UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL

CARRERA: INGENIERIA EN ELECTRÓNICA Y AUTOMATIZACIÓN

Diseño de Trabajo de Titulación

TITULO: Desarrollo de un banco de pruebas utilizando el chip ESP-32 y el asistente virtual Alexa para el laboratorio de domótica.

AUTORES:

Alexis Paúl Loor Torres

Hugo Xavier Alvarez Saltos

TUTOR:

Ing. Luis Antonio Neira Clemente, MSc.

FECHA:

Guayaquil-Ecuador

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE ELECTRONICA Y AUTOMATIZACION

Desarrollo de un banco de pruebas utilizando el chip ESP-32 y el asistente virtual Alexa para el laboratorio de domótica.

Trabajo de titulación previo a la obtención del

Título de Ingeniero Electrónico

AUTORES: Hugo Xavier Alvarez Saltos

Alexis Paúl Loor Torres

TUTOR: Ing. Luis Antonio Neira Clemente, MSc.

Guayaquil-Ecuador
2022

Resolución CS. N°166-09-2021-07-21

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Hugo Xavier Alvarez Saltos con documento de identificación N°1804376240 y Alexis Paúl Loor Torres con documento de identificación N°0950442251; manifestamos que:
Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar demanera total o parcial el presente trabajo de titulación.

Guayaquil, 20 de julio del año 2022

Atentamente,

Hugo Xavier Alvarez Saltos	Alexis Paúl Loor Torres
1804376240	0950442251

Resolución CS. N°166-09-2021-07-21

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Hugo Xavier Alvarez Saltos con documento de identificación No. 1804376240 y Alexis Paúl Loor Torres con documento de identificación No. 0950442251, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico:“ Desarrollo de un banco de pruebas utilizando el chip ESP-32 y el asistente virtual Alexa para el laboratorio de domótica.“, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 20 de julio del año 2022

Atentamente,

Hugo Xavier Alvarez Saltos 1804376240	Alexis Paúl Loor Torres 0950442251

Resolución CS. N°166-09-2021-07-21

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Luis Antonio Neira Clemente, con documento de identificación N°0909136582, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: Desarrollo de un banco de pruebas utilizando el chip ESP-32 y el asistente virtual Alexa para el laboratorio de domótica, realizado por Hugo Xavier Alvarez Saltos, con documento de identificación N°1804376240 y por Alexis Paúl Loor Torres con documento de identificación N°0950442251 , obteniendo como resultado final el trabajo de titulación bajo la opción que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.
2 de septiembre del año 2022

Atentamente,

Ing. Luis Antonio Neira Clemente MSc.
0909136582

Resolución CS. N°166-09-2021-07-21

DEDICATORIA

Dedico esta tesis a mis padres Hugo Humberto Alvarez García y Doris Del Consuelo Saltos Del Salto, quienes desde la distancia siempre nos apoyaron y estuvieron pendientes de mi hermano y de mí, a mi hermano Alexis David Alvarez Saltos al que quiero mucho y que aunque no seamos tan cercanos él sabe que lo daría todo para que sea feliz, a mi querida abuelita Noemí Angélica Del Salto Gaibor que con su amor y ternura siempre estuvo para mí, a mis tías, primas y a mi novia ya que siempre estuvieron apoyándome y nunca dejaron de creer en mí, me dieron todo el apoyo que necesitaba para poder llegar a ser un profesional de bien y especialmente a mi tía, Florita Isabel Saltos Del Salto que con su amor, su entereza y carisma, cuidó de mi desde pequeño y aún lo sigue haciendo como una madre abnegada por su hijo y que siempre tendrá mi más eterno agradecimiento.

Hugo Xavier Alvarez Saltos

AGRADECIMIENTO

En primer lugar, doy gracias a Dios, porque creo y doy fe de su poder actuando en mi vida, cada día, cuidando y protegiéndome no solo a mí, sino que sé que él escucha mis oraciones y cuida de toda mi familia y los protege de todo mal, tengo la absoluta seguridad de que él como padre siempre cuida de mí.

