



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

**“IMPLEMENTACIÓN DE UN MÓDULO DOMÓTICO
ENFOCADO A LA EFICIENCIA ENERGÉTICA DE HOGARES,
UTILIZANDO SENSORES Y ACTUADORES CONVENCIONALES
CONTROLADOS INALÁMBRICAMENTE POR UNA TARJETA
DE DESARROLLO”**

Trabajo de integración curricular.

Tipo: Dispositivo Tecnológico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTOR:

LUIS MARCELO PILLAJO COLCHA.

Riobamba - Ecuador

2022



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

**“IMPLEMENTACIÓN DE UN MÓDULO DOMÓTICO
ENFOCADO A LA EFICIENCIA ENERGÉTICA DE HOGARES,
UTILIZANDO SENSORES Y ACTUADORES CONVENCIONALES
CONTROLADOS INALÁMBRICAMENTE POR UNA TARJETA
DE DESARROLLO”**

Trabajo de integración curricular:

Tipo: Dispositivo Tecnológico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTOR: LUIS MARCELO PILLAJO COLCHA.

DIRECTOR: ING. PABLO EDUARDO LOZADA YÁNEZ.

Riobamba - Ecuador

2022

©2022, Luis Marcelo Pillajo Colcha.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Luis Marcelo Pillajo Colcha, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 22 de marzo de 2022.



.....
Luis Marcelo Pillajo Colcha

C.I: 160057703-3

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

El Tribunal de trabajo de titulación certifica que: El trabajo de titulación: Tipo: Dispositivo Tecnológico, “**IMPLEMENTACIÓN DE UN MÓDULO DOMÓTICO ENFOCADO A LA EFICIENCIA ENERGÉTICA DE HOGARES, UTILIZANDO SENSORES Y ACTUADORES CONVENCIONALES CONTROLADOS INALÁMBRICAMENTE POR UNA TARJETA DE DESARROLLO**”, de responsabilidad del señor **LUIS MARCELO PILLAJO COLCHA**, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

	FIRMA	FECHA
Ing. José Enrique Guerra Salazar Msc. PRESIDENTE DEL TRIBUNAL	09-03-2022
Ing. Pablo Eduardo Lozada Yáñez Msc. DIRECTOR DEL TRABAJO DE TITULACIÓN	09-03-2022
Ing. Jorge Luis Paucar Samaniego Msc. MIEMBRO DE TRIBUNAL	09-03-2022

DEDICATORIA

El presente trabajo de titulación va dedicado con mucho amor a mis padres Luis Pillajo y María Colcha, por ser mi apoyo durante todo este tiempo, por hacer de mí una persona de bien, por brindarme su amor, apoyo y confianza desde el primer momento que ingrese a la ESPOCH, por no dejarme vencer de las adversidades que se me presentaron a lo largo de la carrera, por siempre motivarme a seguir adelante y luchar por mis sueños, a pesar de que hemos pasado por muchos momentos difíciles siempre han estado a mi lado y estaré en deuda eternamente con ellos.

Luis

AGRADECIMIENTO

En primer lugar, agradezco a Dios por acompañarme y guiarme a lo largo de mi carrera profesional, a mis padres por el sacrificio que hacen todos los días por darnos lo mejor a cada uno de sus hijos, por siempre predicar con el ejemplo y enseñarme a ser un hombre de bien.

A ti Paola por ser mi compañera en este viaje, por compartir muchos momentos, por sus consejos, por su felicidad contagiosa y por ser mi fortaleza en momentos difíciles, gracias por su amor incondicional y su paciencia, a su mamá por brindarme su confianza, siempre estaré agradecido con ustedes.

Por último, pero no menos importante a mi entrañable amigo Carlos por ser la mejor persona que he conocido y ser como un hermano más que la vida me dio.

Luis

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE GRÁFICOS.....	xiii
ÍNDICE DE ANEXOS.....	xiv
GLOSARIO.....	xv
RESUMEN.....	xvi
ABSTRACT.....	xvii
INTRODUCCIÓN.....	1

CAPÍTULO I

1.	MARCO TEÓRICO REFERENCIAL.....	5
1.1	Eficiencia energética en el hogar.....	5
1.1.1	<i>Iluminación eficiente en el hogar.....</i>	<i>6</i>
1.1.2	<i>Eficiencia en el control de temperatura.....</i>	<i>6</i>
1.2	Sistemas domóticos.....	7
1.2.1	<i>Tipos de arquitectura.....</i>	<i>7</i>
<i>1.2.1.1</i>	<i>Arquitectura centralizada.....</i>	<i>7</i>
<i>1.2.1.2</i>	<i>Arquitectura distribuida.....</i>	<i>8</i>
1.2.2	<i>Comunicación inalámbrica de sistemas domóticos.....</i>	<i>9</i>
<i>1.2.2.1</i>	<i>Redes inalámbricas personales (WPAN).....</i>	<i>9</i>
<i>1.2.2.2</i>	<i>Redes inalámbricas locales (WLAN).....</i>	<i>10</i>
1.3	Tecnología IoT.....	11
1.3.1	<i>Protocolos de comunicación IoT.....</i>	<i>12</i>
1.3.2	<i>Análisis y elección del servidor IoT.....</i>	<i>13</i>
<i>1.3.2.1</i>	<i>Servidor Ubidots.....</i>	<i>14</i>
1.4	Sensores para domótica.....	14
<i>1.4.1</i>	<i>Sensores de movimiento pasivos.....</i>	<i>14</i>
<i>1.4.2</i>	<i>Sensores de luz.....</i>	<i>15</i>
<i>1.4.3</i>	<i>Sensores de temperatura.....</i>	<i>15</i>
1.5	Tarjetas de desarrollo.....	15

CAPITULO II17

2.	PROPUESTA Y DISEÑO DE PROTOTIPO	17
2.1	Requerimientos para la implementación del prototipo domótico	17
2.2	Concepción general del prototipo domótico	18
2.2.1	<i>Etapa de medición</i>	19
2.2.2	<i>Etapa de envío de datos</i>	19
2.2.3	<i>Etapa de visualización</i>	20
2.2.4	<i>Etapa de activación</i>	20
2.3	Elección de hardware para la implementación del prototipo domótico	20
2.3.1	<i>Selección de las tarjetas de desarrollo</i>	21
2.3.2	Selección de los sensores	23
2.3.2.1	<i>Sensores para el módulo medidor de consumo</i>	23
2.3.2.2	<i>Sensor para el módulo medidor de temperatura</i>	23
2.3.2.3	<i>Sensor para el módulo detector de movimiento</i>	24
2.3.2.4	<i>Elección del sensor para el módulo medidor de luminosidad</i>	25
2.3.3	Selección de los elementos para el módulo de activación	26
2.4	Esquema de conexión del prototipo domótico	26
2.4.1	<i>Esquema del módulo medidor de consumo</i>	27
2.4.2	<i>Esquema del módulo de activación</i>	28
2.4.3	<i>Esquema del módulo de medición de temperatura</i>	28
2.4.4	<i>Esquema del módulo medidor de luminosidad.</i>	29
2.4.5	<i>Esquema del módulo detector de movimiento</i>	30
2.5	Diseño del Software para el prototipo domótico	31
2.5.1	<i>Requerimientos del software del prototipo domótico</i>	31
2.5.2	<i>Diagrama de flujo para la etapa de medición</i>	31
2.5.3	<i>Diagrama de flujo de la etapa de activación</i>	33
2.6	Diseño de carcasa para el prototipo domótico	34
2.7	Instalación del prototipo domótico en el hogar	37

CAPITULO III

3.	VALIDACIÓN DE PROTOTIPO	42
3.1	Comprobación de efectividad de los sensores	42
3.1.1	<i>Validación del sensor de temperatura</i>	42
3.1.2	<i>Comprobación del sensor de movimiento</i>	45
3.1.3	<i>Comprobación del sensor de luminosidad</i>	45

3.2	Estabilidad del prototipo domótico.....	46
3.3	Prueba de transmisión de datos	48
3.4	Prueba de latencia de comunicación.....	49
3.5	Potencia que se consume en el hogar	49
3.6	Comprobacion de la reducción de consumo energetico en el hogar	52
	CAPITULO IV.....	54
4.	EVALUACIÓN ECONÓMICA.....	54
	CONCLUSIONES	56
	RECOMENDACIONES.....	57
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-1:	Valores mínimos de iluminación por estancias.....	6
Tabla 2-1:	Principales servidores IoT	13
Tabla1-2:	Comparativa de tarjetas Arduino con comunicación inalámbrica.....	21
Tabla2-2:	Comparación entre tres tarjetas de desarrollo	22
Tabla3-2:	Comparativa de sensores de temperatura	24
Tabla4-2:	Comparativa de sensores de movimiento PIR	24
Tabla5-2:	Comparación de sensores de luz	25
Tabla6-2:	Elementos para la creación del circuito del módulo de activación remota	26
Tabla 1-3:	Valores de temperatura medidos con el multímetro y el sensor LM35.	44
Tabla 2-3:	Lecturas obtenidas del sensor LDR.....	45
Tabla 3-3:	Resultados de las mediciones de voltaje y corriente	47
Tabla 4-3:	Detalle de los datos procesados	48
Tabla 5-3:	Mediciones de voltaje y corriente que se consumen en el hogar	50
Tabla 6-3:	Consumo de potencia en el hogar.	51
Tabla 7-3:	Consumo eléctrico de los meses previos a la instalación del módulo.	52
Tabla 8-3:	Consumo eléctrico de los meses posteriores a la instalación del módulo.....	52
Tabla 1-4:	Costo general en hardware del módulo domótico implementado.	54
Tabla 2-4:	Costos de comercialización de módulos domóticos similares	55

ÍNDICE DE FIGURAS

Figura 1-1:	Esquema de arquitectura centralizada.....	8
Figura 2-1:	Arquitectura distribuida.	9
Figura1-2:	Concepción general del prototipo domótico.	18
Figura2-2:	Tarjeta ESP01.....	21
Figura3-2:	Tarjeta NodeMCU.	22
Figura4-2:	Arduino nano.	22
Figura5-2:	Sensor voltaje ZMPT101B.....	23
Figura6-2:	Sensor corriente SCT013.	23
Figura7-2:	Sensor LM35.	24
Figura8-2:	Sensor HCSR501.....	25
Figura9-2:	Sensor LDR.....	25
Figura10-2:	Conexión del módulo medidor de consumo.....	27
Figura11-2:	Conexiones del módulo de activación.	28
Figura12-2:	Conexiones del módulo medidor de temperatura.	29
Figura13-2:	Conexiones del módulo medidor de luminosidad.	30
Figura14-2:	Conexiones del módulo detector de movimiento.	31
Figura15-2:	Diagrama de flujo de la etapa de medición.	33
Figura16-2:	Diagrama de flujo de la etapa de activación.....	34
Figura17-2:	Impresión de carcasas con Artillery Genius y PLA.....	34
Figura18-2:	Carcasa para el módulo medidor de voltaje y corriente.	35
Figura19-2:	Carcasa para el dispositivo de medición de luminosidad.....	35
Figura20-2:	Carcasa para el módulo medidor de luminosidad.	36
Figura21-2:	Carcasa para módulos de activación.	36
Figura22-2:	Carcasa para el módulo detector de movimiento.....	37
Figura23-2:	Prototipo domótico completo previo a la instalación.	38
Figura24-2:	Instalación del interruptor IoT.....	38
Figura25-2:	Tomacorriente para control del ventilador.	39
Figura26-2:	Instalación del tomacorriente para controlar el radio.....	39
Figura27-2:	Conexión del medidor de consumo a la red principal.....	40
Figura28-2:	Comprobación de fase.....	40
Figura29-2:	Instalación del detector de movimiento en la bodega.	41
Figura30-2:	Instalación del interruptor IoT en la sala del hogar.	41
Figura 1-3:	Circuito de prueba para el sensor LM35.....	42

Figura 2-3:	Temperatura mostrada en el monitor serial.....	43
Figura 3-3:	Medición de temperatura con multímetro.	43
Figura 4-3:	Valores de voltaje y corriente del hogar medidos.	46
Figura 5-3:	Valores de voltaje y corriente mostrados en la interfaz de Ubidots.....	47
Figura 6-3:	Dirección IP del NodeMCU.....	49
Figura 7-3:	Datos obtenidos en el CMD de Windows.	49
Figura 8-3:	Planillas mensuales año 2021 según CENEL EP.	53

ÍNDICE DE GRÁFICOS

Grafica1-3 :	Grafica obtenida de los datos enviados por el sensor de movimiento.	45
Gráfico 2-3:	Datos de corriente recibidos por Ubidots.	48

ÍNDICE DE ANEXOS

ANEXO A:	HOJA DE DATOS TÉCNICOS DEL SENSOR LM35
ANEXO B:	HOJA DE DATOS TÉCNICOS DEL SENSOR LDR
ANEXO C:	HOJA DE DATOS TÉCNICOS DEL SENSOR HC-SR501
ANEXO D:	HOJA DE DATOS TÉCNICOS DEL SENSOR ZMPT101B
ANEXO E:	HOJA DE DATOS TÉCNICOS DEL SENSOR SCT
ANEXO F:	LÍNEAS DE CÓDIGO DEL NODEMCU
ANEXO G:	LÍNEAS DE CÓDIGO DEL ESP01 INTERRUPTORES
ANEXO H:	LÍNEAS DE CÓDIGO DEL ESP01 MEDIDOR DE CONSUMO
ANEXO I:	LÍNEAS DE CÓDIGO DE ARDUINO NANO

GLOSARIO

ADC: Convertidor Analógico Digital.

APP: Aplicación móvil

IoT: Internet of Things

LED: Diodo Emisor de Luz

mA: Miliamperio (unidad de medida)

MQTT: Message Queuing Telemetry Transport Technical.

mV: Mili voltios (unidad de medida).

KW: Kilo Watio

PIR: Passive Infrared.

LDR: Light Dependent Resistor.

DC: Corriente directa

AC: Corriente alterna

VDC: Voltaje de corriente continua

VAC: Voltaje de corriente alterna

RESUMEN

Este trabajo tuvo como objetivo la creación de un módulo domótico para medir el consumo de energía eléctrica del hogar y reducirlo mediante la aplicación de conceptos, estrategias de eficiencia energética y el uso de la tecnología IoT. El módulo se dividió en etapas, partiendo por la etapa de medición, etapa de envío, etapa de visualización y etapa de activación, se utilizaron sensores para medir voltaje, corriente, temperatura, movimiento y luminosidad, los datos obtenidos se ingresaron en las entradas analógicas y digitales de las tarjetas de desarrollo de la familia Arduino, en la etapa de envío se utilizaron tarjetas con comunicación inalámbrica ESP01 y NodeMCU que se conectan con el servidor Ubidots para almacenar los datos de las mediciones de los sensores, desde aquí se puede interactuar con la información mediante la creación de eventos y tableros de control. Para la etapa de activación, se crearon circuitos que activan de manera remota los dispositivos conectados a la red eléctrica del hogar. Se realizó una prueba de estabilidad, donde se determinó que el coeficiente de variación no excede el 1.293% tal como lo determina la Norma INEN, por lo que se considera que el módulo es estable y de poca varianza. Para comprobar la reducción de consumo se analizaron los valores de las planillas de CELEC EP antes y después de la instalación del módulo, se obtuvo una reducción del 32 % en el valor de la planilla, la instalación del módulo domótico en el hogar permitió reducir el consumo eléctrico y como consecuencia el costo de la planilla a través del control de las actividades de iluminación y ventilación. Se recomienda implementar el uso de comandos de voz como alternativa de activación remota del módulo domótico.

Palabras claves: <MÓDULO DOMÓTICO> <INTERNET DE LAS COSAS (IOT)> <SENSORES> <TARJETA DE DESARROLLO> <SERVIDOR UBIDOTS>.



0498-DBRA-UPT-2022

2022-03-17

ABSTRACT

The objective of this work was the creation of a home automation module to measure the consumption of home electrical energy and reduce it through the application of concepts, strategies of energy efficiency and the use of IoT technology. The module was divided into stages, starting with the measurement stage, sending stage, display stage and activation stage, they were used sensors to measure voltage, current, temperature, motion and light, the data obtained were entered in the analog and digital inputs of the development cards of the Arduino family, ESP01 wireless communication boards were used at the sending stage and NodeMCU that connect to the Ubidots server to store the measurement data of the sensors, from here you can interact with the information by creating events and dashboards. For the activation stage, circuits were created that activate remotely the devices connected to the electrical network of the home. A test of stability was carried out, where it was determined that the coefficient of variation does not exceed 1.293% as determines the INEN Standard, for which the module is stable and with little variance. To verify the reduction in consumption, the values of the CELEC EP spreadsheets were analyzed before and after the installation of the module, a 32% reduction in the value of the spreadsheet, the installation of the home automation module in the home made it possible to reduce electricity consumption and therefore the cost of the payroll through the control of lighting activities and ventilation. It is recommended to implement the use of voice commands as an alternative to remote activation of the home automation module.

Keywords: <HOME AUTOMATION MODULE> <INTERNET OF THINGS (IOT)> <SENSORS> <DEVELOPMENT BOARD> <UBIDOTS SERVER>.

INTRODUCCIÓN

El dispositivo tecnológico que se presenta permite cuantificar el consumo eléctrico en el hogar y reducirlo mediante la aplicación de conceptos y estrategias de eficiencia energética; está conformado por dispositivos con tecnología IoT que controlan y varían el nivel de iluminación dependiendo de los valores medidos por un sensor de luz, además se controla la temperatura de la habitación a través de un ventilador que varía su velocidad para ajustarse a los requerimientos del usuario.

Considerando que el consumo de electricidad dentro de un hogar residencial se encuentra repartido en: iluminación 49%, electrodomésticos 46% y 5% otros equipos como computadoras, dispositivos electrónicos y pequeños electrodomésticos (Ministerio de Electricidad y Energía Renovable 2013), el proyecto se enfoca en el control de iluminación y ventilación de una edificación residencial en la ciudad de Guayaquil; la vivienda consta de una habitación, sala, cocina y bodega, para la automatización de las actividades del hogar se crearon dispositivos que permiten activar los elementos del hogar de manera automática, además se puede medir la temperatura, movimiento y el nivel de luminosidad por medio de sensores. Estas lecturas se envían hacia un servidor que presenta una interfaz amigable, de fácil configuración y manipulación que gobierna a los dispositivos que componen al módulo domótico, desde el servidor se pueden activar o desactivar de manera remota los dispositivos y crear eventos utilizando los datos de los sensores, garantizando que tanto la iluminación y la ventilación solo se activen cuando sea necesario.

Cuando nos referimos a la domótica hablamos de un conjunto de tecnologías que se aplican para controlar y automatizar hogares, donde la reducción de consumo eléctrico ha sido una consecuencia de la aplicación de esta tecnología, pero no se considera en sí para ningún proceso de investigación. Es por esto que se ha desarrollado el término hogar inteligente, donde el hogar adquiere la funcionalidad que la vida moderna requiere, convirtiendo a la domótica en una herramienta que permite la gestión de todas las actividades del hogar y de sus diferentes elementos (Arredondo García 2010).

Considerando que en la ciudad de Guayaquil se desconoce de estudios o proyectos similares al tema de investigación y que los dispositivos que se comercializan a nivel nacional no permiten la interconexión de elementos para monitorizar y controlar el consumo de energía eléctrica en un

hogar, se determinó la necesidad de diseñar un prototipo de módulo domótico enfocado en la eficiencia energética de hogares utilizando sensores y actuadores convencionales controlados de manera inalámbrica por tarjetas de desarrollo.

JUSTIFICACIÓN TEÓRICA

En la Universidad Politécnica Salesiana de la ciudad de Quito se presentó un estudio de “Caracterización de energía eléctrica de clientes residenciales por medio de IoT” donde se plantea un método de medición de consumo eléctrico utilizando elementos de medición no invasivos de bajo costo; el objetivo principal de esta investigación es dar a conocer al usuario el consumo de energía eléctrica del hogar para que pueda tomar las acciones correctivas para reducirlo, ya sea sustituyendo electrodomésticos que sean poco eficientes o re programando el tiempo de uso de ciertos elementos que tienen mayor consumo (Cargua Ramos 2020).

La demanda de energía en el Ecuador se encuentra sectorizado y se divide en: residencial, comercial, industrial y alumbrado público, el sector residencial representa el 35% del consumo total de energía eléctrica en el país (ARCOTEL 2019). Es por esto por lo que este proyecto de investigación se focaliza en implementar tecnologías que permitan monitorizar, controlar y reducir el consumo energético en los hogares residenciales mediante el uso de dispositivos domóticos y la tecnología IoT.

Actualmente existen sistemas domóticos comerciales que tienen la capacidad de conectarse a internet para poder controlarlos de manera remota por medio del celular y el uso de aplicaciones. Existe una gran variedad de elementos que componen estos sistemas: focos, interruptores, tomacorrientes, cámaras e inclusive electrodomésticos, pero el costo que representa adquirirlos es elevado por lo que muchas personas rechazan la idea de adquirirlos para poder automatizar sus hogares (Vargas Rafael 2019).

Conociendo los problemas en cuanto al consumo energético residencial se plantea la implementación de un módulo domótico enfocado en la eficiencia energética que mide el consumo del hogar y controla la activación de iluminación, ventilación y algunos electrodomésticos que son los elementos que tienen mayor consumo dentro del hogar.

JUSTIFICACIÓN APLICATIVA

El proyecto de investigación implementó un módulo domótico enfocado en la eficiencia energética de hogares utilizando sensores y actuadores convencionales controlados de manera inalámbrica por una tarjeta de desarrollo.

El tema se basó en el desarrollo de un módulo domótico compuesto por seis circuitos, el primero se encarga de la medición de voltaje y corriente mediante la utilización de medidores no invasivos que permiten obtener una medida proporcional de los valores de voltaje y corriente que se consumen en el hogar; los datos obtenidos se envían hacia una tarjeta arduino nano que calcula la potencia y factor de potencia mediante el uso de la librería EmonLib. El segundo circuito es un interruptor que está compuesto por una fuente de alimentación que proporciona 3.3V y una corriente de un amperio mismo que posee un opto acoplador que aísla la etapa de potencia del resto del circuito para proteger a los elementos en especial a la tarjeta de desarrollo que permite al circuito conectarse a internet para enviar los datos de las lecturas. Y que posee un TRIAC que actúa como elemento de conmutación que controla la carga que se requiere activar. El tercer circuito es un tomacorriente inteligente basado en tecnología IoT que posee un sensor LM35 para medir la temperatura de la habitación, luego activa el ventilador y varia su velocidad para mantener la temperatura a los niveles requeridos por el usuario. El circuito número cuatro se trata de un tomacorriente IoT que controla la activación del radio; el quinto circuito detecta el movimiento mediante el uso de un sensor HC-SR501 conectado hacia la entrada digital de la tarjeta de desarrollo NodeMCU, que mide la variación infrarroja que producen los objetos cuando están en movimiento. Este tiene una salida digital que envía al servidor 1 cuando detecta y 0 cuando no; el sexto elemento mide el nivel de luminosidad por medio de un sensor LDR y varia la intensidad de la iluminación en la sala.

