcontrollers”, Vienna University of Technology Institute of Computer Engineering Embedded Computing Systems Group, 26 Febrero 2007.
- [2] Sistemas empotrados, Home Assistant y NodeRed, Departamento de tecnología electrónica, Universidad de Málaga, curso 2019-2020.
- [3] Instrumentación electrónica 2, Presentación Introducción KNX, Departamento de tecnología electrónica, Universidad de Málaga, curso 2019-2020.
- [4] Christopher Hallinan. *Embedded Linux Primer*. Segunda edición. PRENTICE HALL, 2010.
- [5] Catalog No. BCM2837B0, Raspberry Pi 3 B+, Raspberry, UK.
- [6] NodeMCU, “NodeMCU Documentation”, URL: <https://nodemcu.readthedocs.io/en/release/> (última consulta el 5/Junio/2021).
- [7] Catalog No. J182-E1-04, G6RN PCB Power Relay, OMRON Corporation, Japón.
- [8] Catalog No. DocID16558, 2N2222AHR, STMicroelectronics, Suiza.
- [9] Catalog No. DS28002, 1N4001/L - 1N4007/L, Diodes Incorporated, Texas (EEUU).
- [10] ULN2001A-ULN2002A ULN2003A-ULN2004A, STMicroelectronics, Suiza.
- [11] Leonard Richardson, “Beautiful Soup Documentation”, URL: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/#> (última consulta el 5/Junio/2021).

[12] Node-RED, “Documentation”, URL: <https://www.nodered.org> (última consulta el 25/Junio/2021).

[13] Home Assistant, “Documentation”, URL: <https://www.home-assistant.io/docs> (última consulta el 25/Junio/2021).