Luego agradezco a mis padres que con sus cuidados fueron parte de mi desarrollo y despertaron en mí las ganas de aprender y analizar el porqué de las cosas, forjando un hombre que vela por los demás, con pericia y calidad humana. Me han dado el amor y me mostraron la ruta que debo seguir para ser una persona de bien como ellos lo son.

Agradezco a mi novia Karla Ximena Villamarín Sánchez, que siempre fue paciente, creyó en mis capacidades y estuvo acompañándome en el transcurso de mi carrera.

Agradezco también a mis amigos Kevin Alvarado y Fernanda Delgado que fueron un pilar en momentos en los que quería desfallecer y supieron como animarme a seguir adelante.

Agradezco a mi tutor MSC. Luis Antonio Neira Clemente que estuvo siempre pendiente apoyándonos en la investigación y desarrollo de nuestra tesis.

Agradezco a mi amigo y compañero de tesis Alexis Loor, ya que desde que empezamos a congeniar supe que seríamos un gran equipo y no me equivoqué, Dios pone a las personas correctas en mi camino.

Hugo Xavier Alvarez Saltos

DEDICATORIA

Dedico este agradecimiento a mis padres que siempre me apoyaron a pesar de todo y siempre estuvieron para mi cuando más los necesite, ellos fueron la razón por la que pude terminar mis estudios, gracias a su sacrifico y la fe que tuvieron en mi e llegado a donde estoy, espero poder pagar todo lo que hicieron por mi y este es el primer paso para poder hacerlo, también agradezco al resto de mi familia que también fueron parte de este proceso y siempre se preocuparon por mis estudios y también me brindaron su apoyo.

Alexis Paúl Loor Torres

AGRADECIMIENTO

Agradezco a mis padres ellos fueron el motor que me permitió seguir adelante, que a pesar de mis fallas siempre creyeron en mí y de lo que soy capaz de lograr que a pesar de las difíciles situaciones siempre supieron estar ahí para mí.

Agradezco también a mi amigo Xavier Alvarez por ser parte de este proceso y poder cumplir lo que ambos nos propusimos.

Alexis Paúl Loor Torres

I. RESUMEN

La domótica en la actualidad es un campo en constante crecimiento debido a las comodidades que esta brinda no solo en el área residencial si no también en el área industrial, debido a la comunidad que esta brinda ya sea con luces que se encienden al detectar una persona o que se enciendan en una hora especifica.

En el ámbito de la seguridad la domótica permite, detectar fugas de gases, incendios e intrusos y alertar al usuario de lo que pasa en tiempo real, así también la domótica nos permite controlar diferentes factores como lo son las variables de temperatura y humedad de una habitación o un proceso.

El objetivo del trabajo es darle una herramienta a los estudiantes para que estos se familiaricen con la domótica y como esta se puede implementar, mediante un banco de pruebas en la que los estudiantes podrán aprender con las problemáticas más comunes que suelen presentar en la domótica, como lo es el control de temperaturas, el encendido de luces, la supervisión de un proceso como lo es la temperatura de un calentador de agua.

Para esto el banco de pruebas lleva SOC Esp-32 que fue escogido debido a que posee conexión WIFI por lo cual el proceso domótico se podrá controlar mediante una conexión WIFI, sumado a esto para controlar al ESP-32 se empleó el asistente virtual ALEXA de Amazon el cual es muy utilizado en la domótica.

Para concluir el presente trabajo de titulación va encaminado a mejorar la curva de aprendizaje de los estudiantes de la materia de domótica, mediante el banco de pruebas en el cual se ver beneficiados al poder realizar prácticas en este.