Se eligió la comunicación inalámbrica wifi para interconectar los circuitos con el servidor, la elección de esta tecnología se hace debido a que se busca reducir en lo posible las afectaciones a la edificación y considerando que no se tiene cableado destinado a la domótica en la vivienda.

Con los datos obtenidos se busca cuantificar el consumo de energía eléctrica del hogar y verificar si efectivamente la instalación del módulo reduce el consumo energético; esto se hará comparando los valores de las planillas de consumo eléctrico otorgados por la empresa eléctrica CELEC-EP antes y después de la instalación.

OBJETIVO GENERAL

Implementar un módulo domótico enfocado a la eficiencia energética de hogares, utilizando sensores y actuadores convencionales controlados de manera inalámbrica por una tarjeta de desarrollo programable.

OBJETIVOS ESPECÍFICOS

- Diseñar un módulo domótico, que pueda ser instalado en hogares que no cuenten con cableado dedicado.
- Adquirir datos de los sensores de manera inalámbrica para luego procesarlos con la ayuda de la tarjeta de desarrollo programable.
- Implementar un módulo que permita controlar el hogar y disminuir el consumo de energía eléctrica.
- Desarrollar un prototipo que pueda ser configurado y manipulado por cualquier usuario.

El presente documento está formado por cuatro capítulos que son los siguientes:

Primer capítulo: presenta el marco teórico donde se detalla toda la sustentación teórica que avala el desarrollo de este proyecto, partiendo por definir conceptos, tecnologías, protocolos y realizar comparativas que sirvieron para la elección de los elementos que conforman el módulo.

Segundo capítulo: define las etapas que componen el módulo, presenta el diseño, muestra la selección de hardware y software y evidencia la instalación del módulo en el hogar.

En el tercer capítulo se realizan las pruebas y se obtienen resultados que demuestran el funcionamiento correcto del módulo.

Por último, en el cuarto capítulo se hace un análisis de costos y se compara con sistemas comerciales que presentan características similares.

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL

En este capítulo se recopila toda la información teórica partiendo por definir conceptos fundamentales sobre tecnologías, arquitecturas y elementos usados que facilitan una mejor comprensión de las características y funcionamiento del módulo domótico al que nos referiremos de aquí en adelante como prototipo domótico para evitar confusión con los demás conceptos.

1.1 Eficiencia energética en el hogar

El aumento de la eficiencia energética significa mejorar la calidad de vida permitiendo al individuo tener iguales o mayores beneficios con menor consumo energético. Algunas medidas de eficiencia energética son conocidas entre nosotros, por ser de “sentido común”, por ejemplo, apagar la luz cuando no se está en la habitación, apagar los equipos que no se están utilizando; otras son alternativas tecnológicas que existen ya hace varios años pero que muchos tienen desconocimiento de su existencia y funcionamiento. Sea cual sea la decisión que se tome, el objetivo principal es reducir el consumo energético (Jiménez Buendía 2009).

Dentro de un hogar o una edificación existen una gran variedad de dispositivos eléctricos que consumen diferentes cantidades de energía eléctrica; el consumo de estos equipos se relaciona directamente a la potencia y su tiempo de utilización. De acuerdo a un estudio realizado por el ministerio de energía y recursos renovables del año 2013, se identificó que el 49% de consumo energético de hogares residenciales en el Ecuador es a causa de la iluminación por esto es importante controlarla para reducir significativamente este consumo (Ministerio de Electricidad y Energía Renovable 2013).

En la actualidad los dispositivos del hogar se pueden controlar de manera automática por medio de módulos de control que permiten activarlos o desactivarlos; estos módulos se caracterizan por tener un reducido consumo energético además se pueden conectar a internet para ser controlados de manera remota desde cualquier lugar. Un hogar eficiente desde el punto de vista domótico controla iluminación y temperatura mediante el uso de sensores para brindar confort y eficiencia energética.

1.1.1 Iluminación eficiente en el hogar

Como se menciona en el apartado anterior la iluminación en el hogar produce el mayor consumo energético, es por esto por lo que gestionar es importante si se desea obtener un consumo eficiente de energía eléctrica; el nivel de iluminación deberá ajustarse a la estancia y la actividad que vamos a desarrollar en ese ambiente. Si la cantidad de luz es insuficiente, el usuario se verá forzado a trabajar su visión en exceso, lo que deriva en fatiga, por el contrario, si la iluminación es excesiva provocara deslumbramiento que es una sensación de molestia que se produce cuando los ojos reciben demasiada de luz (Alvaro Vilca 2011).

Lo que busca la domótica al ser implementada para controlar la iluminación del hogar es proporcionar un ambiente confortable donde los niveles de luz sean los adecuados y donde la intensidad de luz se regule de manera automática.

En la tabla 1-1, se muestran los niveles de iluminación recomendados para el hogar de acuerdo con lo que establece la NEC (*Norma Ecuatoriana de Construcción*), dependiendo de la actividad a desarrollar en cada estancia de la vivienda.

Tabla 1-1: Valores mínimos de iluminación por estancias.

Áreas	Mínimo (LUX)	Recomendado (LUX)	Óptimo (LUX)
Dormitorios	100	150	200
Baños	100	150	200
Cuartos de estar/Sala	200	300	500
Cocinas	100	150	200
Cuarto de estudio o trabajo	300	500	750

Fuente:(Guillén et al. 2017).

Realizado por: Pillajo, Luis, 2022.

1.1.2 Eficiencia en el control de temperatura

Los dispositivos destinados a controlar la climatización deben considerar el costo económico y consumo energético a la hora de ser implementados en un hogar. Si lo que se busca es conseguir eficiencia, el consumo que representa el uso de electrodomésticos destinados a regular la temperatura debe ser mínimo; cuanto mayor sea el consumo de energía para obtener estabilidad térmica en una edificación el sistema será menos eficiente.

El clima depende mucho de la ubicación, topografía y altitud. Además se debe considerar que la temperatura en las ciudades es considerablemente más elevada que en las zonas rurales aproximadamente de 1 a 2 °C; en la ciudad de Guayaquil, la temperatura promedio varía entre 20 y 31 grados, el uso de electrodomésticos destinados a regular esta temperatura representa un

consumo de energía eléctrica considerable, el ventilador se vuelve el electrodoméstico más utilizado para controlar la temperatura debido al costo reducido en comparación al aire acondicionado (Rojas Vega 2015).

1.2 Sistemas domóticos

Se conoce así al conjunto de dispositivos electrónicos centralizados que controlan de manera eficiente todas las actividades permitiendo al usuario una interacción sencilla con el hogar. Para nombrar a un sistema como domótico se debe incorporar al mismo tecnologías innovadoras que permitan al programador generar aplicaciones sencillas y amigables con el usuario, esto quiere decir que está dedicado para usuarios que no tiene conocimiento de programación o de cómo se usa el sistema lo que origina un sistema robusto pero fácil de manipular y configurar que a su vez permite desarrollar actividades de manera automática (Plasencia Alva 2018).

En términos generales se considera que un sistema domótico está formado por una red de comunicación que se encarga de la transmisión de datos provenientes del entorno en el cual está instalado a través de elementos de adquisición de información (sensores); dicha información será ingresada al sistema para ser procesada y transformada en señales de control que permitan realizar determinadas acciones y modificar parámetros del entorno (Sanclemente 2016).

1.2.1 Tipos de arquitectura

Para poder identificar a los sistemas domóticos es necesario conocer qué tipo de arquitectura se está utilizando, esta define la manera en la que se distribuyen los diferentes elementos que lo componen y como se transmite la información. Los tipos de arquitectura básicos son:

1.2.1.1 Arquitectura centralizada

Se denomina así cuando toda la parte lógica del sistema se encuentra alojada en un servidor que va a actuar como maestro y el resto de dispositivos que se conecten a él serán esclavos, esto quiere decir que el programa solo podrá ejecutar órdenes provenientes del servidor y permitirá la comunicación entre placas (Jiménez Buendía 2009). Tal como se muestra en la figura 1-1.

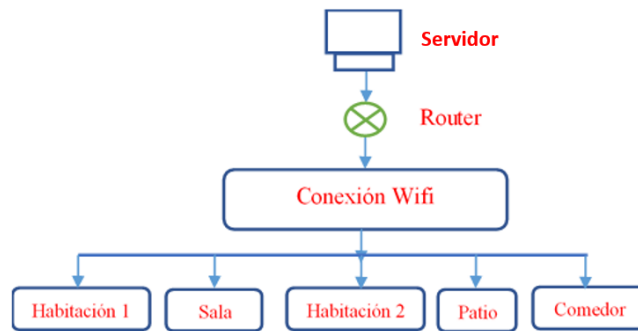


Figura 1-1: Esquema de arquitectura centralizada.

Realizado por: Pillajo, Luis; 2022

En este sistema el mensaje parte del servidor y pasa por el router que distribuye la orden hacia todos los dispositivos que van a ser controlados mediante comunicación alámbrica o inalámbrica, de esta manera el flujo de información va desde el servidor a los clientes y viceversa.

La principal desventaja de este sistema es la dependencia de que el servidor funcione de manera permanentemente, ya que, si existe un problema de comunicación todo el sistema falla debido a que esta arquitectura hace que el flujo de información se sitúe en una sola parte; se recomienda utilizarlo solo en sistemas que no tienen muchos dispositivos conectados y donde no haya mucho flujo de información.

1.2.1.2 Arquitectura distribuida

En lo que respecta a esta configuración cada dispositivo conectado a la red se convierte en servidor y permite el flujo de información aun cuando el servidor principal haya fallado. El servidor será el encargado de difundir hacia los dispositivos conectados todas las instrucciones que hayan sido programadas por el usuario, además identificara y almacenara los valores que resulten de las mediciones como puede ser el caso de cambios en la temperatura, cantidad de luz o la presencia de una persona en una habitación (Lledó Sánchez 2012).

Tal como se aprecia en la figura 2-1, la información viaja por varias rutas, desde el servidor hacia cada una de las unidades de control que se convierten en servidores de respaldo en caso de que existan fallas con el servidor principal esto evita que la comunicación se pierda.

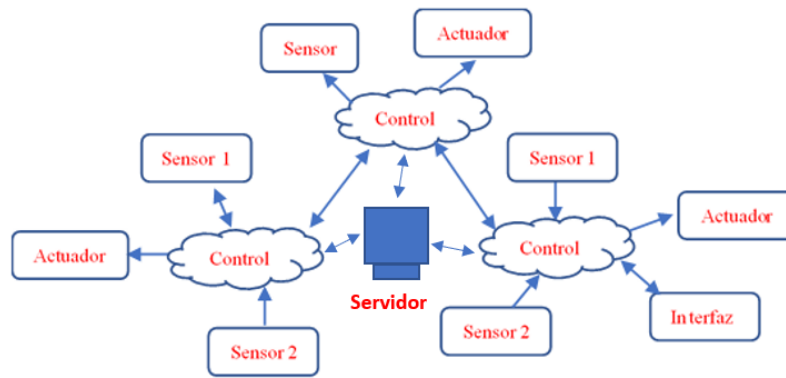


Figura 2-1: Arquitectura distribuida.

Realizado por: Pillajo, Luis; 2022.

El prototipo domótico implementado utiliza una arquitectura centralizada donde el servidor gobierna a todos los dispositivos conectados hacia él, la elección de este tipo de arquitectura se hace debido a la facilidad de implementación y el reducido costo al no requerir servidores de respaldo en caso de fallas del servidor principal.

1.2.2 Comunicación inalámbrica de sistemas domóticos

Este tipo de comunicación entre emisor y receptor se caracteriza por la eliminación del cableado conectando varios nodos o dispositivos sin utilizar una conexión física sino estableciendo la comunicación mediante ondas electromagnéticas (Ruiz Ayala, Vides Herrera y Pardo García 2018).

Como enuncia (Mendoza et al. 2020), una conexión inalámbrica es el enlace que se establece entre instrumentos móviles y fijos que intercambian información utilizando el espectro electromagnético, esta tecnología no es nueva pero actualmente ha tenido un gran auge debido a la facilidad de acceder a un internet de mejor calidad y con mayor velocidad. Los tipos de redes más usados son:

1.2.2.1 Redes inalámbricas personales (WPAN)

Son redes que debido a sus limitaciones de alcance generalmente se usan en hogares de pocas habitaciones y para conectar dispositivos cercanos. Las tecnologías más usadas son: Bluetooth, DECT, NFC y Zigbee.

Bluetooth. - Permite que los dispositivos se comuniquen entre sí sin necesidad de cables. Se basa en la radiofrecuencia de corto alcance y cualquier dispositivo que incorpore la tecnología puede

comunicarse siempre que esté dentro de la distancia requerida, se utiliza a menudo para permitir que dos tipos diferentes de dispositivos se comuniquen entre sí. Dependiendo de cómo esté configurada, la tecnología Bluetooth puede ser bastante segura, pudiendo aprovechar el uso de la autenticación de claves y el cifrado (Roving Networks 2013).

DECT. - (*Digital Enhanced Cordless Telecommunications por sus siglas en inglés*) aparece a partir de los años ochenta para brindar mejoras a las comunicaciones analógicas de la telefonía. Esta tecnología permite obtener los beneficios que otorga una transmisión digital: menos interferencia, más seguridad, mayor capacidad de dispositivos conectados. Su velocidad de transmisión es de hasta 2 Mbps (Beal 2007).

NFC. - (*Near Field Communication por sus siglas en inglés*) es un tipo de comunicación basada en radiofrecuencia que utiliza la banda ISM de 13.56 MHz. Para que la comunicación sea efectiva se requiere que los dispositivos se encuentren cercanos a una distancia mínima de 20 centímetros, actualmente ha tenido un auge respecto al pago electrónico pues muchas aplicaciones y agencias bancarias sobre todo en Europa utilizan este servicio como alternativa de pago (Souza Oliveira y Franco Pereira 2015).

Zigbee.- Es un estándar de comunicaciones inalámbricas regulado por el grupo de trabajo IEEE 802.15.4 en el 2004 que permite habilitar redes inalámbricas con capacidades de control y monitorizar que sean seguras, de bajo consumo energético y de bajo coste de procesador de manera bidireccional (Alliance 2015).

1.2.2.2 Redes inalámbricas locales (WLAN)

Es una red que permite una cobertura geográfica limitada con una velocidad de transmisión de información relativamente alta y bajo porcentaje de errores que usa microondas para la comunicación (Mayné 2005).

El uso de WLAN según (Rackley 2007) permite al usuario tener una red cómoda, fácil de instalar y configurar, todo esto acompañado de una mejora en el ancho de banda, un aumento en la seguridad y fiabilidad de la red siendo una alternativa perfecta frente a las redes de área local o LAN (*Local Area Network por sus siglas en inglés*). Los dispositivos conectados a una WLAN pueden acceder a información y datos al igual que una red cableada sin depender de una conexión física. Poseen ventajas considerables respecto a una red LAN algunas son:

- **Movilidad:** los dispositivos conectados a la red pueden acceder a ella desde cualquier lugar.
- **Instalación simple:** no se requiere instalación física para interconectar a los dispositivos siempre y cuando se encuentren dentro del área de cobertura.
- **Flexibilidad:** llega a todos los lugares donde una instalación física no lo permitiría.
- **Costo reducido:** a pesar de que la inversión inicial es elevada, una vez instalada una red WLAN es más fácil de mantener y aumentar su cobertura.

Las redes de comunicación utilizadas para la implementación de sistemas domóticos permiten la conexión entre dispositivos. La diferencia más significativa es la distancia y la cantidad de usuarios que pueden conectarse a dicha red, para la implementación del prototipo domótico se utilizó una red comunicación inalámbrica del tipo WLAN que conecta el servidor alojado en internet con todos los dispositivos que componen el prototipo domótico, de esta manera la información se envía a través del espectro electromagnético y se reduce la necesidad de cableado físico.

1.3 Tecnología IoT

También llamado "Internet de las cosas", es una tecnología que se encuentra prácticamente disponible en la mayoría de dispositivos desde televisores inteligentes, teléfonos y computadores hasta elementos del hogar como cafeteras, cocinas y cualquier equipo que se pueda conectar a internet para ser activado de manera remota (Bohórquez Heras 2020).

Dentro de la domótica el IoT (*Internet of things, por sus siglas en inglés*) no es más que la interconexión de dispositivos eléctricos y electrónicos a internet con el objetivo de controlarlos, gestionarlos y monitorearlos, por medio de la utilización de sensores. Luego la información se transmite hacia el elemento de control quien analiza y ejecuta las ordenes que serán enviadas a los actuadores, esta es la mejor forma de automatizar actividades o procesos cotidianos y son la base para el diseño e implementación de hogares inteligentes cuyo enfoque es brindar a los usuario una experiencia de confort, seguridad y eficiencia energética (Van Kranenburg y Bassi 2012).

Los dispositivos IoT utilizan bases de datos y servidores que les permiten comunicarse para enviar y recibir información utilizando como medio el internet; esta información llega hacia la unidad de control, quien se encarga de programar acciones específicas o definir sus estados.

Características de servidores IoT (Somantri, Yustiana y Nugraha 2020).

- Conectividad.
- Gestión de dispositivos conectados.
- Base de datos.
- Gestión y administración de datos.
- Representación de datos por medio del uso de aplicaciones o páginas web.
- Permite analizar los datos obtenidos del entorno.

1.3.1 *Protocolos de comunicación IoT*

Dentro de la tecnología IoT existen protocolos que se usan para transmitir información de sensores hacia internet y son:

MQTT. - (*Message Queuing Telemetry Transport por sus siglas en ingles*), que es un protocolo de comunicación de mensajes sencillo y ligero que permite el envío de información entre dispositivos conectados a internet; este se volvió popular debido a que los dispositivos IoT tienen limitaciones de potencia, consumo y ancho de banda. Está basado en el método de publicadores y suscriptores corriendo sobre el protocolo TCP/IP. Requiere pocos recursos a nivel de procesamiento por parte del controlador y un reducido ancho de banda (Hunkeler, Truong y Stanford-Clark 2008).

COAP. - (*Constrained Application Protocol por sus siglas en ingles*), es un protocolo diseñado para facilitar la comunicación máquina a máquina. Está diseñado para sensores de baja potencia y tiene como objetivo transmitir el modelo HTTP pero incluye otros requisitos como multidifusión y baja sobrecarga y simplicidad que son muy importantes para la transmisión de información hacia los servidores IoT (Suquinagua Otavalo y Muñoz Abad 2018).

De los protocolos analizados se elige MQTT para gobernar a los dispositivos IoT porque permite reducir el consumo de potencia y ancho de banda considerando que el módulo está compuesto por varios elementos que estarán conectados permanentemente al fluido eléctrico y a la red de internet del hogar. Con esto se busca que el consumo eléctrico del prototipo domótico sea bajo y que la velocidad de internet no se vea afectada.

1.3.2 Análisis y elección del servidor IoT

En términos generales lo que se necesita para la implementación de un sistema domótico es un conjunto de dispositivos que se conectan entre sí con un servidor IoT. Este servidor permite una conexión estable que tiene la capacidad de visualizar e interactuar con los datos de los sensores enviados hacia internet. Actualmente existen varios servidores IoT de código abierto; se denominan así debido a la facilidad que tiene el programador para escribir, modificar y distribuir el código.

La tabla 2-1, muestra los principales servidores IoT y sus características generales.

Tabla 2-1: Principales servidores IoT

Plataforma	HW, compatible	Ámbito de uso	Ventajas	Desventajas
Thingspeak	Arduino, Spark Raspberry Pi	Smart Home Prototipos	Interfaz. Aplicación	Documentación limitada a cierto HW.
Carriots	Arduino, Raspberry	Agrícola Edificios, consumidor Logística	HW compatible Ámbitos de aplicación.	Poca Documentación
Spark	Spark	Prototipos, Smart Home, Logística	Ideal para enseñar. Escalable.	Compatibilidad HW.
Ubidots	Arduino, raspberry	Smart home, prototipos Empresas Proyectos	Interfaz amigable Aplicación Creación de eventos Uso de redes sociales	Documentación limitada a cierto tipo de HW.

Realizado por: Pillajo, Luis, 2022.

El servidor Ubidots es quien brinda las mayores prestaciones, que son útiles para los requerimientos del proyecto siendo una de las funcionalidades más útiles de esta plataforma la posibilidad de interactuar con los datos mediante la creación de eventos. De esta manera se puede automatizar la activación de los dispositivos instalados en el hogar sin tener que modificar el código y volver a programar las tarjetas, por ejemplo: cuando el usuario desea modificar el valor de temperatura al que desea que se active el ventilador, basta con modificar el valor del punto de consigna en el evento creado.

1.3.2.1 Servidor Ubidots

Permite el manejo de datos enviados desde sensores mediante la creación de tableros donde se pueden visualizar la información por medio de widgets, así también se crean objetos para interactuar como interruptores, botones, indicadores entre otros. Se puede utilizar también como una herramienta analítica para desarrollar cálculos con los datos obtenidos y mostrarlos en gráficos en tiempo real (Espinosa Apolo y Orellana Guayas 2021).

Entre los principales términos utilizados en Ubidots para manejar datos almacenados en sus servidores, podemos mencionar: (Mohammed, Noor Saleh; Selman 2021).

Data Source: Hace referencia a un dispositivo en el que se crean una o varias variables.

Variable: Conjuntos de datos que pueden cambiar dependiendo de las mediciones de los sensores o de la información que se envía a la nube.

Value: Es el valor numérico que se muestra en tiempo real.

Event: Es un evento o una acción que se puede realizar dependiendo de los datos medidos por sensores, por ejemplo, con los datos de temperatura de un sensor se puede crear un evento que active un dispositivo de manera remota cuando el valor de temperatura exceda al valor configurado.

1.4 Sensores para domótica

Son importantes elementos de los sistemas domóticos. Equivalente a nuestros sentidos, se encargan de detectar y medir magnitudes físicas del entorno como la temperatura, iluminación y presencia. Estos datos se envían al elemento de control donde se procesa la información para luego enviar ordenes hacia los actuadores (Vetelino y Reghu 2017). A continuación, se presentan los tipos de sensores que se usan en proyectos domóticos.

1.4.1 Sensores de movimiento pasivos

También denominados PIR (*Passive Infrared Sensor por sus siglas en ingles*), son los más comunes y usados en aplicaciones domóticas debido a que no necesitan emitir una señal de manera continua para detectar movimiento; esto se traduce en que tienen un consumo más reducido de potencia. Este tipo de sensores se activan cuando miden una variación en la radiación infrarroja que emiten los objetos, su componente principal es un dispositivo denominado piro eléctrico diseñado para detectar cambios en la radiación infrarroja recibida por los objetos cuando

están en movimiento y a su vez emite una señal eléctrica que luego es amplificada para enviar a su salida un pulso por un tiempo determinado (Andrés y Godoy 2017).

1.4.2 *Sensores de luz*

Este tipo de sensores reciben alimentación desde un dispositivo de control electrónico. Responden a la variación de la intensidad de luz y entrega un impulso eléctrico de intensidad proporcional a la luz medida por medio de una célula fotoeléctrica. Se usan para determinar la iluminancia que es la cantidad de flujo luminoso emitida por una fuente de luz. En domótica se usa para medir el nivel de iluminación en una estancia siendo su unidad de medida el lux (Pallás 2003).

1.4.3 *Sensores de temperatura*

Entre los dispositivos de domótica más comunes que podemos instalar en una vivienda inteligente se encuentran los sensores de temperatura que son capaces de detectar magnitudes físicas o químicas llamadas variables de instrumentación como la temperatura. Para realizar una medición de temperatura, el sensor la convierte en una señal eléctrica de voltaje que se entrega a su salida y es proporcional al valor de temperatura que se está midiendo; estos datos se pueden ingresar a un microprocesador o tarjeta de desarrollo para determinar el valor en grados centígrados o grados Fahrenheit (Rodríguez, Ocampo y Ortega 2007).

1.5 Tarjetas de desarrollo

Dentro del campo de la domótica existen varias familias de tarjetas de desarrollo con diferentes características, considerando que para la creación del prototipo domótico se requiere que posean compatibilidad con el servidor seleccionado en este caso Ubidots. Se analizan dos familias importantes que cumplen con los requerimientos y son las siguientes:

Arduino. - Existen muchas empresas que fabrican hardware libre una de las más representativas y conocidas es sin dudar la empresa italiana Arduino. Las tarjetas Arduino facilitan la conexión e implementación de proyectos para satisfacer las necesidades de la población. La necesidad de almacenar la información histórica de datos medidos o calculados es un aspecto relevante en especial en la creación de prototipos IoT donde se busca que los dispositivos puedan conectarse a internet para enviar y recibir información (Enríquez Herrador 2009).

Dentro de las características más representativas de esta familia se tiene:

- Al tener una arquitectura simple su costo también se reduce considerablemente en relación con otras tarjetas.
- Gran variedad de modelos que permiten implementar diferentes tipos de proyectos con requerimientos específicos.
- La facilidad de programación usando una variante del lenguaje C y el uso de software libre fácil de comprender y ejecutar.

Raspberry. - Es una computadora muy pequeña similar a una tarjeta de crédito compuesta por varios elementos. Se aloja en una placa sobre la que se monta un procesador, memoria RAM y un chip gráfico. Estos paneles son muy versátiles. Se pueden utilizar para realizar una gran variedad de proyectos que van desde sistemas de video vigilancia, creación de centros multimedia, conmutadores telefónicos, consolas de videojuegos, sustitución de ordenadores de sobremesa, o como controladores domóticos (Cardona et al. 2019).

Algunas de las características de este tipo de tarjetas son:

- Amplia memoria RAM con capacidad de ejecutar varias tareas de manera simultánea.
- Posee la capacidad de poder instalar un sistema operativo y conectar periféricos para poder convertirlo en un ordenador portable.
- Utiliza el lenguaje de programación Python que soporta parcialmente la orientación a objetos.
- Posee una gran variedad de puertos de entrada/salida.

De las familias de tarjetas antes mencionadas y tomando en cuenta las necesidades y requerimientos generales del proyecto, se eligen las tarjetas de la familia Arduino por su compatibilidad con el servidor, su costo reducido y la facilidad de programar e implementar.

CAPITULO II

2. PROPUESTA Y DISEÑO DE PROTOTIPO

En el presente capítulo hablaremos del prototipo domótico que se implementó para este proyecto de investigación. A continuación, se presenta el proceso de diseño partiendo por la identificación de las etapas que lo componen, diagramas de los circuitos implementados, características del *hardware* y *software* seleccionado; además del funcionamiento del prototipo al controlar iluminación, ventilación y activación de algunos electrodomésticos con la finalidad de reducir el consumo energético del hogar.

2.1 Requerimientos para la implementación del prototipo domótico

Los requerimientos con los que el prototipo debe cumplir se establecen considerando que el hogar se encuentra habitado por dos usuarios. Se localiza en la ciudad de Guayaquil con una temperatura promedio de 21-30 °C. No se dispone de cableado destinado a domótica y no existe ningún dispositivo IoT. Tomando en cuenta estos aspectos el prototipo domótico debe cumplir con los siguientes requerimientos:

- Automatizar la activación de la iluminación en cada una de las estancias utilizando dispositivos con tecnología IoT.
- Medir el nivel de luminosidad y variar la intensidad de la iluminación para que se ajusten a los valores recomendados por la NEC.
- Medir la temperatura de la habitación utilizando un sensor y variar la velocidad del ventilador para mantener el nivel de temperatura configurado por el usuario.
- Detectar movimiento mediante un sensor y activar de manera automática la iluminación.
- Almacenar y presentar en Ubidots los datos obtenidos de los sensores e interactuar con ellos.
- Utilizar tarjetas de desarrollo de la familia Arduino para conectar y controlar el prototipo domótico.

- El prototipo domótico debe conectarse hacia el servidor de manera inalámbrica utilizando la red wifi del hogar.
- Usar la aplicación propia de Ubidots como alternativa para visualizar datos y activar de manera remota todos los módulos que componen el prototipo domótico utilizando para ello un teléfono móvil.

2.2 Concepción general del prototipo domótico

Para tener una idea clara del funcionamiento del prototipo implementado, se presenta el diagrama de la figura 1-2. Para la transmisión de información entre los distintos módulos que integra el prototipo domótico, se utilizó comunicación inalámbrica wifi que funciona bajo el protocolo IEEE 802.15.4. El prototipo propuesto tiende a ser escalable pudiendo ser replicado e implementado en varios hogares con el objetivo de realizar estudios de consumo eléctrico de hogares en un barrio, urbanización o sector.

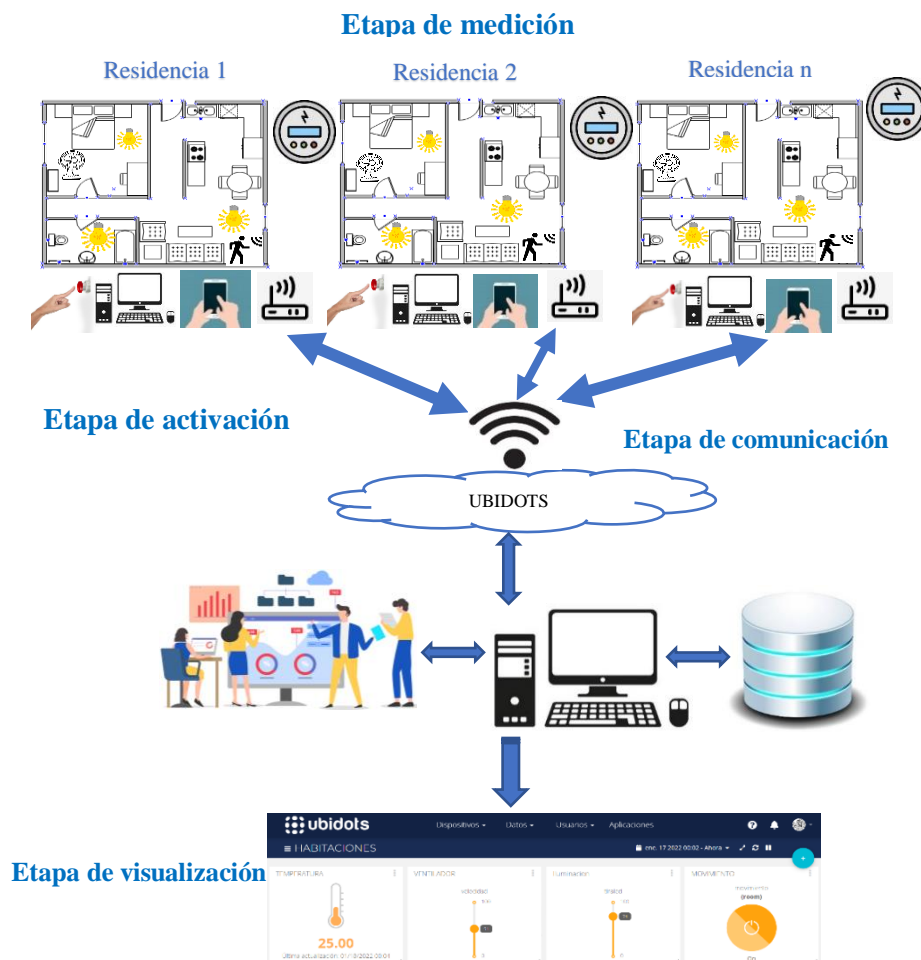


Figura1-2: Concepción general del prototipo domótico.
Realizado por: Pillajo, Luis; 2022.

2.2.1 Etapa de medición

El prototipo presenta una arquitectura centralizada donde Ubidots es el servidor que controla todos los dispositivos del hogar. Se comienza por la etapa de medición donde se cuantifica voltaje, corriente, temperatura, nivel de luminosidad y se detecta movimiento a través de sensores. En esta etapa se crearon varios módulos que se repartieron por las diferentes estancias del hogar; los datos medidos se envían hacia el servidor alojado en internet mediante una conexión wifi.

Medición de consumo. - Se realiza la medición de voltaje y corriente utilizando sensores no invasivos instalados en la línea principal del hogar. Estos permiten caracterizar la señal de voltaje y corriente que se consume en el hogar. Los datos que se almacenan en el servidor se usan para monitorear el consumo energético del hogar y aplicar técnicas enfocadas en reducir el consumo con el objetivo final de disminuir el costo de la planilla eléctrica.

Medición de temperatura. - Para la medición de temperatura del lugar se utiliza un sensor que proporciona un valor de voltaje a la salida que es proporcional al nivel de temperatura medido. Considerando que la temperatura en la ciudad de Guayaquil es elevada, se crearon dispositivos IoT que varían la velocidad del ventilador para mantener los niveles de temperatura de acuerdo con los requerimientos del usuario.

Detección de movimiento. - En la estancia denominada bodega con un área de dos metros cuadrados, se quiere que la iluminación se active de manera automática cuando detecte la presencia del usuario. De esta manera se garantiza un ahorro de energía al activar la iluminación solo cuando sea requerida.

Medición del nivel de luminosidad. - Para la estancia de sala se requiere medir la cantidad de luz natural que ingresa por la ventana. Para esto se mide el nivel de iluminación con ayuda de un sensor; los datos se envían al servidor y se crea un evento que varía la intensidad de la luz para garantizar que el nivel de iluminación sea de 300 luxes, de acuerdo al nivel recomendado por la NEC descrito en el capítulo de fundamentación teórica.

2.2.2 Etapa de envío de datos

Considerando que uno de los objetivos del módulo es reducir al mínimo las afectaciones a la edificación, se eligió la comunicación inalámbrica mediante wifi para conectar el módulo con Ubidots. Para esto, se eligieron las tarjetas de la familia Arduino porque son compatibles con el

servidor. Estas tarjetas se conectan con la red de internet de uso doméstico mediante el uso de credenciales como: el nombre de la red, la contraseña y el TOKEN. Para esta etapa los módulos se distribuyeron por el hogar buscando la ubicación donde tengan mejor cobertura de internet.

2.2.3 Etapa de visualización

Los datos de las mediciones de los sensores se muestran en la interfaz de Ubidots por medio de la creación de tableros donde se visualiza la información de voltaje, corriente, temperatura, nivel de luz y la detección de movimiento.

Para representar a los dispositivos IoT, se crearon botones que cambian de color dependiendo del estado. Esto ayuda a los usuarios para identificar en cualquier momento si existen dispositivos activados debido a que olvidaron desactivarlos o como consecuencia de algún problema en el hogar. Esto permitirá tomar las acciones correctivas necesarias.

2.2.4 Etapa de activación

Se crearon circuitos de activación remota que actúan como interruptores que controlan la iluminación y tomacorrientes que activan los electrodomésticos. Cada módulo creado cuenta con un pulsador para activarlo de manera manual; esto se hizo para tener una alternativa de activación y para no depender únicamente de un dispositivo móvil o un computador para la activación del módulo.

Una alternativa para la activación de los módulos que componen al prototipo domótico es la aplicación de Ubidots que a diferencia de la interfaz que se ejecuta en un computador tiene limitaciones respecto a la creación de eventos.

2.3 Elección de hardware para la implementación del prototipo domótico

A continuación, se detallan las características de los elementos que se usaron para implementar el prototipo domótico, el cual está compuesto por seis módulos, además se presenta una breve reseña del porqué de su selección.

2.3.1 Selección de las tarjetas de desarrollo

Para elegir la tarjeta de desarrollo adecuada debemos considerar que el proyecto requiere conectarse al servidor de manera inalámbrica usando wifi. Al ser un módulo destinado a reducir el consumo energético, el consumo de las tarjetas de desarrollo elegidas debe ser bajo.

Tomando en cuenta los requerimientos mencionados anteriormente, se han identificado 4 tarjetas que cumplen con los requisitos buscados y se detallan en la tabla 1-2.

Tabla1-2: Comparativa de tarjetas Arduino con comunicación inalámbrica

Características	NodeMCU	Arduino Yun	ESP01	Arduino UNO wifi
Voltaje de alimentación.	5V	5V	3.3V	5V
Tecnología Wi-Fi.	IEEE 802.11b/g/n	IEEE 802.11b/g/n	IEEE802.11b/g/n	IEEE802.11b/g/n
Dimensiones	48x26x13mm	68.6 x 53.3 mm	24x14mm	68.6 x 53.3 mm
Entrada analógica.	SI	SI	NO	SI
Entradas digitales.	SI	SI	SI	SI
Consumo de corriente.	60mA	50mA – 120mA	40mA	50mA
Costo.	7\$	40\$	5\$	20.30\$

Realizado por: Pillajo, Luis; 2022.

Luego de analizar la tabla 2-2 para la creación de los dispositivos de activación, se elige la tarjeta ESP01 de la figura 2-2, por su reducido consumo de corriente. Además, presentan una mejor relación costo beneficio es decir presentan las mismas características que otras tarjetas, pero con un costo más bajo.

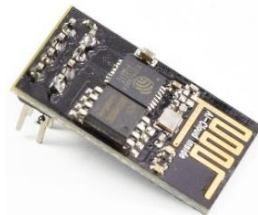


Figura2-2: Tarjeta ESP01.

Realizado por: Pillajo, Luis; 2022.

Para la creación de los módulos que miden temperatura, movimiento y nivel de luz, se elige la tarjeta NodeMCU de la figura 3-2, debido a que se requiere entradas analógicas para recibir los datos de los sensores, además presenta un costo y tamaño reducidos.



Figura3-2: Tarjeta NodeMCU.
Realizado por: Pillajo, Luis; 2022.

Para el módulo medidor de consumo se requiere de una tarjeta de tamaño reducido y con un bajo nivel de consumo de corriente. Considerando estos requerimientos se tienen como candidatos las tarjetas de la tabla 2-2.

Tabla2-2: Comparación entre tres tarjetas de desarrollo

Característica.	Arduino Mega.	Arduino nano	Arduino Uno
Microcontrolador.	ATmega2560.	ATmega328	ATmega328
RAM.	256KB	32 KB	32KB
Voltaje de Alimentación.	7 – 12V	5V	7-12V
GPIO.	54 pines digitales 16 pines analógicos	40 pines	14 pines digitales 6 pines PWM 6 pines analógicos
Frecuencia de reloj.	16 MHz	16 MHz	16 MHz
Corriente que consume.	90mA	40mA	50mA
Entrada de USB	Regular	Regular	Mini

Fuente: (McRoberts Michael 2010).
Realizado por: Pillajo, Luis; 2022.

Analizando la tabla 2-2, para el módulo de medición de consumo se utilizó el Arduino nano de la figura 4-2, puesto que presenta el tamaño y costo reducido respecto a las otras tarjetas; además de bajo consumo al ser similar al Arduino uno, pero de tamaño reducido es perfecto para desarrollar una PCB compacta.



Figura4-2: Arduino nano.
Realizado por: Pillajo, Luis; 2022.

2.3.2 Selección de los sensores

Para la creación de los módulos que componen al prototipo domótico se eligieron varios sensores a continuación se detallan las características de cada uno.

2.3.2.1 Sensores para el módulo medidor de consumo

Como sensor de voltaje se elige el ZMPT101B de la figura 5-2 cuyas características se encuentran detalladas en el ANEXO D; permite la medición de voltajes bajos y proporciona a su salida una señal analógica proporcional a la señal medida. Este tipo de dispositivos es ideal para aplicaciones de monitoreo de energía y es muy común en aplicaciones de domótica.



Figura5-2: Sensor voltaje ZMPT101B.

Realizado por: Pillajo, Luis; 2022.

Para medir la corriente se utilizó el sensor SCT013 de la figura 6-2. Sus características se detallan en el ANEXO E. Funciona como un transformador en donde la corriente que circula por el cable que deseamos medir actúa sobre el devanado primario (1 espira) e internamente tiene un devanado secundario que dependiendo del modelo puede tener más de 2000 espiras. Este sensor se debe ubicar rodeando solo la línea de fase caso contrario la medición será 0 puesto que los cables tienen corrientes opuestas y se anulan.



Figura6-2: Sensor corriente SCT013.

Realizado por: Pillajo, Luis; 2022.

2.3.2.2 Sensor para el módulo medidor de temperatura

Dentro de los elementos más usados en la creación de dispositivos electrónicos destinados a medir temperatura en el hogar, destacan los sensores de la tabla 3-2.

Tabla3-2: Comparativa de sensores de temperatura

Características	DTH11	DTH 22	DS18B20	LM35
Voltaje de alimentación	3-5V	3-6V	3-5.5V	4-30V
Corriente de operación	2mA	2.2mA	1.3mA	60µA
Rango de medición	0-50 °C	-40-80 °C	-55-125°C	-55-150 °C
Costo	2.25\$	7\$	4\$	2\$

Realizado por: Pillajo, Luis, 2022.

Se elige el sensor LM35 de la figura 7-2, considerando que es un sensor analógico fácil de implementar. Para que la tarjeta Arduino pueda determinar el valor de temperatura no requiere del uso de librerías basta con usar el valor en mili voltios que ingresa por la entrada analógica y convertirlo a grados centígrados mediante operaciones matemáticas sencillas, esto sumado a un reducido consumo de corriente y a un rango de medición más amplio lo hacen idóneo para la medición de temperatura en el módulo.



Figura7-2: Sensor LM35.

Realizado por: Pillajo, Luis; 2022.

2.3.2.3 Sensor para el módulo detector de movimiento

Como se revisó en el capítulo anterior los sensores de movimiento PIR son los más usados en domótica De acuerdo con la investigación realizada se determinó que existen tres tipos, en la tabla 4-2, se presenta una comparativa de las características de cada uno.

Tabla4-2: Comparativa de sensores de movimiento PIR

Características	HC-SR501	HC-SR04	AM-312
Voltaje de alimentación	4.5-5V	5V	9V
Corriente de operación	10mA	15mA	20mA
Distancia de detección	3-7m	2-45 cm	1m
Costo	2.5\$	3.5\$	3\$

Realizado por: Pillajo, Luis, 2022.

De la tabla 4-2, se elige el sensor HC-SR501 de la figura 8-2 considerando que tiene un rango de detección más amplio que permite cubrir toda el área de la bodega, además presenta un reducido costo y consumo de corriente.



Figura8-2: Sensor HCSR501.

Realizado por: Pillajo, Luis; 2022.

2.3.2.4 Elección del sensor para el módulo medidor de luminosidad

De acuerdo con la investigación realizada para aplicaciones domóticas, existen tres tipos de sensores de luz. L la tabla 5-2 detalla las características de cada uno.

Tabla5-2: Comparación de sensores de luz

Características	BH-1750	LDR	LM393
Voltaje de alimentación	3.3V	5-150V	5V
Consumo de corriente	15mA	10mA	12mA
Tipo de salida	digital	analógica	digital
Dimensiones	18X14mm	10X10mm	30X16mm
Costo	2.5\$	1\$	3\$

Realizado por: Pillajo, Luis, 2022.

Se elige el sensor LDR de la figura 9-2 por su reducido costo, tamaño y consumo de corriente. Además, debido a su facilidad de instalación no requiere calibración. Para la medición del nivel de luz basta con medir la variación de la resistencia; la señal de salida se envía a entrada analógica del NodeMCU y de ahí al servidor. De esta manera se puede variar la intensidad de la luz, mediante la creación de eventos con el objetivo para aprovechar de mejor manera la iluminación natural.



Figura9-2: Sensor LDR.

Realizado por: Pillajo, Luis; 2022.

El sensor LDR proporciona un valor de resistencia a su salida; cuanto más luz detecte la resistencia disminuye, mientras que al detectar menos luz su resistencia aumenta. Para convertir el valor de resistencia a luxes se utiliza la siguiente fórmula:

$$L = \frac{120 \times 10^3}{R} \quad (1)$$

Donde:

R: es el valor de resistencia que entrega el sensor.

L: es el nivel de luminosidad en luxes.

2.3.3 Selección de los elementos para el módulo de activación

En la creación de estos circuitos se utilizaron los elementos que se detallan en la tabla 6-2.

Tabla6-2: Elementos para la creación del circuito del módulo de activación remota

Elemento electrónico.	Voltaje de alimentación.	Consumo de corriente.
Módulo ESP01	3.3 V	60 mA
Fuente de alimentación 110/3.3V	110 V	80 mA
Capacitor 100nF/400V	-	-
MOC3021	3-5 V	30mA
Resistencias	-	-
Consumo total		170mA

Realizado por: Pillajo, Luis; 2022.