PALABRAS CLAVES:

WIFI, ESP-32, Asistente virtual ALEXA, Domótica

ABSTRACT

Home automation is currently a field in constant growth due to the comforts that it provides not only in the residential area but also in the industrial area, due to the community that it provides, either with lights that turn on when a person is detected. or turn on at a specific time.
In the field of security, home automation allows us to detect gas leaks, fires and intruders and alert the user of what is happening in real time, as well as home automation allows us to control different factors such as the temperature and humidity variables of a room or a process.
The objective of the work is to give a tool to the students so that they become familiar with home automation and how it can be implemented, through a test bank in which students can learn with the most common problems that they usually present in home automation, such as temperature control, lighting, monitoring of a process such as the temperature of a water heater.

For this, the test bench has SOC Esp-32, which was chosen because it has a WIFI connection, so the home automation process can be controlled through a WIFI connection, added to this, to control the ESP-32, the virtual assistant ALEXA of Amazon which is widely used in home automation.
To conclude this titling work, it is aimed at improving the learning curve of students in the subject of home automation, through the test bench in which they will benefit from being able to carry out practices in it.

II. INTRODUCCION

La importancia del proyecto consiste en que al diseñar e implementar un banco de pruebas basado en el ESP-32 se está brindando a los estudiantes la posibilidad de conocer una herramienta diferente para implementar un proyecto de Domótica. Se empleó el ESP-32 debido a su pequeño tamaño permite ser empleado en lugares con espacios reducidos y de su fácil programación basado en el IDE de Arduino.

El ESP-32 cuenta con bastantes herramientas a su disposición, pero la más importante es que permite una conexión a internet, lo que para proyectos de domótica nos permite realizar distintas tareas, desde cualquier lugar siempre y cuando contemos con conexión a internet inalámbrico.

Otra ventaja que ofrece el proyecto es la implementación de los denominados asistentes virtuales en este caso el asistente de Amazon ALEXA el cual, por medio de comando de voz, se envían órdenes al ESP-32 mediante lo cual se pueden desarrollar aplicaciones muy prácticas como es el monitoreo de temperaturas, humedad, distancia, entre otras.

La razón por la que se empleó el Asistente virtual Alexa es porque este posee una compatibilidad con más accesorios que otros asistentes virtuales como se lo observa en la siguiente imagen.

Esta compatibilidad con un gran abanico de sensores es lo que hace que el asistente virtual Alexa sea uno de los mejores asistentes virtuales de mercado y uno de los preferidos a la hora de usarlo en los hogares.

El banco de pruebas está diseñado para que los estudiantes puedan realizar prácticas en las cuales por medio del ESP-32 puedan captar datos y que el asistente virtual le informe dichos datos cuando el usuario le solicite, a su vez, el asistente virtual permitirá dar órdenes al ESP-32 en el caso que deseemos activar elementos de salida.

III. CAPÍTULO UNO: DESCRIPCIÓN DEL PROBLEMA

1. ANTECEDENTES

Los laboratorios de la Universidad Politécnica Salesiana no cuentan con módulos destinados a la domótica. Por tal motivo se implementó un banco de pruebas basado en el SOC (system on chip) ESP-32, el cual debido a sus características ha demostrado tener funciones que pueden ser aplicadas al campo de la domótica.

Dentro de las aplicaciones que se realizaron con el ESP-32 se encuentra el monitoreo de la temperatura y humedad de una habitación, así como el control de luces en la casa, este monitoreo de temperatura y control de luces en una habitación son problemas que se suelen presentar en los proyectos domóticos, así como el, monitoreo de temperatura de un calentador de agua, y el control de sensores de presencia que mejoraran la seguridad del hogar.

A pesar de las grandes ventajas que ofrece el ESP-32 muchos estudiantes no están familiarizados con él y de las aplicaciones que este posee en el campo de la domótica en el cual siempre se ha buscado adaptar las nuevas tecnologías, como son los asistentes virtuales.

2. IMPORTANCIA Y ALCANCE

En la actualidad, los bancos de pruebas con ESP-32 controlados por la asistente Alexa no se encuentran dentro de la Universidad.