Para la creación de estos dispositivos se parte por la fuente de alimentación que convierte el voltaje de entrada 110-220V AC a 3.3 V DC para energizar la tarjeta de desarrollo y el resto de los elementos del circuito. Posee un TRIAC BTA16 que es un transistor activado por voltaje que funciona como interruptor para activar o desactivar la carga, incorpora un MOC3021 que aísla las etapas de alta y baja tensión. Se usa como medio de protección para la placa de desarrollo. Por último posee algunos elementos pasivos, resistencias y capacitores. Los tomacorrientes e interruptores permiten controlar cualquier carga que tenga un consumo menor a 16 A.

2.4 Esquema de conexión del prototipo domótico

Una vez seleccionados todos los elementos utilizados para la elaboración de los dispositivos que componen el prototipo domótico, se utilizó *Easy EDA*, para simular los circuitos y diseñar las placas de circuito impreso.

2.4.1 Esquema del módulo medidor de consumo

El esquema que se aprecia en la figura 10-2 utiliza dos tarjetas de desarrollo para calcular el valor de voltaje y corriente, luego se envían los datos al servidor a través de comunicación inalámbrica:

- El Arduino nano a través de la librería EmonLib, determina los valores efectivos de voltaje y corriente que el hogar está consumiendo.
- La señal del SCT-013 se conecta al terminal de entrada analógica A1 del Arduino nano.
- Debido a que el SCT-013 entrega a su salida una señal de corriente se requiere convertir a voltaje por lo que se conecta a una resistencia de burden y a un filtro que elimina las partes negativas de la señal que podrían causar afectaciones a la tarjeta de desarrollo.
- La señal de voltaje que entrega el sensor ZMPT101B ingresa por A0.
- Los datos obtenidos del Arduino nano se envían hacia Ubidots que está alojado en internet. Mediante comunicación serial se envía la información hacia una tarjeta ESP01 permitiendo a la placa conectarse a la red doméstica para enviar los datos hacia el servidor.
- Se establece conexión entre el servidor y la placa de medición mediante el SSID, contraseña y el TOKEN.
- Considerando que la tarjeta ESP01 requiere 3.3V de alimentación, se conecta un regulador de voltaje

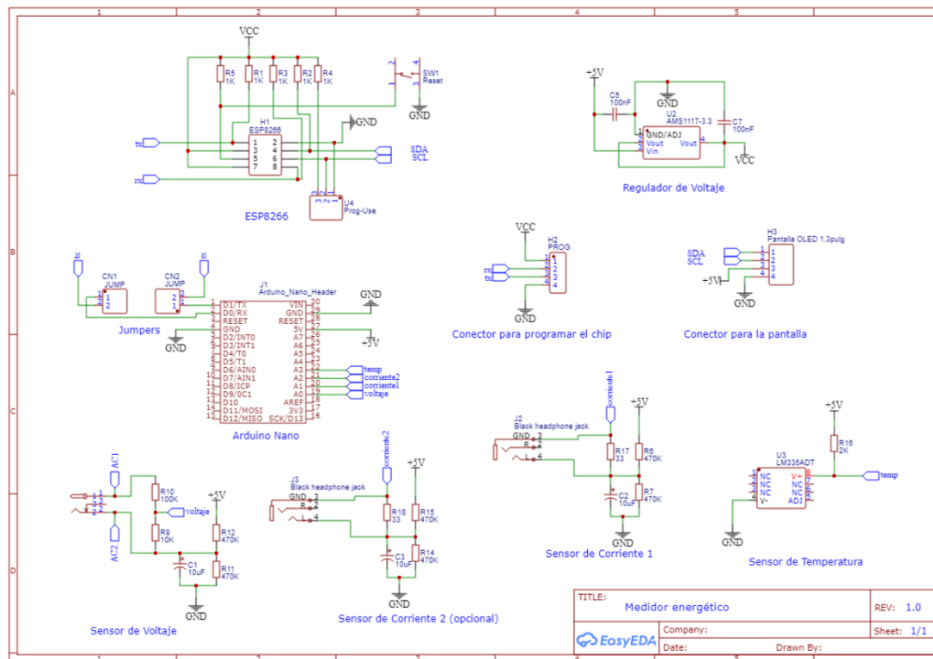


Figura10-2: Conexión del módulo medidor de consumo.

Realizado por: Pillajo, Luis; 2022.

2.4.2 Esquema del módulo de activación

El circuito de la figura 11-2, tiene como elemento central la tarjeta ESP01 que permite enviar y recibir datos del servidor a través de una conexión inalámbrica. Este circuito funciona así:

- Se parte por la alimentación del circuito que se hace mediante una fuente que convierte de 110V AC a 3.3V DC, considerando que los elementos electrónicos sobre todo la tarjeta ESP01 soporta únicamente este voltaje, caso contrario ocasionaría daños.
- Se conecta la tarjeta en modo programación; esto se logra conectando resistencias de 1KΩ al terminal GPI 0 y reset, el terminal de salida será el GPIO 2.
- Para aislar las etapas de baja y alta tensión se utiliza un opto acoplador MOC3010 que funciona como interruptor activado por luz; a su vez para activar la carga de alta tensión se usa el TRIAC BTA16 junto con elementos pasivos como capacitor y resistencias. De esta manera se obtiene un circuito de activación remota que al no tener elementos mecánicos que accionen la carga, reduce el desgaste y aumenta la vida útil del interruptor wifi.

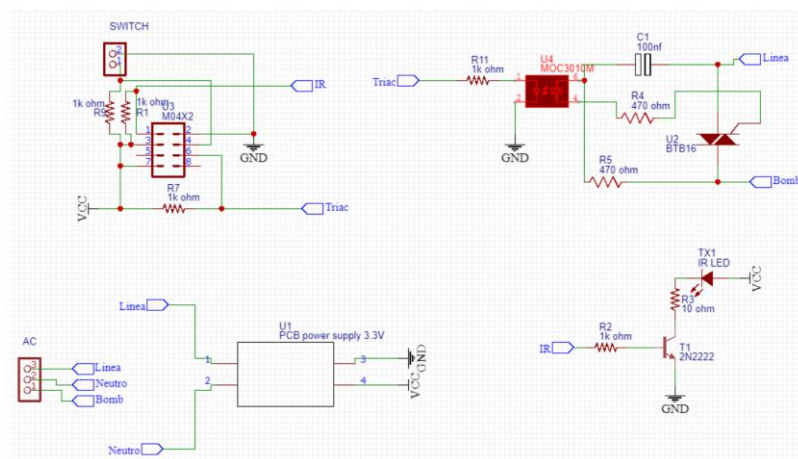


Figura11-2: Conexiones del módulo de activación.

Realizado por: Pillajo, Luis; 2022.

2.4.3 Esquema del módulo de medición de temperatura

Como tarjeta de control del circuito de la figura 12-2, se tiene el NodeMCU que tiene entradas analógicas para recibir los datos del sensor de temperatura. El funcionamiento del circuito se detalla a continuación:

- Se utiliza un regulador de voltaje LM7805 y se reduce a 5V para que pueda funcionar la tarjeta de desarrollo y los elementos del circuito.

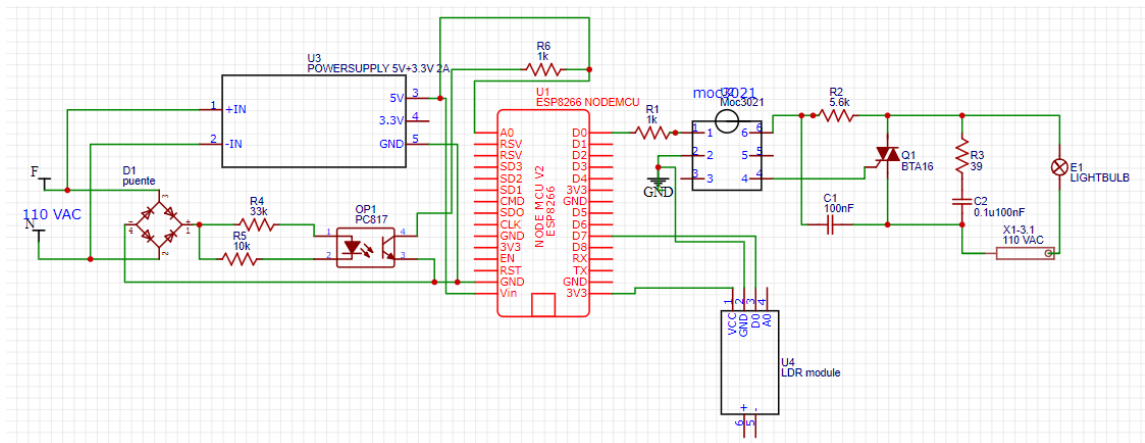


Figura13-2: Conexiones del módulo medidor de luminosidad.

Realizado por: Pillajo, Luis; 2022.

2.4.5 Esquema del módulo detector de movimiento

Para la bodega se requiere que la luz se active de manera automática cuando alguien ingrese, esta acción se realizara de manera remota, usando el circuito de la figura14-2, cuando el servidor envié la señal de activar la luz por medio de comunicación inalámbrica.

- Se utiliza una tarjeta NodeMCU para recibir los datos por medio del terminal D5 donde se conecta el sensor de movimiento PIR HCSR501.
- Energizamos el circuito mediante el uso de un regulador de voltaje 7805 que entrega a su salida 5V.
- Para activar la carga conectamos la salida D0 hacia el MOC3010. Esto protege a la tarjeta de desarrollo puesto que se va a trabajar con alto voltaje; además se dispone de un TRIAC BTA16 que actúa como interruptor sin contacto.
- La conexión con Ubidots se hace de manera inalámbrica mediante el uso de la red wifi del hogar.

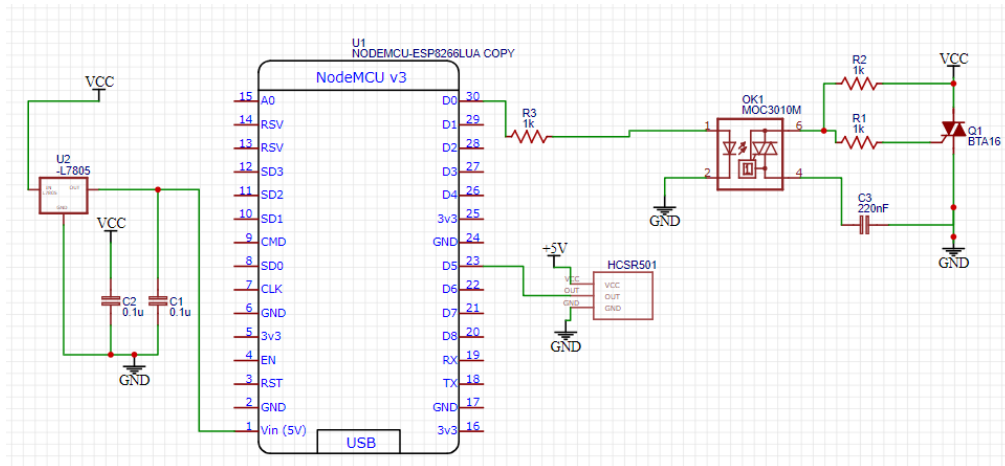


Figura14-2: Conexiones del módulo detector de movimiento.

Realizado por: Pillajo, Luis; 2022.

2.5 Diseño del Software para el prototipo doméstico

Para diseñar el Software se utiliza el IDE de Arduino partiendo por definir los requerimientos que debe cumplir. A continuación, se presentan los diagramas de flujo de las distintas etapas, las funciones y las librerías usadas.

2.5.1 Requerimientos del software del prototipo doméstico

- Conectar a cada dispositivo con Ubidots a través de la red doméstica de internet usando el SSID, PASSWORD y el TOKEN.
- Obtener mediciones de voltaje, corriente, temperatura, luminosidad y detección de movimiento.
- Determinar los valores efectivos de voltaje y corriente a través de la librería EmonLib.
- Visualizar el estado activado o desactivado de los dispositivos.
- Transmitir los datos y almacenarlos en Ubidots para poder interactuar con ellos.
- Controlar la iluminación y ventilación en el hogar de manera remota mediante la interfaz de Ubidots o la aplicación.

2.5.2 Diagrama de flujo para la etapa de medición

En la figura 15-2, se presenta el diagrama de flujo que detalla el funcionamiento del programa que gobierna las placas de medición de voltaje, corriente, temperatura, nivel de luminosidad y detección de movimiento.

Para el ciclo de inicio:

- Se incluye la librería EmonLib (Hudson 2012) que se encarga de convertir los valores de voltaje y corriente que ingresan por las entradas analógicas de la tarjeta Arduino nano.
- Inicialización de las librerías ESP8266WiFi.h, DNSServer.h, ESP8266WebServer.h, WiFiManager.h, Arduino.h (Grokhotkov 2017).
- Inicialización de la librería UbidotsESPMQTT.h (Hernandez 2021).
- Se inicializan las entradas y salidas.
- Activación del puerto serial.

Para el ciclo de repetición:

- Se mide el voltaje, corriente, temperatura y se detecta movimiento a través de sensores.
- En las placas de desarrollo los datos de los sensores se procesan: en el caso del sensor de temperatura se cambia el valor de milivoltios a grados centígrados, los datos de luminosidad se convierten a luxes; para los datos de voltaje y corriente se calculan sus valores efectivos a través de la librería EmonLib y finalmente el sensor de movimiento mide las variaciones infrarrojas producidas por los objetos en movimiento.
- Los datos obtenidos luego de esta conversión se almacenan en variables y se envían mediante comunicación inalámbrica hacia Ubidots donde se almacenan.

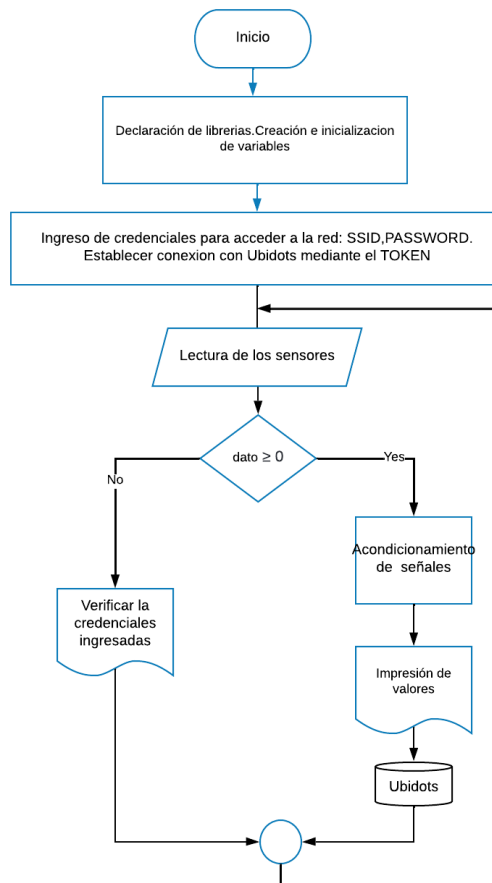


Figura15-2: Diagrama de flujo de la etapa de medición.

Realizado por: Pillajo, Luis; 2022.

2.5.3 Diagrama de flujo de la etapa de activación

El diagrama de flujo de la figura 16-2, muestra el funcionamiento de la placa como interruptor inalámbrico que permite activar o desactivar luces y electrodomésticos pequeños del hogar de manera remota.

Para el ciclo de inicio:

- Se inicializan las entradas y salidas.

Para el ciclo de repetición:

- Se comienza por identificar el estado en el que se encuentra el dispositivo, este se envía hacia una variable denominada estado y se evalúa.
- Cuando se detecte que la variable estado cambia, se envía un pulso desde el terminal GPIO hacia el opto acoplador este pulso activará o desactivará la carga mediante un TRIAC.

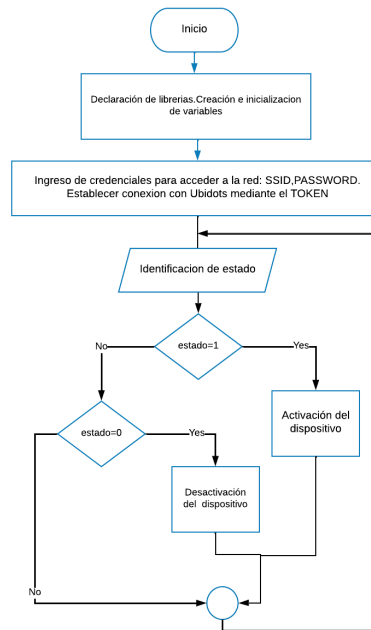


Figura16-2: Diagrama de flujo de la etapa de activación.

Realizado por: Pillajo, Luis; 2022.

2.6 Diseño de carcasa para el prototipo domótico

Mediante el software Fusion360 se diseñaron las carcasas para cada uno de los dispositivos que componen el módulo; esto se hizo con el objetivo de proteger al usuario de descargas eléctricas considerando que se está trabajando con voltajes y corrientes elevadas que podrían causar daños y además, se protege de agentes externos que podrían afectar el funcionamiento de los circuitos. En cada uno de los bocetos realizados se detallan las dimensiones en caso de querer replicar el diseño para esto, se utilizó una impresora 3D Artillery Genius y PLA. En la figura 17-2, se muestra la creación de las carcasas para los módulos de activación proceso de impresión que se replica para las demás carcasas, variando únicamente el tiempo que tarda en completar la impresión.



Figura17-2: Impresión de carcasas con Artillery Genius y PLA.

Realizado por: Pillajo, Luis; 2022.

Para el módulo de medición de voltaje y corriente se creó una carcasa que abarca todo el circuito. Considerando que los sensores van a conectarse de manera externa, se realizaron las perforaciones necesarias para las entradas de alimentación y conexión con los sensores tal como se muestra en la figura 18-2.

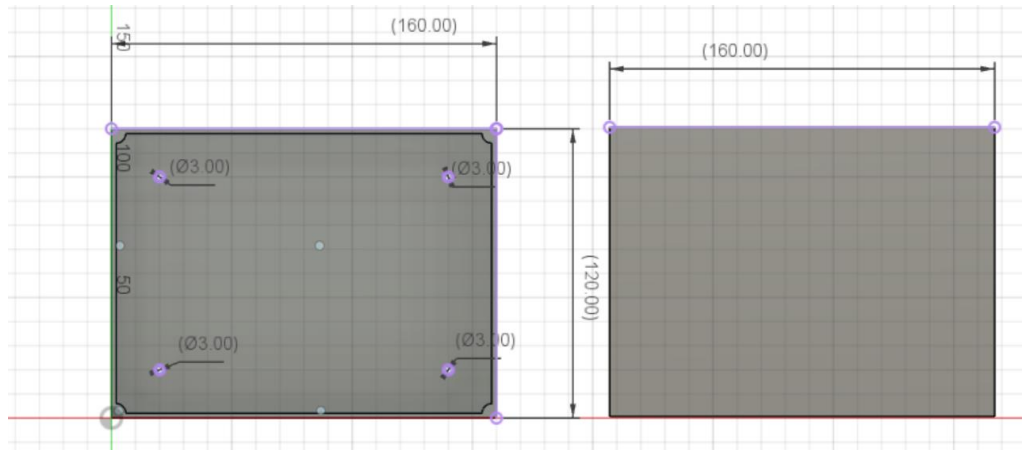


Figura18-2: Carcasa para el módulo medidor de voltaje y corriente.

Realizado por: Pillajo, Luis; 2022.

En la figura 19-2, se muestra la carcasa elaborada para el módulo de medición de luminosidad, esta carcasa se imprimió en 3D a un espesor de 2 mm y una altura de capa de 0.3 mm.

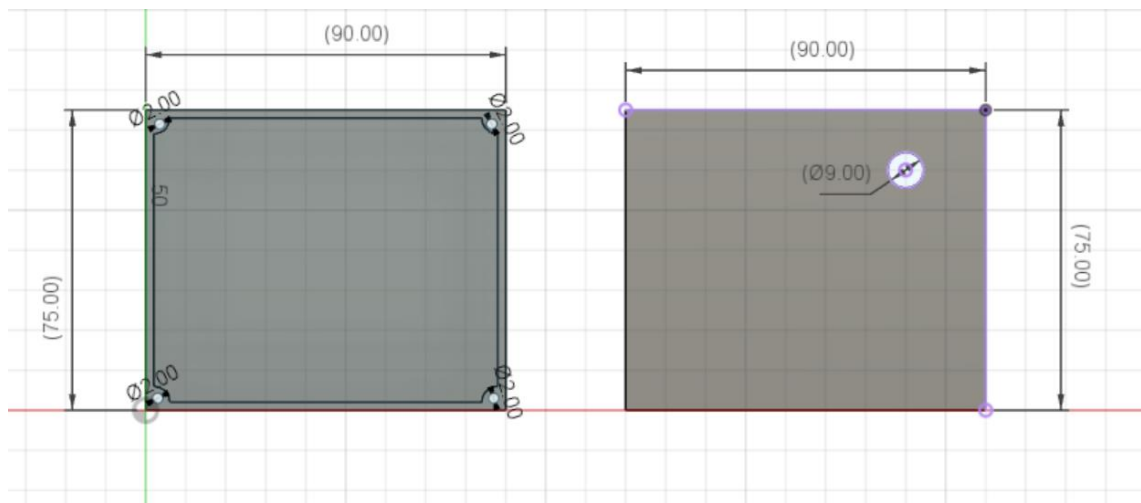


Figura19-2: Carcasa para el dispositivo de medición de luminosidad.

Realizado por: Pillajo, Luis; 2022.

En caso del módulo medidor de luminosidad se creó una carcasa tal como se muestra la figura 20-2 que permite ubicar un pulsador y un conector para el dispositivo que se va controlar. De esta manera se presenta un dispositivo estéticamente más agradable.

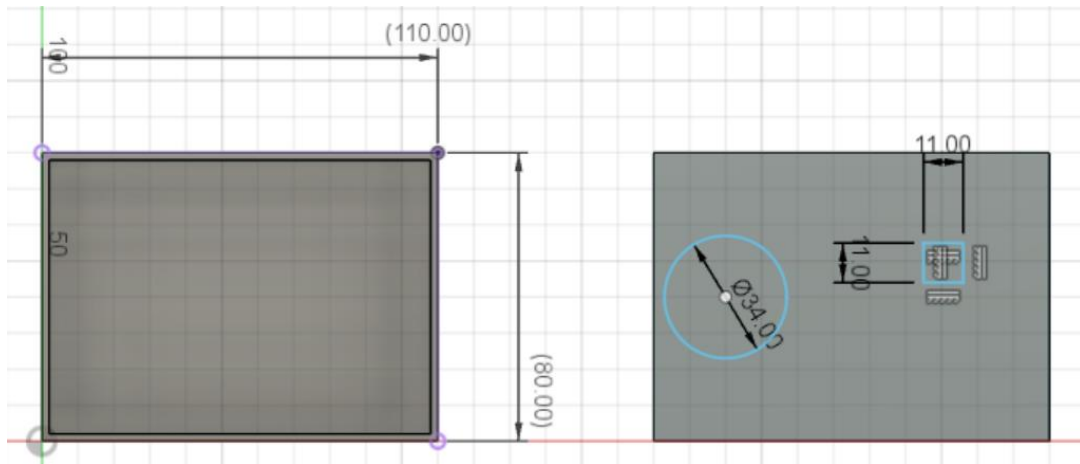


Figura20-2: Carcasa para el módulo medidor de luminosidad.

Realizado por: Pillajo, Luis; 2022.

Para el dispositivo denominado interruptor inteligente, se replicaron las carcasas basándose en el plano de la figura 21-2. Lo que se busca es aislar las placas de la intemperie para evitar corto circuitos o fallas con el dispositivo.

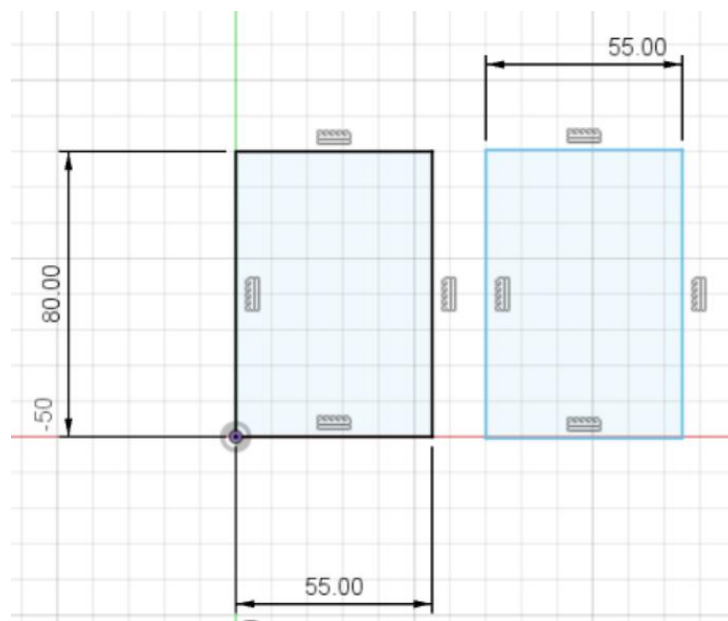


Figura21-2: Carcasa para módulos de activación.

Realizado por: Pillajo, Luis; 2022.

El módulo detector de movimiento posee una carcasa fabricada de acuerdo con el plano de la figura 22-2. Se procuró un diseño compacto que no ocupe demasiado espacio y que pueda ser instalado en cualquier lugar de la bodega.

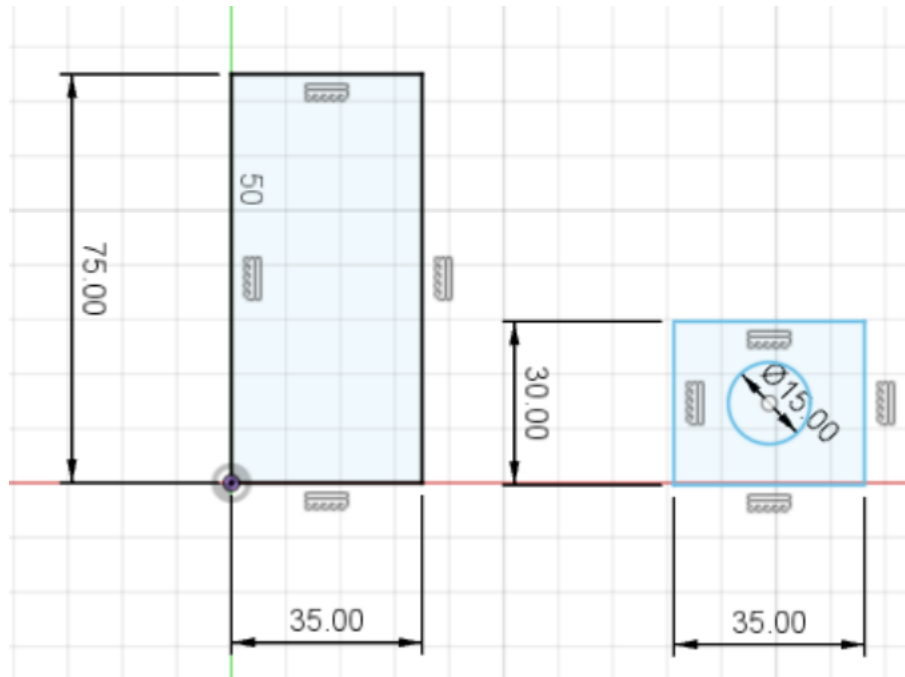


Figura22-2: Carcasa para el módulo detector de movimiento.

Realizado por: Pillajo, Luis; 2022.

2.7 Instalación del prototipo domótico en el hogar

El prototipo domótico se instaló en una vivienda en la ciudad de Guayaquil donde la iluminación de todo el hogar se activa de manera manual mediante el uso de interruptores. Estos se reemplazaron por interruptores con tecnología IoT que tienen conexión inalámbrica con el servidor denominado Ubidots.

Los electrodomésticos disponibles son un radio y un ventilador. Este último tiene un mayor tiempo de uso debido a que en la ciudad Guayaquil se presentan elevados niveles de temperatura. Para automatizarlos se crearon tomacorrientes inteligentes de activación remota.

El prototipo domótico que se muestra en la figura 23-2, se instaló en un hogar que dispone de cinco estancias: dormitorio, sala, cocina, patio y bodega.



Figura23-2: Prototipo domótico completo previo a la instalación.

Realizado por: Pillajo, Luis; 2022.

Para controlar la iluminación se reemplazó los interruptores mecánicos tal como se muestra en la figura 24-2. Se añadió un switch para poder reiniciar el dispositivo y un pulsador como alternativa manual de activación. De esta manera se automatizó la iluminación.



Figura24-2: Instalación del interruptor IoT.

Realizado por: Pillajo, Luis; 2022.

Para controlar el ventilador ubicado en la habitación se instaló un tomacorriente tal como se muestra en la figura 25-2 que tiene un sensor de temperatura. A través de un evento creado en la interfaz de Ubidots, se programa la activación de la ventilación cuando la temperatura sobrepase

el set point; de la misma manera se crea otro evento que desactiva el ventilador cuando disminuye la temperatura por debajo del valor configurado.

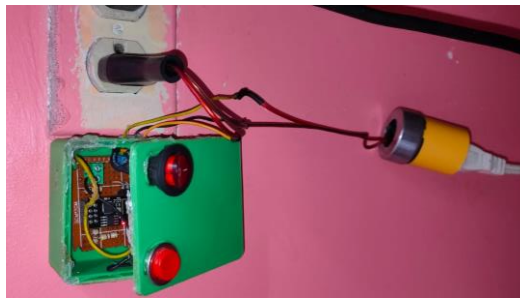


Figura25-2: Tomacorriente para control del ventilador.

Realizado por: Pillajo, Luis; 2022.

Instalación de un tomacorriente para gobernar un radio: este electrodoméstico es antiguo y no posee conexión a internet. Lo que se busca con la adición del dispositivo IoT es automatizar el encendido y apagado por medio del uso de la interfaz desde un computador o a través de la aplicación, tal como se muestra en la figura 26-2.



Figura26-2: Instalación del tomacorriente para controlar el radio.

Realizado por: Pillajo, Luis; 2022.

La instalación de la placa de medición de consumo energético comienza por conectar los sensores tanto de voltaje y corriente a la línea principal del hogar tal como se muestra en la figura 27-2. La carcaza que se creó facilita la instalación y protege al dispositivo de afectaciones externas.



Figura27-2: Conexión del medidor de consumo a la red principal.

Realizado por: Pillajo, Luis; 2022.

Cabe mencionar que para que el sensor de corriente o CT entregue un valor de salida, debe instalarse obligatoriamente en el cable de fase; para esto es necesario identificar con ayuda de un comprobador tal como se muestra en la figura 28-2.



Figura28-2: Comprobación de fase.

Realizado por: Pillajo, Luis; 2022.

En la bodega se instaló el dispositivo que controla la iluminación mediante el sensor de movimiento tal como se muestra en la figura 29-2. Se creó un evento que activa la iluminación cada vez que el sensor envié un valor de 1 hacia Ubidots.



Figura29-2: Instalación del detector de movimiento en la bodega.

Realizado por: Pillajo, Luis; 2022.

En la locación de sala se colocó el dispositivo que mide el nivel de luminosidad a través del sensor LDR tal como se muestra en la figura 30-2. En la interfaz de Ubidots se muestra la cantidad de luxes. El objetivo de este dispositivo es variar la intensidad del bombillo para que el valor de luxes de la estancia se mantenga en el valor requerido que para esta locación son 300 luxes. De esta manera se puede aprovechar la luz natural que ingresa a través de la ventana y reducir el consumo eléctrico de la bombilla.



Figura30-2: Instalación del interruptor IoT en la sala del hogar.

Realizado por: Pillajo, Luis; 2022.

CAPITULO III

3. VALIDACIÓN DE PROTOTIPO

En el presente capítulo se ejecutan las pruebas correspondientes que permiten validar el prototipo domótico implementado. Las pruebas que se realizaron son las siguientes:

- Comprobación de efectividad de los sensores (temperatura, movimiento nivel de luz).
- Estabilidad del prototipo domótico.
- Prueba de transmisión de datos.
- Prueba de latencia de comunicación.
- Potencia que se consume en el hogar.
- Comprobación de la reducción de consumo energético en el hogar.

3.1 Comprobacion de efectividad de los sensores

3.1.1 Validacion del sensor de temperatura

En este caso se implementó el circuito de prueba de la figura 1-3 y se programó la tarjeta Arduino para medir el valor de temperatura en la habitación con un intervalo de 5 min.

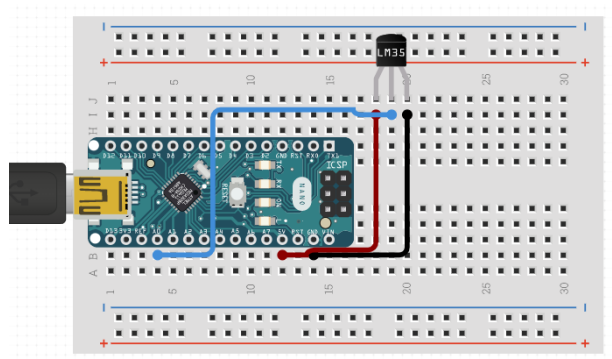


Figura 1-3: Circuito de prueba para el sensor LM35

Realizado por: Pillajo, Luis; 2022.

Tabla 1-3: Valores de temperatura medidos con el multímetro y el sensor LM35.

N° medición	Valor patrón °C	Valor aproximado	Ea.	% Er
1	30.80	30.33	0.47	1.53
2	33.66	33.72	-0.06	0.18
3	62.15	62.01	0.14	0.23
4	57.89	58.11	-0.22	0.38
5	48.65	49.03	-0.38	0.78
6	44.00	44.43	-0.43	0.98
7	38.91	39.04	-0.13	0.33
8	36.54	36.6	-0.06	0.16
9	34.66	35.14	-0.48	1.38
10	33.76	33.67	0.09	0.27
11	33.30	33.20	0.10	0.30
12	32.87	33.09	-0.22	0.67
13	32.79	33.28	-0.49	1.49
14	32.34	32.71	-0.37	1.14
15	31.94	32.20	-0.26	0.81
16	30.80	31.17	-0.37	1.20
17	30.70	31.17	-0.47	1.53
18	31.65	31.25	0.40	1.26
19	32.66	32.23	0.43	1.32
20	32.55	32.71	-0.16	0.49

Realizado por: Pillajo, Luis; 2022.

Luego calculamos el error absoluto de la medición con la siguiente formula:

$$Ea = \text{valor patrón} - \text{valor aproximado} \quad (2)$$

Donde:

Valor patrón: es el valor obtenido por el multímetro TRUE RMS que nos garantiza un valor más confiable de medida.

Valor aproximado: valor obtenido del monitor serial de Arduino.

Para calcular el error relativo porcentual utilizamos la siguiente formula:

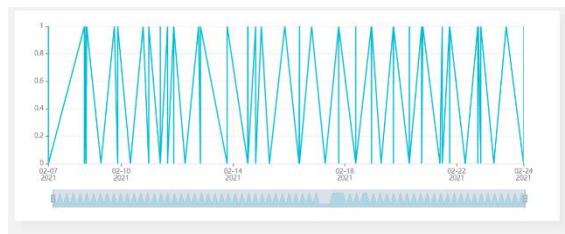
$$\%Er = \frac{|\text{valor patrón} - \text{valor aproximado}|}{\text{valor real}} (100) \quad (3)$$

Luego de analizar la hoja de trabajo del sensor LM35 mostrada en el ANEXO A, se identifica que el sensor tiene una resolución de 10 mV por cada grado centígrado que mide, el sensor tiene un

error de medición a temperatura ambiente de 0.2 °C, el error del dispositivo patrón que para mediciones menores a 400°C equivale a 0.01°C. Tal como determina (Mendenhall, Wackerly & Scheaffer, 2014) para que una medición se considere aceptable el error no debe superar el $\pm 0.5^\circ\text{C}$, lo que demuestra que la medición del sensor es buena, además analizando el error relativo porcentual, verificamos que se encuentra dentro del rango entre 1% y 5% para considerar que la medición es correcta según (Santo y Lacumberry 2005). De esta manera se puede concluir que el dispositivo lee correctamente el valor de temperatura en la habitación.

3.1.2 *Comprobacion del sensor de movimiento*

Para determinar la efectividad del sensor de movimiento, se creó un evento en el servidor, que activa la iluminación de la bodega cuando detecte movimiento. El sensor envía un valor de uno cuando detecta movimiento y cero cuando no. Luego de varias pruebas el servidor Ubidots obtuvo varios datos mismos que se muestran en la gráfica 1-3 que representan los datos recibidos del sensor.



Grafica1-3: Datos enviados por el sensor de movimiento.
Realizado por: Pillajo, Luis, 2022.

3.1.3 *Comprobacion del sensor de luminosidad*

Para evaluar la efectividad del sensor de luminosidad se tomaron 10 muestras en la sala, de 5:00 am a 20:00 pm en intervalos de 1 hora, se obtuvieron los datos de la tabla 2-3.

Tabla 2-3: Lecturas obtenidas del sensor LDR.

Muestra	Lectura 1	Lectura 2
4:00	7	290
6:00	108	300
8:00	150	291
10:00	172	289
12:00	178	289
14:00	182	285
16:00	184	305
18:00	155	310
20:00	10	300
22:00	6	299

Realizado por: Pillajo, Luis; 2022.

Los datos de la lectura 1 muestran los valores de luxes medidos por el sensor sin variar la intensidad de luz, para tomar los datos mostrados en la lectura 2, se activó el evento que varía la intensidad de iluminación en la sala. Luego de analizar los datos de la tabla 2-3 se determina que el prototipo domótico permite variar la iluminación para mantener el nivel de luminosidad de 300 luxes como sugiere la norma.

3.2 Estabilidad del prototipo domótico

El propósito de esta prueba es conocer el coeficiente de variación que permita verificar que el prototipo domótico sea estable y no se considere de alta varianza, este coeficiente según INEN debe ser menor o igual al 1.293% para nuevos proyecto (INEN 2010).

Para establecer si el prototipo domótico actúa de una manera estable se procedió a tomar diez muestras de voltaje y corriente con la ayuda de un multímetro y un amperímetro tal como se muestra en la Figura 4-3. En función de esto, se compara con el valor de voltaje y corriente mostrado en Ubidots y se verifica si existe una variación considerable.



Figura 4-3: Valores de voltaje y corriente del hogar medidos.

Realizado por: Pillajo, Luis; 2022.

Luego se compara con los valores de la Figura 5-3 que representan los datos obtenidos por la plataforma Ubidots. De esta manera se determinar si existen errores de medición.

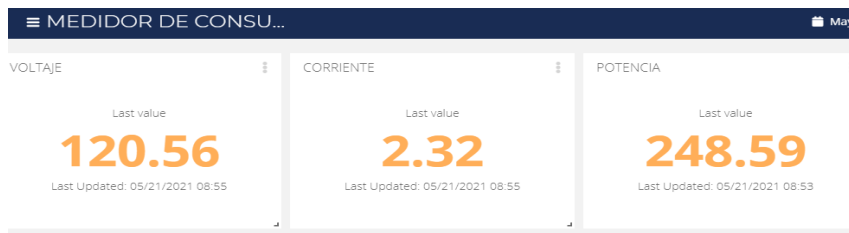


Figura 5-3: Valores de voltaje y corriente mostrados en la interfaz de Ubidots.

Realizado por: Pillajo, Luis; 2022.

La tabla 3-3, indica los resultados de voltaje y corriente obtenidos con el multímetro y los presentados por Ubidots en su interfaz.

Tabla 3-3: Resultados de las mediciones de voltaje y corriente

Nº	Voltaje medido (V)	Corriente medida (I)	Voltaje mostrado en Ubidots (V)	Corriente medida Ubidots (I)
1	122.3	2.421	120.56	2.32
2	122.1	2.422	122.03	2.38
3	122.3	2.422	121.19	2.38
4	122.2	2.422	120.08	2.39
5	122.3	2.422	122.2	2.39
6	122.2	2.422	121.77	2.39
7	122.3	2.422	122.43	2.36
8	122.1	2.422	121.33	2.39
9	122.3	2.421	120.42	2.36
10	122.1	2.422	121.47	2.35
Media	122.22	2.4218	121.348	2.371
Varianza	0.008444444	1.77778E-07	0.629995556	0.000543333
Desviación estándar	0.091893658	0.000421637	0.793722594	0.023309512
Coefficiente de variación (%)	0.07518709	0.01741007	0.65408791	0.98310888

Realizado por: Pillajo, Luis; 2022.

Analizando la tabla 3-3, se establece que el prototipo domótico es estable y de baja varianza ya que el coeficiente de variación es de 0.07% en 122.22 V, 0.65% en 121.34V, 0.01% en 2.42 A y 0.98% en 2.37 A. En los cuatro casos los valores no exceden el 1.293%, que estipula el INEN; además se observa una estabilidad ideal al sistema ya que no existen alteraciones considerables en los valores de voltaje y corriente.

3.3 Prueba de transmisión de datos

Para verificar el número de datos que son enviados desde el prototipo domótico hacia Ubidots, se analiza la variable corriente en un intervalo de recolección de datos de 6 am a 6 pm y se lo compara con la información que indica la base de datos. En la tabla 4-3 se muestra la cantidad obtenidos por Ubidots.

Tabla 4-3: Detalle de los datos procesados

Intervalo de tiempo	Número de datos recibidos
5 minutos	5 datos
1 hora	12 datos
12 horas	144 datos

Realizado por: Pillajo, Luis; 2022.

Los datos se almacenan durante 12 horas y se comprueba a través del grafico 2-3, que efectivamente están siendo recibidos por la plataforma de Ubidots.

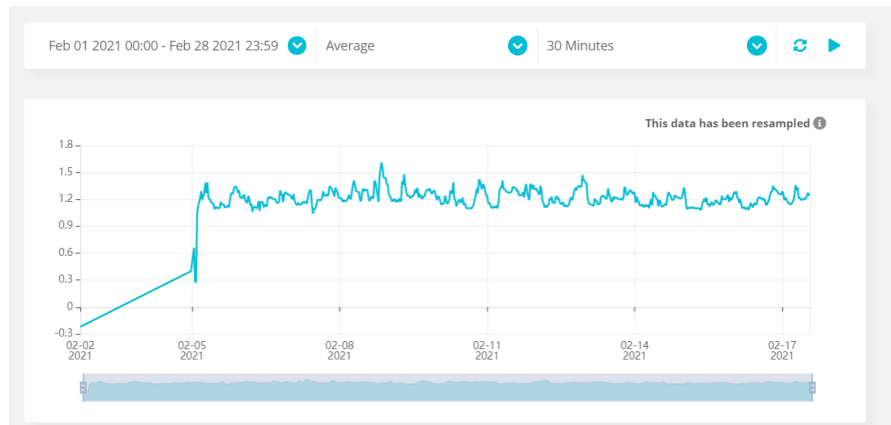


Gráfico 1-3: Datos de corriente recibidos por Ubidots.

Realizado por: Pillajo, Luis; 2022.

En función de lo antes mencionado se comprueba que no existe pérdida de datos durante todo el tiempo en que opera el prototipo domótico. Cabe mencionar que durante la prueba de transmisión de datos la conexión del prototipo domótico a internet se mantuvo constante por lo que se concluye que para que no exista pérdida de información entre el prototipo domótico y Ubidots se requiere de manera obligatoria una conexión a internet estable debido a que no se almacena información sino solo se transmite y al no tener conexión con el servidor los datos se pierden hasta que se reestablezca la comunicación.

3.4 Prueba de latencia de comunicación

Para esta prueba se utilizó la tarjeta NodeMCU, primero se identifica la IP del dispositivo tal como se muestra en la Figura6-3. La IP es 192.168.1.2

Detalles de red	
Dirección IP	192.168.1.2
Dirección MAC	EC:FA:BC:08:20:8E
Proveedor de MAC	Espressif

Figura 6-3: Dirección IP del NodeMCU.

Realizado por: Pillajo, Luis; 2022.

Sabiendo esta dirección utilizamos la consola de comandos de Windows CMD e introducimos el comando ping y se obtuvo los resultados de la Figura 7-3.

```
C:\Users\edpc_>ping 192.168.1.2
Haciendo ping a 192.168.1.2 con 32 bytes de datos:
Respuesta desde 192.168.1.2: bytes=32 tiempo=18ms TTL=255
Respuesta desde 192.168.1.2: bytes=32 tiempo=2ms TTL=255
Respuesta desde 192.168.1.2: bytes=32 tiempo=5ms TTL=255
Respuesta desde 192.168.1.2: bytes=32 tiempo=2ms TTL=255

Estadísticas de ping para 192.168.1.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 2ms, Máximo = 18ms, Media = 6ms
```

Figura 7-3: Datos obtenidos en el CMD de Windows.

Realizado por: Pillajo, Luis; 2022.