Los estudiantes necesitan más conocimientos extracurriculares que aporten a su desarrollo profesional con tecnologías que se encuentran disponibles y en apogeo como lo son los asistentes virtuales que, facilitan la vida de los usuarios aplicando la electrónica además de su alcance interconectado con IoT desde el que podremos utilizar las funciones del sistema desde cualquier parte del mundo por medio de una conexión a internet. Es por ello que se vio la necesidad de desarrollar un banco de pruebas con el módulo ESP-32 y la asistente Alexa para uso del plantel estudiantil.

3. JUSTIFICACION

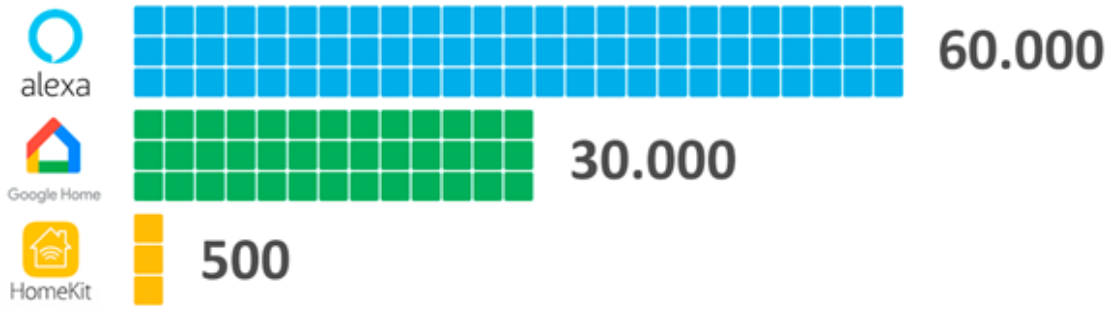
La importancia del proyecto consiste en que al diseñar e implementar un banco de pruebas basado en el ESP-32 se está brindando a los estudiantes la posibilidad de conocer una herramienta diferente para implementar un proyecto de Domótica. Se empleará el ESP-32 debido a su pequeño tamaño permite ser empleado en lugares con espacios reducidos y de su fácil programación basado en el IDE de Arduino.

ESP-32 cuenta con bastantes herramientas a su disposición, pero la más importante es que permite una conexión a internet, lo que para proyectos de domótica nos permitiría realizar distintas tareas, desde cualquier lugar siempre y cuando contemos con conexión Wi-Fi.

Otra ventaja que ofrece el proyecto es la implementación de los denominados asistentes virtuales en este caso el asistente de Amazon ALEXA el cual, por medio de comando de voz, se pueden enviar ordenes al ESP-32 mediante lo cual se pueden desarrollar aplicaciones muy prácticas como es el monitoreo de temperaturas, humedad, distancia, entre otras.

La razón por la que se empleara el Asistente virtual Alexa es porque este posee una compatibilidad con más accesorios que otros asistentes virtuales como se lo observa en la siguiente imagen.

Figura 1: Compatibilidad con accesorios domóticos



4. DELIMITACIÓN

4.1 DELIMITACIÓN TEMPORAL

El proyecto se diseña y desarrolla en la ciudad de Guayaquil también conocida como la Perla del Pacífico del país Ecuador en el periodo académico de abril a septiembre de 2022.

4.2 DELIMITACIÓN ESPACIAL

El proyecto se desarrolló para los laboratorios de electrónica y automatización de la Universidad Politécnica Salesiana sede Guayaquil, campus Centenario.

4.3 DELIMITACIÓN ACADÉMICA

Se espera que esta implementación brinde a los estudiantes de la carrera de electrónica y automatización de la Universidad Politécnica Salesiana de Guayaquil herramientas de aprendizaje para el diseño y manejo de bancos de pruebas utilizando módulos ESP-32 en combinación con asistentes virtuales. Alexa

OBJETIVOS GENERALES Y ESPECÍFICO

A continuación, se presentan los objetivos de este trabajo de titulación, los cuales se determinaron para la investigación de una problemática y que a su vez busca comprobar cada uno de ellos como parte fundamental y experimentación de este proceso.

4.4 OBJETIVO GENERAL

Desarrollar un banco de pruebas utilizando el SOC ESP-32 y el asistente virtual Alexa para el laboratorio de domótica.

4.5 OBJETIVOS ESPECÍFICOS

- Investigar sobre los sistemas domóticos usando el ESP-32.
- Utilizar el asistente virtual Alexa para la interacción con elementos tales como sensor de temperatura, humedad y proximidad.
- Implementar el control de elementos de salida tales como luces, motores y calentador.
- Diseñar e implementar la estructura del banco de pruebas.

5. Revisión de la literatura o fundamentos teóricos.

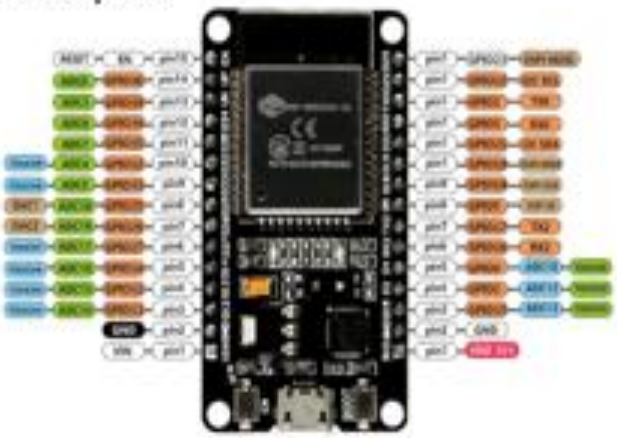
ESP-32:

El microcontrolador ESP-32 es un SoC (System on Chip) desarrollado por la empresa china Espressif y fabricado por TSMC. Integra un procesador Tensilica Xtensa de doble núcleo de 32 bits a 160 MHz (potencialmente hasta 240 MHz) en un solo chip con conectividad Wi-Fi y Bluetooth. (Guerra, 2020)

El módulo ESP32 es un paquete completo de Wi-Fi/Bluetooth certificado para proporcionar un procesador integrado con radio inalámbrica y conectividad entre varios elementos. El procesador posee 2 núcleos de procesamiento de datos cuya frecuencia de funcionamiento se puede controlar libremente entre los 80 (MHz) y 240 (MHz). Los elementos que

se pueden conectar al procesador permiten una fácil realización de tareas.
Además, en combinación con IOT (Internet of Things), permite una amplia gama de aplicaciones de automatización del hogar. Esta conexión a internet te permite conectarte al asistente virtual Alexa, que se encarga de enviar datos al ESP-32.

Figura 2: Pines del esp-32



Adaptado de:

https://www.formacionprofesional.info/los-modulos-esp32-y-lora-mas-vendidos/

Se decidió usar el SOC ESP-32 Debido a que este posee un tamaño reducido y una gran velocidad de procesamiento, así como una mayor estabilidad de conexión a internet que su predecesor el ESP-8266 como se lo explica en la siguiente tabla.

Figura 3: Tabla comparativa entre el ESP-32 y el ESP-8266

Características	ESP8266	ESP32
Procesador	Tensilica LX106 32 bit a 80Mhz (hasta 160MHz)	Tensilica Xtensa LX6 32 bit Dual Core a 160Mhz (hasta 240MHz)
Memoria RAM	80 KB (40 kB disponibles)	520 KB
Memoria Flash	Hasta 4MB	Hasta 16MB
ROM	No	448 KB
Alimentación	3.0v a 3.6v	2.2v a 3.6v
Rango temperaturas	-40ºC a 125ºC	-40ºC a 125ºC
Consumo de corriente	80 mA (promedio) 225 mA máximo	80 mA (promedio) 225 mA máximo
Consumo en deep-sleep	20uA (RTC + memoria RTC)	2.5uA (RTC +memoria RTC)
Coprocesador ULP	No	Si. Consumo inferior a 150uA
WiFi	802.11 b/g/n (hasta +20dBm) WEP, WPA	802.11 b/g/n (hasta +20dBm) WEP, WPA
Soft-AP	Si	Si
Bluetooth	No	v4.2 BR/EDR y BLE
UART	2 puertos	3 puertos
I2C	1 interfaz	2 interfaces
SPI	2 interfaces	4 interfaces
GPIO (utilizables)	32 pines	11 pines
PWM	8 canales	16 canales
ADC	1(10bit)	2 (Hasta 18 canales) (12 bit)
ADC preamplificador	No	Si. Bajo ruido hasta 60dB
DAC	No	2 canales (8bit)
1-wire	Vía software	Vía software
I2S	1 interfaz	2 interfaces
CAN 2.0	No	1 bus
Ethernet	No	10/100Mbps MAC
Sensor temperatura	No	Si
Sensor efecto hall	No	Si
Infrarrojos	No	Si
Timers	3	4(64bits)
Encriptación	No (TLS por software)	Si (AES, SHA, RSA, ECC)
RNG	No	Si
Encriptación flash	No	Si
Secure Boot	No	Si