La latencia de comunicación entre Ubidots depende netamente del ancho de banda que se tenga puesto que todos los dispositivos van a estar en el rango de cobertura de wifi. Esto quiere decir que los dispositivos van a estar cercanos uno de otro por lo que se reduce la latencia de comunicación. Esto se puede comprobar al activar el prototipo domótico desde la página de Ubidots el tiempo que tarda en realizar la acción es casi imperceptible.

3.5 Potencia que se consume en el hogar

Para determinar la potencia que se consume en el hogar se utilizan los datos de los sensores de voltaje y corriente que se presentan en Ubidots. Se tomaron 30 muestras en un periodo de una hora y se obtienen los datos de la tabla 5-3.

Tabla 5-3: Mediciones de voltaje y corriente que se consumen en el hogar

Nº de muestra	Tensión AC (V)	Corrientes DC (A)
1	110.5	0.9
2	108.8	0.8
3	110.8	0.8
4	111	0.8
5	110.8	0.9
6	109.6	1.1
7	107.9	1.0
8	109.1	1.0
9	108.1	1.1
10	111	1.2
11	110.2	0.8
12	110.4	1.1
13	110.2	0.9
14	111.4	1.2
15	109.5	1.2
16	109.2	1.2
17	109.2	1.0
18	108.2	1.0
19	110.2	1.0
20	110.7	1.1
21	112.3	1.1
22	111.6	1.2
23	109.2	1.3
24	113.1	1.3
25	112.4	1.2
26	111.8	1.2
27	110.4	1.1
28	111.4	1.2
29	109	1.2
30	111.3	1.2
Media	110.31	1.06

Realizado por: Pillajo, Luis;2022.

La tabla 5-3, muestra que al analizar las 30 muestras de voltaje y corriente existe un consumo promedio de 110.31 V y 1.06 A. Se debe considerar que el valor de corriente que se obtiene es en

directa; para realizar el cálculo de potencia del prototipo domótico se realiza el cálculo de I_{AC} , mediante la ecuación:

$$I_{AC} = \frac{I_p}{\sqrt{2}} \quad (3)$$

Donde:

$$I_p = I_{DC} * \frac{\pi}{2} \quad (4)$$

Reemplazando la ecuación 4 en la ecuación 3 se obtiene:

$$I_{AC} = I_{DC} * \frac{\pi}{2\sqrt{2}} \quad (5)$$

A partir de la ecuación anterior, se obtiene el consumo de corriente I_{AC}

$$I_{AC} = 1.06 * \frac{\pi}{2\sqrt{2}}$$

$$I_{AC} = 1.17A$$

Una vez obtenida la corriente y el voltaje se procede a calcular la potencia del dispositivo a través de la siguiente expresión:

$$P = V_{AC} * I_{AC} \quad (6)$$

$$P = 116.18V * 1.17 = 129.06 W$$

Para determinar la potencia que se consume en un mes aplicamos la siguiente formula y obtenemos el consumo en kilovatios.

$$Consumo = \frac{P * 730}{1000} = 94.21 KW \quad (7)$$

La tabla 6-3, indica el voltaje de alimentación, el consumo de corriente y la potencia que se consume en hogar luego de la implementación del módulo en el hogar.

Tabla 6-3: Consumo de potencia en el hogar.

Consumo de voltaje	116.18V
Consumo de corriente	1.48A.
Potencia	94.21 KW

Realizado por: Pillajo, Luis;2022.

De acuerdo con los datos de la tabla 6-3, se determina que el consumo mensual de potencia del hogar se aproxima a los valores obtenidos de las planillas de los meses donde estuvo instalado el módulo en el hogar.

3.6 Comprobación de la reducción de consumo energético en el hogar

De acuerdo con los estudios realizados por CONELEC se considera que el 49% del consumo energético de los hogares en el Ecuador es debido a la iluminación, es por esto que controlar la activación y desactivación de las luces del hogar produjo una reducción significativa en los valores de consumo energético y en los costos de las planillas obtenidas de la CNEL sucursal Guayas.

Se hizo un estudio de las planillas obtenidas de la CNEL acerca de los meses anteriores a la instalación del módulo domótico. Para esto se identificó el consumo mensual y el costo de la planilla. Estos valores se detallan en la tabla 7-3.

Tabla 7-3: Consumo eléctrico de los meses previos a la instalación del módulo.

Mes	Consumo de potencia	Costo de planilla
Febrero	123 KW	5.18 \$
Marzo	132 KW	5.39 \$
Abril	129 KW	5.19 \$
Consumo promedio	128 KW	5.25 \$

Realizado por: Pillajo, Luis;2022.

A partir del mes de mayo de 2021 se instaló el módulo domótico en el hogar durante tres meses y se determinaron los datos de la tabla 8-3.

Tabla 8-3: Consumo eléctrico de los meses posteriores a la instalación del módulo.

Mes	Consumo de potencia	Costo de planilla
Mayo	108 KW	3.79 \$
Junio	93 KW	3.55 \$
Julio	99 KW	3.49 \$
Consumo promedio	100 KW	3.61 \$

Realizado por: Pillajo, Luis;2022.

De esta manera se puede comprobar que la instalación del prototipo domótico en el hogar representa una reducción del consumo energético igual al 21.9%, lo que implica una reducción en el costo de la planilla de 31.2%.

Para determinar si la reducción se debió a la instalación del prototipo domótico, a partir del mes de junio del año 2021 se desactivo el prototipo y las actividades en el hogar se desarrollaron con normalidad. Esto evidencio que a partir de ese mes y los posteriores donde el prototipo domótico se encontraba desactivado el consumo volvió a elevarse y por ende el costo de las planillas aumento. Tal como se muestra en la figura 8-3, que detalla los valores mensuales de las planillas de consumo, estos datos se obtuvieron de la CNEEL.

SEL.	PDF	SECUENCIA	CLIENTE (CUEN)	CONTRATO	FECHA DE EMISION	NÚMERO DE DOCUMENTO	TOTAL CONSUMO	FECHA DE VENCIMIENTO
<input type="checkbox"/>		1	0400799682	200019274543	25/10/2021	001910501131	5,18	09/11/20
<input type="checkbox"/>		2	0400799682	200019274543	24/09/2021	001920475346	5,18	09/10/20
<input type="checkbox"/>		3	0400799682	200019274543	24/08/2021	001920437944	5,18	08/09/20
<input type="checkbox"/>		4	0400799682	200019274543	29/07/2021	002441103940	3,49	13/08/20
<input type="checkbox"/>		5	0400799682	200019274543	25/05/2021	002930932747	3,79	09/06/20
<input type="checkbox"/>		6	0400799682	200019274543	23/04/2021	002980899641	5,19	08/05/20

Figura 8-3: Planillas mensuales año 2021 según CENEL EP.

Realizado por: Pillajo, Luis;2022.

CAPITULO IV

4. EVALUACIÓN ECONÓMICA

Este capítulo evalúa los costos que representaron la implementación del prototipo domótico y lo compara con sistemas domóticos comerciales que presentan características similares para determinar si la implementación de este prototipo domótico representa un beneficio económico respecto a módulos que se pueden encontrar en el mercado.

La implementación del prototipo domótico que se desarrolló tiene un costo a nivel de hardware de 144.10 dólares americanos (USD) y se lo detalla en la tabla 1-4.

Tabla 1-4: Costo general en hardware del módulo domótico implementado.

Elementos	Precio unitario (\$)	Cantidad	Precio total (\$)
Sensor LM35	2	1	2
NodeMCU	8	2	16
ESP01	5	6	30
Arduino Nano	7	1	7
Sensor hc-sr501	2	1	2
LDR	1	1	1
MOC 3021	0.35	6	2.10
Fuente de alimentación	3	6	18
Rollo de PLA	21	1	21
Gastos adicionales	-	-	45
		Total	144.10

Realizado por: Pillajo, Luis;2022.

Comercialmente no existe un módulo similar en el mercado así que se optó por comparar con sistemas domóticos comerciales. Z-wave Nvr es un kit compuesto por varios dispositivos destinados a la automatización y seguridad de hogares inteligentes, Reolink comercializa un kit enfocado al confort y seguridad de hogares. Luego de analizar los costos se obtuvo los valores de la tabla 2-4.

Tabla 2-4: Costos de comercialización de módulos domóticos similares

Empresa	Nombre	Precio (\$)
Z-wave Nvr	Kit Domótica Cámara Sensores Magnéticos Inalámbricos	560
ReolinK	Kit Domótica Seguridad Y Confort	500

Realizado por: Pillajo, Luis;2022.

Analizando las dos tablas, se demuestra que prototipo domótico construido tiene un ahorro del 73% respecto al Kit Z-WAVE y un ahorro del 68% con respecto al Kit de ReoLink, por lo que se considera un porcentaje aceptable de inversión teniendo en cuenta que este equipo dispone de características similares a los comerciales.

CONCLUSIONES

- Se implementó un módulo domótico que permite reducir el consumo energético del hogar mediante la gestión de actividades de iluminación y ventilación utilizando el servidor Ubidots.
- De la prueba de fiabilidad del sensor de temperatura, se comprobó que la medición es correcta debido a que los valores de error absoluto no superan ± 0.5 como se establece en la bibliografía para considerar una medida aceptable.
- El servidor permite interactuar con los datos obtenidos de los sensores por medio de la creación de eventos que automatizan las actividades del hogar sin la necesidad de volver a programar las tarjetas de desarrollo.
- Se comprobó que, al automatizar la iluminación y ventilación del hogar mediante el uso del módulo domótico, se produjo una reducción del 21.9% en el consumo energético del hogar tal como se evidencia en las planillas emitidas por la empresa eléctrica en los meses donde estuvo instalado el prototipo domótico.
- Se demostró la efectividad del módulo domótico a la hora de variar la intensidad de la iluminación en la sala con el objetivo de aprovechar la iluminación natural y mantener un valor de luxes de acuerdo con las recomendaciones de la NEC.
- Se determinó que el módulo es estable y de poca varianza al analizar los datos de la prueba de estabilidad donde los coeficientes de variación no exceden el 1.293% como lo determina la norma en la creación de proyectos nuevos; además presenta una certeza de comunicación del 100% ya que no se evidencia pérdida de datos.
- De la evaluación económica se determina que el módulo domótico representa un ahorro del 73% respecto al Kit Z-WAVE y un 68% con respecto al Kit de ReoLink, brindando una mejor relación costo beneficio.

RECOMENDACIONES

- Se recomienda el uso de un servidor propio que use una red de área local para que el módulo domótico pueda funcionar en caso de no tener internet.
- Para proyectos futuros, se recomienda añadir la posibilidad de usar comandos de voz como una alternativa de activación remota.
- Para obtener un valor de consumo más fiable se recomienda tomar más muestras durante todo el mes para determinar un valor de consumo que se aproxime más al valor obtenido en la planilla de CELEC.
- Se recomienda la adición de técnicas de machine learning para poder desarrollar rutinas que se ajusten a los requerimientos de cada usuario.
- Perfeccionar el apartado estético del prototipo domótico mediante el diseño de dispositivos más pequeños utilizando tecnología SMD para miniaturizar los circuitos y carcasas.
- Replicar las placas de medición de consumo energético para conseguir desagregación de cargas y evaluar el consumo del hogar mediante técnicas de carga no intrusivas.
- El gobierno debería fomentar la implementación de proyectos domóticos destinados a la reducción de consumo energético de hogares, como alternativa para reducir el gasto público debido al subsidio eléctrico.
- Usar el presente proyecto como referencia para crear dispositivos domóticos de bajo costo destinados a reducir el consumo energético y mejorar el estilo de vida de las personas.

BIBLIOGRAFÍA

ALLIANCE, Z., ZigBee Specification. *Standard*, 2015, Oct, ISSN 0040-8905.

ALVARO VILCA, W.A., *Diseño y construcción de un prototipo para el control de iluminación natural y artificial en una habitación*. 2011. S.l.: Escuela Politécnica Nacional.

ANDRÉS, C. y GODOY, G., Sistema de seguridad para locales comerciales mediante Raspberry Pi, cámara y sensor PIR. *Revista Virtual Universidad Católica del Norte*, 2017.vol. 0, no. 51, pp. 175–193. ISSN 0124-5821.

ARCOTEL, Estudio de la Demanda Eléctrica. *Balance Energético Nacional*, 2019.no. 3, pp. 1–71. ISSN 10986596.

ARREDONDO GARCÍA, J.L., *AHORRO DE ENERGÍA APLICADO A CASAS INTELIGENTES, (DOMÓTICA)*. S.l. 2010.: UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.

BEAL, V., What is DECT? | Webopedia. *webopedia* [en línea] ,2007.. [Consulta: 30 noviembre 2021]. Disponible en: <https://www.webopedia.com/definicions/dect/>.

BOHÓRQUEZ HERAS, B.B., *Sistema de información basado en el internet de las cosas (IOT) para mejorar la eficiencia energética en hogares y así contribuir a la reducción del impacto ambiental(Tesis de Grado)*. 2020. S.l.: Universidad Católica Santiago de Guayaquil.

CARDONA, E., STEEVEN, D., OSPINA, V. y MATEO, D., Raspberry pi: la tecnología reducida en placa. *Tecnología en sistemas de información*, 2019. pp. 1–13.

CARGUA RAMOS, P.M., *CHARACTERIZACIÓN DE ENERGÍA ELÉCTRICA DE CLIENTES RESIDENCIALES POR MEDIO DE IOT*. S.l. 2020.: Universidad Politécnica Salesiana.

ENRÍQUEZ HERRADOR, R., Guía de Usuario de Arduino. *I.T.I. Sistemas Universidad de Córdoba*, 2009. pp. 49.

ESPINOSA APOLO, B.A. y ORELLANA GUAYAS, M.E., *Desarrollo de aplicaciones de monitoreo y control basadas en IoT a través de la plataforma Ubidots. Aplicaciones a sistemas de automatización bajo entornos de simulación*. 2021. S.l.: s.n. ISBN 0301489696.

GROKHOTKOV, I., Librería ESP8266WiFi — documentación de ESP8266 Arduino Core - 2.4.0. [en línea]. 2017. [Consulta: 21 enero 2022]. Disponible en: <https://esp8266-arduino-spanish.readthedocs.io/es/latest/esp8266wifi/readme.html>.

GUILLÉN, V., QUESADA, F., LÓPEZ, M., ORELLANA, D. y SERRANO, A., Eficiencia energética en edificaciones residenciales. *Universidad de Cuenca*, 2017.ISSN 1390-9274.

HERNANDEZ, M., Connect a NodeMCU ESP8266 to Ubidots over HTTP | Ubidots Help Center. [en línea]. 2021. [Consulta: 21 enero 2022]. Disponible en: <https://help.ubidots.com/en/articles/513324-connect-a-nodemcu-esp8266-to-ubidots-over-mqtt>.

HUDSON, G., GitHub - openenergymonitor/EmonLib: Electricity monitoring library - install in Arduino IDE's libraries folder then restart the IDE. [en línea]. 2012. [Consulta: 21 enero 2022].

Disponible en: <https://github.com/openenergymonitor/EmonLib>.

HUNKELER, U., TRUONG, H.L. y STANFORD-CLARK, A., MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks. *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)* [en línea]. 2008. S.l.: IEEE, pp. 791–798. ISBN 978-1-4244-1796-4. DOI 10.1109/COMSWA.2008.4554519. Disponible en: <http://ieeexplore.ieee.org/document/4554519/>.

INEN, Documentacion. Presentacion De Tesis, Trabajos De Invertigacion. 2010. , vol. 2402.

JIMÉNEZ BUENDÍA, M., *Desarrollo de sistemas domóticos utilizando un enfoque dirigido por modelos* [en línea]. 2009.S.l.: Universidad Politécnica de Cartagena. Disponible en: <https://www.researchgate.net/publication/280445703>.

LLEDÓ SÁNCHEZ, E., Diseño de un sistema de control domótico basado en la plataforma Arduino. [en línea], 2012. [Consulta: 24 julio 2021]. Disponible en: <https://riunet.upv.es:443/handle/10251/18228>.

MAYNÉ, J., Estado actual de las comunicaciones inalámbricas. *Silica* [en línea], 2005.vol. 1, pp. 21–22. Disponible en: http://www.jcee.upc.edu/JCEE2004/pdf/EstadoActual_WirelessRF_SILICA.pdf.

MCROBERTS MICHAEL, Arduino Básico - Michael McRoberts - Google Libros. [en línea]. 2010. [Consulta: 23 febrero 2021]. Disponible en: <https://books.google.es/books?hl=es&lr=&id=kfZyDwAAQBAJ&oi=fnd&pg=PT4&dq=arduino&ots=1RuXs5JQGT&sig=JyV5kVIumULmEjJHqfFRyDOWvew#v=onepage&q=arduino&f=false>.

MENDENHALL, W., WACKERLY, D.D. y SCHEAFFER, R.L., Mathematical Statistics with Applications. *Biometrics* [en línea], 2014.vol. 48, no. 3, pp. 977. [Consulta: 16 diciembre 2021]. ISSN 0006341X. DOI 10.2307/2532372. Disponible en: <https://books.google.com.ec/books?hl=es&lr=&id=ITgGAAAAQBAJ&oi=fnd&pg=PP1&dq=Wackerly,+Mendenhall,+%26+Scheaffer,+2014&ots=QA13RMikjh&sig=YBSaN9WCTQ5sWuaQN9KzBWonPM#v=onepage&q=Wackerly%2C Mendenhall%2C %26 Scheaffer%2C 2014&f=false>.

MENDOZA, E., FUENTES, P., BENÍTEZ, I., REINA, D. y NÚÑEZ, J., Red de sensores inalámbricos multisalto para sistemas domóticos de bajo costo y área extendida. *Revista Iberoamericana de Automática e Informática industrial* [en línea], 2020. vol. 17, no. 4, pp. 412. ISSN 1697-7920. DOI 10.4995/riai.2020.12301. Disponible en: <https://polipapers.upv.es/index.php/RIAI/article/view/12301>.

MINISTERIO DE ELECTRICIDAD Y ENERGÍA RENOVABLE, Ministerio de Electricidad y Energía Renovable. Plan Maestro de Electrificación 2012-2021. [en línea], 2013. vol. 53, pp. 1689–1699. ISSN 1098-6596. Disponible en: <https://www.regulacionelectrica.gob.ec/wp->

content/uploads/downloads/2015/12/PME-2012-2021.pdf.

MOHAMMED, NOOR SALEH; SELMAN, N.H., Real-time monitoring of the prototype design of electric system by the ubidots platform. *International Journal of Electrical & Computer Engineering (2088-8708)* [en línea], 2021.vol. 11, no. 6, pp. p5568-5577. [Consulta: 27 noviembre 2021]. Disponible en: <https://web.s.ebscohost.com/abstract?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=20888708&AN=151460824&h=2NwRczuwQWj%2F8n7h5TQX3m1HZn2pp9UuPA3SjT%2ByXePlkAqJp2RrlbYfVYsxnF5b6uvgzQodIv%2BKDs02F8cY3A%3D%3D&crl=c&resultNs=AdminWebAuth&resultLo>.

NEIRA CLEMENTE, L.A., *DESARROLLO DE UN SISTEMA PARA LA MONITORIZACIÓN DE VARIABLES FÍSICAS BASADA EN TELEMETRÍA E IOT PARA INSTITUCIONES HOSPITALARIAS DE LA CIUDAD DE GUAYAQUIL*. 2021. S.l.: Universidad Politécnica Salesiana.

PALLÁS, R., *Sensores y Acondicionadores de Señal 4a - Ramón Pallás Areny* - Google Libros. [en línea]. 2003. [Consulta: 23 febrero 2021]. Disponible en: https://books.google.es/books?hl=es&lr=&id=Eevyk28_fVkc&oi=fnd&pg=PR11&dq=sensores+digitales&ots=JYgKZ9Cncd&sig=R_e_OkspRtjywt6Q7a-MbcFSESE#v=onepage&q=sensores+digitales&f=false.

PLASENCIA ALVA, P.F., *Caso: impacto de la política monetaria en el Perú en la actual economía* [en línea]. 2018. Mexico: Universidad de Lima. Disponible en: <https://hdl.handle.net/20.500.12724/8026>.

RACKLEY, S., *The future of wireless networking technology*. 2007. S.l.: s.n. ISBN 9780750667883.

RODRÍGUEZ, E.J.A., OCAMPO, J.W.M. y ORTEGA, C.A.S., Medición de temperatura: sensores termoelectricos. *Scientia Et Technica* [en línea], 2007.vol. 13, no. 34, pp. 1–6. ISSN 0122-1701. Disponible en: <https://doaj.org/article/482a2ec39c83453ba2a590ddfcee4a89>.

ROJAS VEGA, J.C., *Influencia del diseño de una edificación en el consumo de energía eléctrica por sistemas de climatización aplicado al Edificio Torre Sol de la ciudad de Machala*. 2015. S.l.: Universidad Politécnica Salesiana.

ROVING NETWORKS, Bluetooth data module command reference & advanced information: User's guide. *Data Sheets* [en línea], 2013.pp. 1–83. Disponible en: http://ww1.microchip.com/downloads/en/DeviceDoc/bluetooth_cr_UG-v1.0r.pdf.

RUIZ AYALA, D.C., VIDES HERRERA, C.A. y PARDO GARCÍA, A., Monitoreo de variables meteorológicas a través de un sistema inalámbrico de adquisición de datos. *REVISTA DE INVESTIGACIÓN, DESARROLLO E INNOVACIÓN* [en línea], 2018. vol. 8, no. 2, pp. 333. [Consulta: 24 julio 2021]. ISSN 2389-9417. DOI 10.19053/20278306.v8.n2.2018.7971. Disponible en: https://revistas.uptc.edu.co/index.php/investigacion_duitama/article/view/7971.

SANCLEMENTE, O., Casa domótica con Arduino. [en línea], 2016. pp. 182. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/75797/SANCLEMENTE - DISEÑO DE CASA DOMÓTICA CONTROLADA POR ARDUINO.pdf?sequence=5>.

SANTO, M. y LACUMBERRY, G., *El proceso de medición: Análisis y comunicación de datos experimentales* [en línea]. 2005.S.l.: s.n. ISBN 9506653445. Disponible en: https://www.unrc.edu.ar/unrc/digital/El_proceso_de_med.pdf.

SOMANTRI, S., YUSTIANA, I. y NUGRAHA, A., Electrical Consumption Monitoring and Controlling System Based on IoT and Mobile Application. *7th International Conference on ICT for Smart Society: AIoT for Smart Society, ICISS 2020 - Proceeding*, 2020. pp. 1–5. DOI 10.1109/ICISS50791.2020.9307556.

SOUZA OLIVEIRA, A. y FRANCO PEREIRA, M., Near Field Communication (NFC) Technology and Measurements. *Gestao e Producao*, 2015. vol. 21, no. 3, pp. 133–144. ISSN 23208430.

SUQUINAGUA OTAVALO, R.B. y MUÑOZ ABAD, E.O., *Diseño e implementación de un protocolo de comunicación domótico para interacción entre sensores y actuadores*. 2018. S.l.: Universidad de Cuenca.

VAN KRANENBURG, R. y BASSI, IoT Challenges. *Communications in Mobile Computing* [en línea], 2012.vol. 1, no. 1, pp. 9. ISSN 2192-1121. DOI 10.1186/2192-1121-1-9. Disponible en: <https://muxjournal.springeropen.com/articles/10.1186/2192-1121-1-9>.

VARGAS RAFAEL, A., *INCORPORACIÓN DE LA DOMÓTICA EN LA CONSTRUCCIÓN DE VIVIENDAS POPULARES*. 2019.S.l.: UNAM.

VETELINO, J.F. y REGHU, A., Introduction to sensors. *Introduction to Sensors*, no. December, 2017. pp. 1–180. DOI 10.1201/9781315218274.

ANEXOS

ANEXO A: HOJA DE DATOS TÉCNICOS DEL SENSOR LM35

LM35

LM35 Precision Centigrade Temperature Sensors



Literature Number: SNIS159B

November 2000

LM35 Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors, calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1/10°C at room temperature and ±1/10°C over a full -55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a -55 to +150°C temperature range, while the LM35C is rated for a -40 to +110°C range (10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear +10.0 mV/°C scale factor
- 0.1°C accuracy guaranteeable (at +25°C)
- Rated for full -55 to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1/10°C typical
- Low impedance output, 0.1 Ω for 1 mA load

Typical Applications

FIGURE 1. Basic Centigrade Temperature Sensor (°C to +150°C)

FIGURE 2. Full-Range Centigrade Temperature Sensor

© 2000 National Semiconductor Corporation DS006516 www.national.com

LM35 Precision Centigrade Temperature Sensors

Connection Diagrams

TO-46 Metal Can Package*

*Case is connected to negative pin (GND)
Order Number LM35H, LM35AH, LM35SCH, LM35CAH or LM35D
See NS Package Number H03H

SO-8 Small Outline Molded Package

N.C. = No Connection
Top View
Order Number LM35DM
See NS Package Number M08A

TO-92 Plastic Package

Order Number LM35CZ, LM35CAZ or LM35DZ
See NS Package Number Z03A

TO-220 Plastic Package*

*This is connected to the negative pin (GND).
Note: The LM35DZ circuit is different than the discontinued LM35DZ.
Order Number LM35DZT
See NS Package Number T03F

www.national.com 2

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35		LM35C, LM35D		Units
		Typical	Tested Limit (Note 4)	Typical	Design Limit (Note 5)	
Accuracy, LM35C (Note 7)	$T_A = +25^\circ\text{C}$	±0.4	±1.0	±0.4	±1.0	°C
	$T_A = -10^\circ\text{C}$	±0.5	±0.5	±0.5	±1.5	°C
	$T_A = T_{MAX}$	±0.8	±1.5	±0.8	±1.5	°C
	$T_A = T_{MIN}$	±0.8	±1.5	±0.8	±2.0	°C
Accuracy, LM35D (Note 7)	$T_A = +25^\circ\text{C}$			±0.8	±1.5	°C
	$T_A = T_{MAX}$			±0.8	±2.0	°C
Nonlinearity (Note 8)	$T_{MIN} < T_A < T_{MAX}$	±0.3	±0.5	±0.2	±0.5	°C
Sensor Gain (Average Slope)	$T_{MIN} < T_A < T_{MAX}$	+9.8	+9.8	+10.0	+9.8	mV/°C
		+10.2	+10.2		+10.2	
Load Regulation (Note 3) $D_{IL} < 1 \text{ mA}$	$T_A = +25^\circ\text{C}$	±0.4	±2.0	±0.4	±2.0	mV/mA
	$T_{MIN} < T_A < T_{MAX}$	±0.5	±0.5	±5.0	±0.5	±5.0
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	±0.01	±0.1	±0.01	±0.1	mV/V
	$4V < V_{CC} < 30V$	±0.02	±0.02	±0.02	±0.02	mV/V
Quiescent Current (Note 9)	$V_A = +5V, +25^\circ\text{C}$	56	80	56	80	µA
	$V_A = +5V$	105	158	91	138	µA
	$V_A = +30V, +25^\circ\text{C}$	56.2	82	50.2	82	µA
Change of Quiescent Current (Note 3)	$4V < V_{CC} < 30V, +25^\circ\text{C}$	0.2	2.0	0.2	2.0	µA
	$4V < V_{CC} < 30V$	0.5	3.0	0.5	3.0	µA
Temperature Coefficient of Quiescent Current		+0.39	+0.7	+0.39	+0.7	µA/°C
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	+1.5	+2.0	+1.5	+2.0	°C
Long Term Stability	$T_A = T_{MAX}$ for 1000 hours	±0.06		±0.06		°C

Note 1: Unless otherwise noted, these specifications apply -55°C to +150°C for the LM35 and LM35A, -40°C to +110°C for the LM35C and LM35CA, and 0°C to +100°C for the LM35D. V_{CC} is 5V and I_L is 0 µA in the circuit of Figure 1. These specifications also apply from -25°C to T_{MAX} in the circuit of Figure 2. Specifications in boldface apply over the full rated temperature range.
 Note 2: Thermal resistance of the TO-46 package is 40°C/W, junction to ambient, and 30°C/W, junction to case. Thermal resistance of the TO-92 package is 180°C/W, junction to ambient. Thermal resistance of the small outline molded package is 220°C/W, junction to ambient. Thermal resistance of the TO-220 package is 30°C/W, junction to ambient. For additional thermal resistance information see table in the Applications section.
 Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.
 Note 4: Tested limits are guaranteed and 100% tested in production.
 Note 5: Design limits are guaranteed but not 100% production tested over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality rates.
 Note 6: Specifications in boldface apply over the full rated temperature range.
 Note 7: Accuracy is defined as the error between the output voltage and 10mV/°C times the device's case temperature, at specified conditions of voltage, current, and temperature (increased to 1°C).
 Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.
 Note 9: Quiescent current is defined in the circuit of Figure 1.
 Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.
 Note 11: Current limit mode, 100 µA (discharge) through ±1.5 kΩ resistor.
 Note 12: See AN-440 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

www.national.com 4

ANEXO B: HOJA DE DATOS TÉCNICOS DEL SENSOR LDR

Data pack F

Issued March 1997 232-3816



Light dependent resistors

NORP12 RS stock number 651-507
NSL19-M51 RS stock number 596-141

Two cadmium sulphide (CdS) photoconductive cells with spectral responses similar to that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.

Guide to source illuminations

Light source	Illumination (lux)
Moonlight	0.1
60W bulb at 1m	50
1W MBS bulb at 0.1m	100
Fluorescent lighting	500
Bright sunlight	30,000

Electrical characteristics
T_A = 25°C 2854°K tungsten light source

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	100 lux	-	400	-	Ω
Cell resistance	10 lux	-	9	-	kΩ
Dark resistance	-	1.0	-	-	MΩ
Dark capacitance	-	-	35	-	pF
Rise time 1	100 lux	-	23	-	ms
Fall time 1	10 lux	-	13	-	ms
Fall time 2	100 lux	-	10	-	ms
Fall time 2	10 lux	-	10	-	ms

1: Dark to 100% R_L
2: 70:10 × R_L
R_L = photoresistor resistance under given illumination.

Features

- Wide spectral response
- Low cost
- Wide ambient temperature range

Circuit symbol



Light memory characteristics

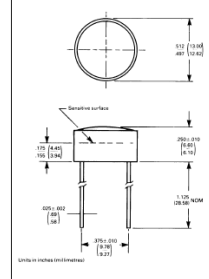
Light dependent resistors have a particular property in that they remember the lighting conditions in which they have been stored. This memory effect can be minimised by storing the LDRs in light prior to use. Light storage reduces equilibrium time to reach steady resistance values.

NORP12 (RS stock no. 651-507)

Absolute maximum ratings

Voltage, ac or dc peak	330V
Current	75mA
Power dissipation at 30°C	250mW
Operating temperature range	-60°C to +75°C

Dimensions



232-3816

Absolute maximum ratings

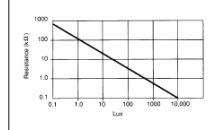
Voltage, ac or dc peak	100V
Current	5mA
Power dissipation at 25°C	50mW
Operating temperature range	-25°C to +75°C

*Derate linearly from 50mW at 25°C to 0W at 75°C.

Electrical characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	10 lux	20	100	-	Ω
Cell resistance	100 lux	-	5	-	kΩ
Dark resistance	10 lux after 10 sec	20	-	-	MΩ
Spectral response	10 lux	-	550	-	nm
Rise time	100%	-	45	-	ms
Fall time	100%	-	55	-	ms

Figure 4 Resistance as a function illumination



Dimensions

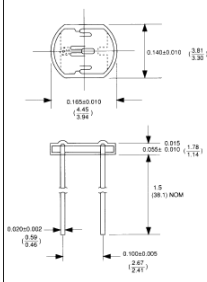
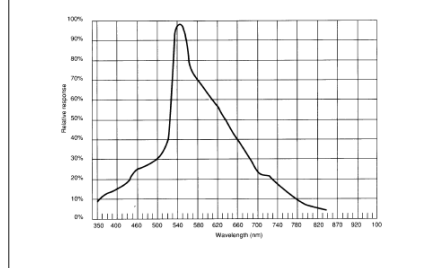


Figure 5 Spectral response



232-3816

Figure 1 Power dissipation derating

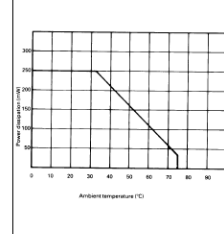


Figure 3 Resistance as a function of illumination

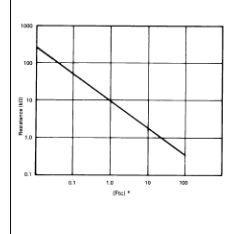
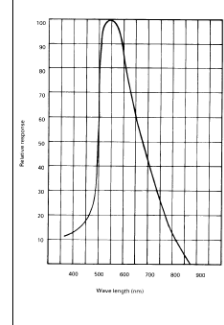


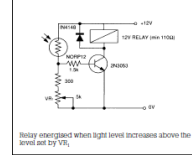
Figure 2 Spectral response



232-3816

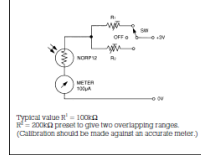
Typical application circuits

Figure 6 Sensitive light operated relay



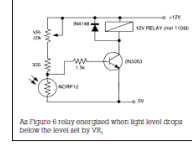
Relay energised when light level increases above the level set by V_{th}.

Figure 9 Logarithmic law photographic light meter



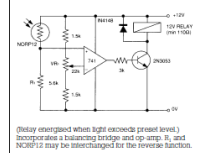
Typical values R₁ = 100kΩ
R₂ = 200kΩ preset to give two overlapping ranges.
(Calibration should be made against an accurate meter.)

Figure 7 Light interruption detector



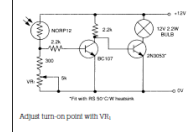
As Figure 6 relay energised when light level drops below the level set by V_{th}.

Figure 10 Extremely sensitive light operated relay



Relay energised when light exceeds preset level. Incorporates a balancing bridge and op-amp. R₁ and NORP12 may be interchanged for the reverse function.

Figure 8 Automatic light circuit



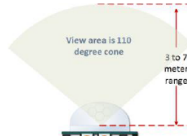
Adjust turn-on point with V_{th}.

ANEXO C: HOJA DE DATOS TÉCNICOS DEL SENSOR HC-SR501

HC-SR501 Passive Infrared (PIR) Motion Sensor

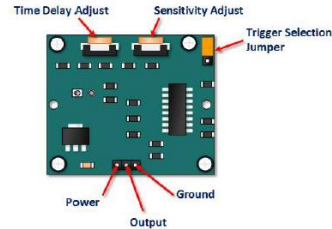


This motion sensor module uses the LM8770 Passive Infrared Sensor and the BSS000 IC to control how motion is detected. The module features adjustable sensitivity that allows for a motion detection range from 3 meters to 7 meters, also includes time delay adjustments and trigger selection that allow for fine tuning within your application.



The device will detect motion inside a 110 degree cone with a range of 3 to 7 meters.

HC-SR501 Pin Outs and Controls



Pin or Control	Function
Time Delay Adjust	Sets how long the output remains high after detecting motion... Anywhere from 3 seconds to 5 minutes.
Sensitivity Adjust	Sets the detection range... from 3 meters to 7 meters.
Trigger Selection Jumper	Set for single or repeatable triggers.
Ground pin	Ground input.
Output pin	Low when no motion is detected. High when motion is detected. High is 3.3V
Power Pin	5 to 20 VDC Supply Input.

HC-SR501 PIR Functional Description

The SR501 will detect infrared changes and if interpreted as motion, will set its output low. What is or is not interpreted as motion is largely dependent on user settings and adjustments.

Device Initialization

The device requires nearly a minute to initialize. During this period, it can and often will output false detection signals. Circuit or controller logic needs to take this initialization period into consideration.

Device Area of Detection

- During the three seconds, the PIR will not detect motion.
- After three seconds, the PIR will detect motion again and detected motion will once again set the output high and the output will remain on as dictated by the Time Delay adjustment and trigger mode selection.

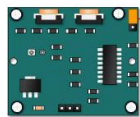
OVERRIDING THE TIME DELAY: If you're connecting your HC-SR501 to an Arduino, it is likely that you are going to take some sort of action when motion is detected. For example, you may wish to brighten lights when motion is detected and dim the lights when motion is no longer connected.

Simply delay dimming within your sketch.

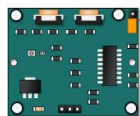
Trigger Mode Selection Jumper

The trigger mode selection jumper allows you to select between single and repeatable triggers. The affect of this jumper setting is to determine when the time delay begins.

- SINGLE TRIGGER** - The time delay begins immediately when motion is first detected.
- REPEATABLE TRIGGER** - Each detected motion resets the time delay. Thus the time delay begins with the last motion detected.



Single Trigger Mode - Time Delay is started immediately upon detecting motion. Continued detection is blocked.



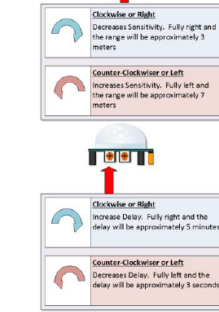
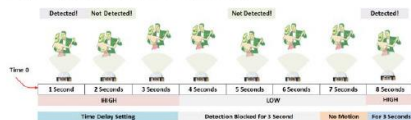
Repeatable Trigger Mode - Time Delay is re-started every time motion is detected.

HC-SR501 Dance Floor Application Examples

Imagine that you want to control lighting on a dance floor based upon where the dancers are dancing. Understanding how the time delay and trigger mode interact will be necessary to controlling that lighting in the manner that you want.

Example One

In this first example, the time delay is set to three seconds and the trigger mode is set to single. As you can see in the illustration below, the motion is not always detected. In fact, there is a period of about six seconds where motion can not be detected.



PIR Range (Sensitivity) Adjustment

As mentioned, the adjustable range is from approximately 3 to 7 meters.

Time Delay Adjustment

The time delay adjustment determines how long the output of the PIR sensor module will remain high after detection motion. The range is from about 3 seconds to five minutes.

3 Seconds Off After Time Delay Completes - IMPORTANT

The output of this device will go LOW (or OFF) for approximately 3 seconds AFTER the time delay completes. In other words, ALL motion detection is blocked during this three second period.

For Example:

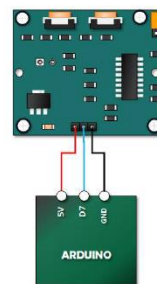
- Imagine you're in the single trigger mode (see below) and your time delay is set to 5 seconds.
 - The PIR will detect motion and set it high for 5 seconds.
 - After five seconds, the PIR will set its output low for about 3 seconds.

Example Two

In the next example, the time delay is still at three seconds and the trigger is set to repeatable. In the illustration below, you can see that the time delay period is restarted. However, after that three seconds, detection will still be blocked for three seconds. As I mentioned previously, you could override the 3 second blocking period with some creative code, but do give that consideration. Some of the electronics you use may not like an on and then off job. The three seconds allows for a little rest before starting back up.



Arduino HC-SR501 Motion Sensor Tutorial, connect Your Arduino to the HC-SR501



The sketch simply turns on Your Arduino LED connected to Pin 13 whenever motion is detected. Be sure to beware of and somehow handle the 1 minute initialization in whatever application you develop.

```
# HC-SR501 Motion Detector Sample Sketch
int ledPin = 13; // LED on Pin 13 of Arduino
int pirPin = 7; // Input for HC-SR501
int pirValue; // Place to store read PIR Value
void setup() {
  pinMode(pirPin, OUTPUT);
  pinMode(ledPin, INPUT);
  digitalWrite(ledPin, LOW);
}
void loop() {
  pirValue = digitalRead(pirPin);
  digitalWrite(ledPin, pirValue);
}
```

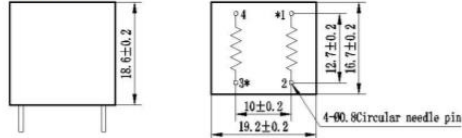
ANEXO D: HOJA DE DATOS TÉCNICOS DEL SENSOR ZMPT101B

ZMPT101B

Current-type Voltage Transformer

Small size, high accuracy, good consistency, for voltage and power measurement

Structural parameters:



Remarks: primary input: 1, 2 pins secondary output: 3, 4 pins
Or
primary input: 3, 4 pins secondary output: 1, 2 pins
*** Same polarity

Front view

Bottom view

The main technical parameters:

Model	ZMPT101B
Rated input current	2mA
Rated output current	2mA
turns ratio	1000:1000
phase angle error	≤20' (input 2mA, sampling resistor 100Ω)
operating range	0-1000V 0-10mA (sampling resistor 100Ω)
linearity	±0.2%(20%dot-120%dot)
Permissible error	-0.3%≤s+0.2% (input 2mA, sampling resistor 100Ω)
isolation voltage	4000V
application	voltage and power measurement
Encapsulation	Epoxy
installation	PCB mounting (Pin Length>3mm)
Operating temperature	-40℃~+60℃
Case Material	ABS (Note: ABS CASE is NOT available for wave-soldering)

Tel: 86-25-52601870

E-mail: zm@zeming-e.com

Direction for use:

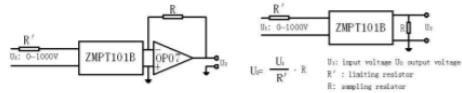


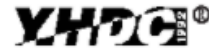
Figure I

Figure II

- The typical usage of the product is for the active output (Figure I). R' is a limiting resistor, R is a sampling resistor.
- The product can be directly through the resistance sampling, easy to use (Figure II).

ANEXO E: HOJA DE DATOS TÉCNICOS DEL SENSOR SCT

0.333V Split core current transformer



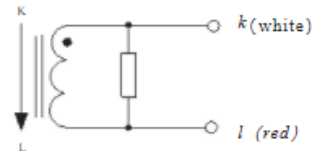
model: SCT013

Characteristics:
 Split core, 0.333V output, Built-in with sampling resistance,
 leading wire 1 metre, standard $\Phi 3.5$ three core plug output,
 (Patent no.: ZL 2015 3 0060067.X)



Technical index:
 Suspended mounting, output with cable
 Operation temperature: $-25^{\circ}\text{C} \sim +70^{\circ}\text{C}$
 Storage temperature: $-30^{\circ}\text{C} \sim +90^{\circ}\text{C}$
 work voltage: $\leq 660\text{V}$
 Work frequency: 50Hz-1KHz
 Dielectric strength: 3.5KV 50Hz 1min

Wiring schematic:



Technical Parameters table:

RatedInput(Effectivevalue)	20	30	50	80	100	A
measurement range	20%-120%					A
Rated output	0.333					V
Accuracy	1					%
Linearity	1					%
frequency	50~1K					Hz
weight	50					g

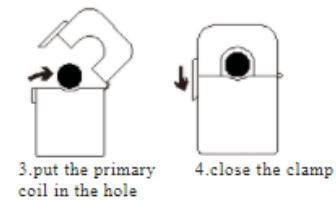
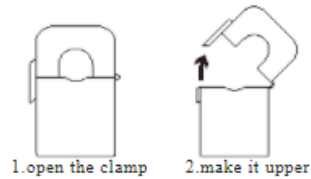
Order Format: model Input output Accuracy
 SCT013/xxA/0.333V/1%

Noted: The rated current $\leq 100\text{A}$ are the standard product

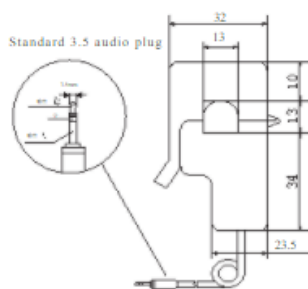
Voltage output type : Built-in with sampling resistance
 Voltage output type : not allowed secondary short circuit.
 When the plug without audio frequency,
 the white line is the end of the same name

Installation Diagram

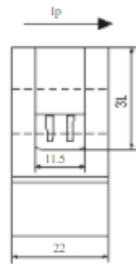
Primary coil through hole method



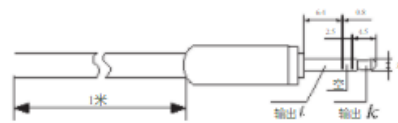
Outline size: (in:mm):



Front view



Side view



Standard three core plug schematic diagram

ANEXO F: LÍNEAS DE CÓDIGO DEL NODEMCU

```
#include "UbidotsESPMQTT.h"
#define TOKEN "....." // Your Ubidots
TOKEN
#define WIFINAME "....." //Your SSID
#define WIFIPASS "....." // Your Wifi Pass
#define DEVICE_LABEL "io"
#define VARIABLE_LABEL1 "leda" // Put here
your Ubidots variable label
#define VARIABLE_LABEL2 "ledb" // Put
here your Ubidots variable label
#define VARIABLE_LABEL3 "ledc" // Put
here your Ubidots variable label
int brillo=0;
int brillo2=0;
const int ERROR_VALUE = 65535; // Set here
an error value
const uint8_t NUMBER_OF_VARIABLES = 3;
// Number of variable to subscribe to
char *
variable_labels[NUMBER_OF_VARIABLES]
= {"leda", "ledb", "ledc"}; // labels of the variable
to subscribe to
#define out1 D0
#define out2 D2
#define out3 D4
int seguro=3;
float estadoout1; // Name of the variable to be
used within the code.
float estadoout2; // Name of the variable to be
used within the code.
float estadoout3; // Name of the variable to be
used within the code.
float value; // To store incoming value.
uint8_t variable; // To keep track of the state
machine and be able to use the switch case.
Ubidots ubiClient(TOKEN);
WiFiClient client;
void callback(char* topic, byte* payload,
unsigned int length) {
    char* variable_label = (char *)
    malloc(sizeof(char) * 30);;
    get_variable_label_topic(topic, variable_label);
    value = btof(payload, length);
    set_state(variable_label);
    execute_cases();
    free(variable_label);
    //////////////////////////////////////
    if(estadoout1==1){
        digitalWrite(out1, HIGH);
    }
    else
    {
        digitalWrite(out1, LOW);
    }
    //////////////////////////////////////
    if(estadoout2==1){
        digitalWrite(out2, HIGH);
    }
    else
    {
        digitalWrite(out2, LOW);
    }
    //////////////////////////////////////
    if(estadoout3==1){
        digitalWrite(out3, HIGH);
    }
    else
    {
        digitalWrite(out3, LOW);
    }
    //////////////////////////////////////
    // Parse topic to extract the variable label which
    changed value
    void get_variable_label_topic(char * topic, char
    * variable_label) {
        Serial.print("topic:");
        Serial.println(topic);
        sprintf(variable_label, "");
    }
}
```

```

for (int i = 0; i < NUMBER_OF_VARIABLES;
i++) {
    char * result_lv = strstr(topic,
variable_labels[i]);
    if (result_lv != NULL) {
        uint8_t len = strlen(result_lv);
        char result[100];
        uint8_t i = 0;
        for (i = 0; i < len - 3; i++) {
            result[i] = result_lv[i];
        }
        result[i] = '\0';
        Serial.print("Label is: ");
        Serial.println(result);
        sprintf(variable_label, "%s", result);
        break;
    }
}

// cast from an array of chars to float value.
float btOf(byte * payload, unsigned int length) {
    char * demo_ = (char *) malloc(sizeof(char) *
10);
    for (int i = 0; i < length; i++) {
        demo_[i] = payload[i];
    }
    return atof(demo_);
}

// State machine to use switch case
void set_state(char* variable_label) {
    variable = 0;
    for (uint8_t i = 0; i <
NUMBER_OF_VARIABLES; i++) {
        if (strcmp(variable_label, variable_labels[i])
== 0) {
            break;
        }
        variable++;
    }
}

```

```

if (variable >= NUMBER_OF_VARIABLES)
variable = ERROR_VALUE; // Not valid
}

// Function with switch case to determine which
variable changed and assigned the value
accordingly to the code variable
void execute_cases() {
    switch (variable) {
        case 0:
            estadoout1 = value;
            Serial.print("LED 1 en: ");
            Serial.println(estadoout1);
            Serial.println();
            break;
        case 1:
            estadoout2 = value;
            Serial.print("LED 2 en: ");
            Serial.println(estadoout2);
            Serial.println();
            break;
        case 2:
            estadoout3 = value;
            Serial.print("LED 3 en: ");
            Serial.println(estadoout3);
            Serial.println();
            break;
        case ERROR_VALUE:
            Serial.println("error");
            Serial.println();
            break;
        default:
            Serial.println("default");
            Serial.println();
    }
}

void setup() {
    ubiClient.ubidotsSetBroker("industrial.api.ubido
ts.com"); // Sets the broker properly for the
business account
}

```

```

    ubiClient.setDebug(true); // Pass a true or false
bool value to activate debug messages
    Serial.begin(115200);
    pinMode(out1, OUTPUT);
    pinMode(out2, OUTPUT);
    pinMode(out3, OUTPUT);
    ubiClient.wifiConnection(WIFINAME,
WIFIPASS);
    ubiClient.begin(callback);
    if(!ubiClient.connected()){
    ubiClient.reconnect();
    }
    char*          deviceStatus      =
getUbidotsDevice(DEVICE_LABEL);
    if (strcmp(deviceStatus, "404") == 0) {
ubiClient.add("leda", 0); //Insert your variable
Labels and the value to be sent
ubiClient.ubidotsPublish(DEVICE_LABEL);
ubiClient.add("ledb", 0); //Insert your variable
Labels and the value to be sent
ubiClient.ubidotsPublish(DEVICE_LABEL);
ubiClient.add("ledc", 0); //Insert your variable
Labels and the value to be sent
ubiClient.ubidotsPublish(DEVICE_LABEL);
ubiClient.loop();
    }

ubiClient.ubidotsSubscribe(DEVICE_LABEL,
VARIABLE_LABEL1); //Insert the Device and
Variable's Labels

ubiClient.ubidotsSubscribe(DEVICE_LABEL,
VARIABLE_LABEL2); //Insert the Device and
Variable's Labels

ubiClient.ubidotsSubscribe(DEVICE_LABEL,
VARIABLE_LABEL3); //Insert the Device and
Variable's Labels

    Serial.println(variable_labels[1]);
}

void loop() {
    if(!ubiClient.connected()){
    ubiClient.reconnect();

ubiClient.ubidotsSubscribe(DEVICE_LABEL,
VARIABLE_LABEL1); //Insert the Device and
Variable's Labels

ubiClient.ubidotsSubscribe(DEVICE_LABEL,
VARIABLE_LABEL2); //Insert the Device and
Variable's Labels

ubiClient.ubidotsSubscribe(DEVICE_LABEL,
VARIABLE_LABEL3); //Insert the Device and
Variable's Labels
    }
    float luz = analogRead(0);
    //float value2 = analogRead(2) // uncomment
this line to read the GPIO2 of the ESP8266
    ubiClient.add("luz", luz); //Insert your variable
Labels and the value to be sent
    ubiClient.ubidotsPublish(DEVICE_LABEL);
    ubiClient.loop();
    delay(500);
    }
char* getUbidotsDevice(char* deviceLabel) {
    char* data = (char *) malloc(sizeof(char) *
700);
    char* response = (char *) malloc(sizeof(char) *
400);
    sprintf(data, "GET /api/v1.6/devices/%s/",
deviceLabel);
    sprintf(data, "%s HTTP/1.1\r\n", data);
    sprintf(data, "Host:
industrial.api.ubidots.com\r\nUser-
Agent:io/1.0\r\n", data);
    sprintf(data, "X-Auth-Token:
%s\r\nConnection: close\r\n\r\n", data, TOKEN);

```



```

    if (client.connect("industrial.api.ubidots.com",
80)) {
client.println(data);
    } else {
return "e";
    }
    free(data);
    int timeout = 0;
    while(!client.available() && timeout < 5000) {
timeout++;
    if (timeout >= 4999){
        return "e";
    }

delay(1);
    }
    int i = 0;
    while (client.available()) {
response[i++] = (char)client.read();
    if (i >= 399){
        break;
    }
    }
    char * pch;
    char * statusCode;
    int j = 0;
    pch = strtok (response, " ");
    while (pch != NULL) {
    if(j == 1 ) {
        statusCode = pch;
    }
    pch = strtok (NULL, " ");
    j++;
    }
    free(response);
    return statusCode;
}

```

ANEXO G: LÍNEAS DE CÓDIGO DEL ESP01 INTERRUPTORES

```
#include "UbidotsESPMQTT.h"

#define TOKEN "BBFF-
vvgSpb9ZYldoAAIV5fCfqhNuwmnGb" // Your Ubidots
TOKEN
#define WIFINAME "CLARO_PAOLALP" //Your SSID
#define WIFIPASS "PAOLA0610LUIS" // Your Wifi Pass
#define DEVICE_LABEL "cuartopaola" // Name of the
device
#define VARIABLE_LABEL1 "foco" // Name of the
Ubidots variable
const int ERROR_VALUE = 65535; // Error value
const uint8_t NUMBER_OF_VARIABLES = 2; // Cantidad
de variables a las que el programa se va a suscribir
char * variable_labels[NUMBER_OF_VARIABLES] =
{"foco"}; // Variables names
#define luz 0
#define boton 2
int seguro=0;
int ledState = LOW; // the current state of the output pin
int buttonState; // the current reading from the input
pin
int lastButtonState = HIGH; // the previous reading from the
input pin
int reading;
unsigned long lastDebounceTime = 0; // the last time the
output pin was toggled
unsigned long debounceDelay = 50;

float estadoluz; // Variable to be used in the code

float value; // Variable to store input data
uint8_t variable; // To use with the switch case

Ubidots ubiClient(TOKEN);

WiFiClient client;
void callback(char* topic, byte* payload, unsigned int length)
{
char* variable_label = (char *) malloc(sizeof(char) * 30);
get_variable_label_topic(topic, variable_label);
value = btof(payload, length);
set_state(variable_label);
execute_cases();
free(variable_label);
//////////Light//////////
digitalWrite(luz, estadoluz);
//////////Light//////////
}
// Parse topic to extract the variable label which changed
value
```

```
void get_variable_label_topic(char * topic, char *
variable_label) {
Serial.print("topic:");
Serial.println(topic);
sprintf(variable_label, "");
for (int i = 0; i < NUMBER_OF_VARIABLES; i++) {
char * result_lv = strstr(topic, variable_labels[i]);
if (result_lv != NULL) {
uint8_t len = strlen(result_lv);
char result[100];
uint8_t i = 0;
for (i = 0; i < len - 3; i++) {
result[i] = result_lv[i];
}
result[i] = '\0';
Serial.print("Label is: ");
Serial.println(result);
sprintf(variable_label, "%s", result);
break;
}
}

// cast from an array of chars to float value.
float btof(byte * payload, unsigned int length) {
char * demo = (char *) malloc(sizeof(char) * 10);
for (int i = 0; i < length; i++) {
demo[i] = payload[i];
}
float value = atof(demo);
free(demo);
return value;
}

// State machine to use switch case
void set_state(char* variable_label) {
variable = 0;
for (uint8_t i = 0; i < NUMBER_OF_VARIABLES; i++) {
if (strcmp(variable_label, variable_labels[i]) == 0) {
break;
}
variable++;
}
if (variable >= NUMBER_OF_VARIABLES) variable =
ERROR_VALUE; // Not valid
}

// Function with switch case to determine which variable
changed and assigned the value accordingly to the code
variable
void execute_cases() {
```

```

switch (variable) {
  case 0:
    estadoLuz = value;
    Serial.print("Luz: ");
    Serial.println(estadoLuz);
    Serial.println();
    break;
  case ERROR_VALUE:
    Serial.println("error");
    Serial.println();
    break;
  default:
    Serial.println("default");
    Serial.println();
}

}

void setup() {

  // put your setup code here, to run once:
  pinMode(luz, OUTPUT);
  pinMode(boton, INPUT);
  ubiClient.ubidotsSetBroker("industrial.api.ubidots.com");
  // Sets the broker properly for the business account
  ubiClient.setDebug(true); // Pass a true or false bool value
  to activate debug messages
  Serial.begin(115200);
  ubiClient.wifiConnection(WIFINAME, WIFIPASS);
  ubiClient.begin(callback);
  if(!ubiClient.connected()){
    ubiClient.reconnect();
  }
  char*          deviceStatus          =
  getUbidotsDevice(DEVICE_LABEL);

  if (strcmp(deviceStatus, "404") == 0) {
    ubiClient.add("foco", 0); //Insert your variable Labels and
    the value to be sent
    ubiClient.ubidotsPublish(DEVICE_LABEL);
    ubiClient.loop();
  }

  ubiClient.ubidotsSubscribe(DEVICE_LABEL, VARIABLE
  _LABEL1); //Insert the Device and Variable's Labels
  Serial.println(variable_labels[1]);
}

void loop() {
  // put your main code here, to run repeatedly:
  if(!ubiClient.connected()){
    ubiClient.reconnect();

    ubiClient.ubidotsSubscribe(DEVICE_LABEL, VARIABLE
    _LABEL1); //Insert the Device and Variable's Labels
  }
  ubiClient.loop();

  Read();

  Debounce();

  // save the reading. Next time through the loop, it'll be the
  lastButtonState:
  lastButtonState = reading;
}

void Read(){
  // read the state of the switch into a local variable:
  reading = digitalRead(boton);
  if (reading != lastButtonState) {
    // reset the debouncing timer
    lastDebounceTime = millis();
  }
}

void Debounce(){
  if ((millis() - lastDebounceTime) > debounceDelay) {
    // whatever the reading is at, it's been there for longer than
    the debounce
    // delay, so take it as the actual current state:

    // if the button state has changed:
    if (reading != buttonState) {
      buttonState = reading;
      Toggle();
    }
  }
}

void Toggle(){
  // only toggle the LED if the new button state is LOW
  if (buttonState == LOW) {
    ledState = !ledState;
    // set the LED:
    digitalWrite(luz, ledState);
    ubiClient.add("foco", ledState); //Insert your variable
    Labels and the value to be sent
    ubiClient.ubidotsPublish(DEVICE_LABEL);
  }
}

```

```

char* getUbidotsDevice(char* deviceLabel) {
char* data = (char *) malloc(sizeof(char) * 700);
char* response = (char *) malloc(sizeof(char) * 400);
sprintf(data, "GET /api/v1.6/devices/%s/", deviceLabel);
sprintf(data, "%s HTTP/1.1\r\n", data);
sprintf(data, "%sHost: industrial.api.ubidots.com\r\nUser-
Agent:wifiswitch/1.0\r\n", data);
sprintf(data, "%sX-Auth-Token: %s\r\nConnection:
close\r\n\r\n", data, TOKEN);
char* data1 = data;
free(data);
if (client.connect("industrial.api.ubidots.com", 80)) {
client.println(data1);
}
else {
free(data);
return "e";
}
int timeout = 0;
while(!client.available() && timeout < 5000) {
timeout++;
if (timeout >= 4999){
free(data);
return "e";
}
delay(1);
}

int i = 0;
while (client.available()) {
response[i++] = (char)client.read();
if (i >= 399){
break;
}
}
char * pch;
char * statusCode;
int j = 0;
pch = strtok (response, " ");
while (pch != NULL) {
if (j == 1) {
statusCode = pch;
}
pch = strtok (NULL, " ");
j++;
}
free(response);
return statusCode;
}

```

ANEXO H: LÍNEAS DE CÓDIGO DEL ESP01 MEDIDOR DE CONSUMO

```

#ifdef U8X8_HAVE_HW_SPI
#include <SPI.h>
#endif
#ifdef U8X8_HAVE_HW_I2C
#include <Wire.h>
#endif
////////////////////
#include "Ubidots.h"
const char* UBIDOTS_TOKEN = "BBFF-
vvgSpb9ZY1doAAIV5fCfqhNuwmnGb"; //
Aquí pones tu TOKEN de Ubidots
const char* DEVICE_LABEL = "casapaola";
// Aquí pones el nombre del dispositivo
float volt, irms, power, pf, kwh, temp;
String volt1, irms1, power1, pf1, kwh1, temp1;
int cont = 0;

U8G2_SH1106_128X64_NONAME_F_HW_I2
C    u8g2(U8G2_R1, /* reset=*/
U8X8_PIN_NONE, 0, 2);
Ubidots    ubidots(UBIDOTS_TOKEN,
UBI_UDP);
WiFiManager wifiManager;

void configModeCallback (WiFiManager
*myWiFiManager) {
    Serial.println("Entered config mode");
    Serial.println(WiFi.softAPIP());
    //if you used auto generated SSID, print it
    Serial.println(myWiFiManager-
>getConfigPortalSSID());
}

String getValue(String data, char separator, int
index)
{
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;

    for (int i = 0; i <= maxIndex && found <=
index; i++) {
        if (data.charAt(i) == separator || i ==
maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i + 1 : i;
        }
    }
    return    found    >    index    ?
data.substring(strIndex[0], strIndex[1]) : "";
}

// Funcion que presenta los datos en pantalla
void pantalla() {
    u8g2.clearBuffer(); // clear the internal
memory
    u8g2.setFont(u8g2_font_6x13_te); // choose a
suitable font
    u8g2.setCursor(0, 13);
    u8g2.print(volt1 + " V");
    u8g2.setCursor(0, 28);
    u8g2.print(irms1 + " A");
    u8g2.setCursor(0, 43);
    u8g2.print(power1 + " W");
    u8g2.setCursor(0, 58);
    u8g2.print(pf1);
    u8g2.setCursor(0, 73);
    u8g2.print(temp1 + " C");
    if (WiFi.status() == WL_CONNECTED) {
        u8g2.setCursor(16, 118);
        u8g2.print("WiFi");
    }
    u8g2.sendBuffer(); // transfer internal
memory to the display
}

void setup() {
    Serial.begin(115200);

```

```

wifiManager.setAPCallback(configModeCallba
ck);

if (!wifiManager.autoConnect()) {
  Serial.println("failed to connect and hit
timeout");
  //reset and try again, or maybe put it to deep
sleep
  ESP.reset();
  delay(1000);
}
Serial.println("connected...yeey :)");

u8g2.begin();
u8g2.setFlipMode(2);
}

void loop() {

if (WiFi.status() == !WL_CONNECTED) {
  wifiManager.autoConnect();
}
// put your main code here, to run repeatedly:
if (Serial.available() > 0) {

  String texto = Serial.readString();
  volt1 = getValue(texto, '!', 0);
  Serial.println(volt1 + " V");
  irms1 = getValue(texto, '!', 1);
  Serial.println(irms1 + " A");
  power1 = getValue(texto, '!', 2);
  Serial.println(power1 + " W");
  pf1 = getValue(texto, '!', 3);
  Serial.println(pf1 + " PF");
  kwh1 = getValue(texto, '!', 4);
  Serial.println(kwh1 + " Kw/h");
  temp1 = getValue(texto, '!', 5);

  Serial.println(temp1 + " C");

  float volt = (volt1).toFloat();
  float irms = (irms1).toFloat();
  float power = (power1).toFloat();
  float pf = (pf1).toFloat();
  float kwh = (kwh1).toFloat();
  float temp = (temp1).toFloat();

  pantalla();

  ubidots.add("voltios", volt); // Change for your
variable name
  ubidots.add("irms", irms); // Change for your
variable name
  ubidots.add("potencia", power); // Change for
your variable name
  ubidots.add("pf", pf); // Change for your
variable name
  ubidots.add("kwh", kwh); // Change for your
variable name
  ubidots.add("temp", temp); // Change for your
variable name

  bool bufferSent = false;
  bufferSent =
ubidots.send(DEVICE_LABEL); // Will send
data to a device label that matches the device Id
if (bufferSent) {
  // Do something if values were sent properly
  Serial.println("Values sent by the device");
}
}
}

```

ANEXO I: LÍNEAS DE CÓDIGO DE ARDUINO NANO

```
#include "EmonLib.h"
EnergyMonitor emon1; // Instancia de la
librería EmonLib
#define sensor A3 // Pin del sensor de
temperatura
float
realPower,apparentPower,powerFactor,supplyV
oltage,Irms,kwh; // Variables que almacenan
los datos procesados por la librería.
int temp; // Variable que guarda el
valor del sensor de temperatura.
// variables para el Delay sin Delay.
unsigned long previousMillis = 0;
const long interval = 1000; //Se envía la
información cada 1000ms

float volt;
void setup()
{
  Serial.begin(115200);
  pinMode(sensor,INPUT);
  emon1.voltage(A0, 119, 1.7); // Voltaje: Pin
de entrada, calibración, cambio de fase
  emon1.current(A1, 10.6); // Corriente: Pin
de entrada, calibración.}
void loop(){
  unsigned long currentMillis = millis();
  emon1.calcVI(20,1000); // Calcular media
longitud de onda en base a 1000ms (timeout).
  realPower = emon1.realPower;
//Guardo la Potencia real en una variable
  realPower = realPower+167.75;
  apparentPower = emon1.apparentPower;
//Guardo la Potencia aparente en una variable
  powerFactor = emon1.powerFactor;
//Guardo el factor de potencia en una variable
  powerFactor = powerFactor+1.67;
  supplyVoltage = emon1.Vrms;
//Guardo el voltaje en una variable

  Irms = emon1.Irms; //Guardo la
corriente en una variable
  Irms = Irms+0.88;
  kwh = (realPower/3600)/1000;
//Calculo la potencia instantanea consumida.
KW/s (En una hora hay 3600 seg. y divido entre
1000 para pasar de W a KW.

  temp = analogRead(sensor)*(4.5/1024)*100;
//El sensor de temperatura tiene esta relación de
salida frente a los Kelvins de salida. (4.653 =
VCC).
  temp = temp+2; //Convierto a
Centígrados

  if (currentMillis - previousMillis >= interval)
  {
    previousMillis = currentMillis;
    //Todo lo que esté aquí dentro, se repite cada
"interval" ms.
    //Envío por el monitor Serial cada valor
medido, seguido de un "!",
    Serial.print(supplyVoltage);
    Serial.print("!");
    Serial.print(Irms);
    Serial.print("!");
    Serial.print(realPower);
    Serial.print("!");
    Serial.print(powerFactor);
    Serial.print("!");
    Serial.print(kwh,8);
    Serial.print("!");
    Serial.print(temp);
    Serial.print("!");
    Serial.println(" "); //Al final envío un " " para
definir el final de los datos enviados.
  }
}
```

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL APRENDIZAJE

UNIDAD DE PROCESOS TÉCNICOS

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 09 / 03 / 2022

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: Luis Marcelo Pillajo Colcha
INFORMACIÓN INSTITUCIONAL
Facultad: Facultad de Informática y Electrónica
Carrera: Ingeniería en Electrónica y Automatización
Título a optar: Ingeniero en Electrónica y Automatización
f. Analista de bibliotecas responsable:



epoch

**Dirección de Bibliotecas y
Recursos del Aprendizaje**

**UNIDAD DE PROCESOS TÉCNICOS Y ANÁLISIS BIBLIOGRÁFICO Y
DOCUMENTAL**

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 03 / 05 / 2022

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: LUIS MARCELO PILLAJO COLCHA
INFORMACIÓN INSTITUCIONAL
Facultad: INFORMÁTICA Y ELECTRÓNICA
Carrera: ELECTRÓNICA Y AUTOMATIZACIÓN
Título a optar: INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN
f. Analista de Biblioteca responsable: Lcdo. Holger Ramos, MSc.

0498-DBRA-UPT-2022