Adaptado de:
https://ebuah.uah.es/dspace/bitstream/handle/10017/35420/TFG_Benito_Herranz_2019.pdf?sequence=1&isAllowed=y

Figura 3: Tabla comparativa entre el ESP-32 y el ESP-8266

Características	ESP8266	ESP32
Procesador	Tensilica LX106 32 bit a 80Mhz (hasta 160MHz)	Tensilica Xtensa LX6 32 bit Dual Core a 160Mhz (hasta 240MHz)
Memoria RAM	80 KB (40 kB disponibles)	520 KB
Memoria Flash	Hasta 4MB	Hasta 16MB
ROM	No	448 KB
Alimentación	3.0v a 3.6v	2.2v a 3.6v
Rango temperaturas	-40ºC a 125ºC	-40ºC a 125ºC
Consumo de corriente	80 mA (promedio) 225 mA máximo	80 mA (promedio) 225 mA máximo
Consumo en deep-sleep	20uA (RTC + memoria RTC)	2.5uA (RTC +memoria RTC)
Coprocesador ULP	No	Si. Consumo inferior a 150uA
WiFi	802.11 b/g/n (hasta +20dBm) WEP, WPA	802.11 b/g/n (hasta +20dBm) WEP, WPA
Soft-AP	Si	Si
Bluetooth	No	v4.2 BR/EDR y BLE
UART	2 puertos	3 puertos
I2C	1 interfaz	2 interfaces
SPI	2 interfaces	4 interfaces
GPIO (utilizables)	32 pines	11 pines
PWM	8 canales	16 canales
ADC	1(10bit)	2 (Hasta 18 canales) (12 bit)
ADC preamplificador	No	Si. Bajo ruido hasta 60dB
DAC	No	2 canales (8bit)
1-wire	Vía software	Vía software
I2S	1 interfaz	2 interfaces
CAN 2.0	No	1 bus
Ethernet	No	10/100Mbps MAC
Sensor temperatura	No	Si
Sensor efecto hall	No	Si
Infrarrojos	No	Si
Timers	3	4(64bits)
Encriptación	No (TLS por software)	Si (AES, SHA, RSA, ECC)
RNG	No	Si
Encriptación flash	No	Si
Secure Boot	No	Si

Adaptado de:
https://ebuah.uah.es/dspace/bitstream/handle/10017/35420/TFG_Benito_Herranz_2019.pdf?sequence=1&isAllowed=y

DISPOSITIVO ALEXA:

Alexa es un ayudante virtual el cual funciona por medio de comando de voz para varios altavoces y teléfonos, desarrollados por la empresa Amazon y lanzado en noviembre. Su nombre fue elegido después de una investigación donde se descubrió que una consonante fuerte era mas fácil de que el ayudante virtual la detectara. Esto facilita que los asistentes reconozcan su nombre y es un homenaje a la Biblioteca de Alejandría (Amazon, 2018).

Alexa funciona de manera muy similar a algunos asistentes que se encuentran en el mercado como lo son el Google Assistant, Siri, Cortana. Estos están disponibles a través de aplicaciones en dispositivos de terceros, como parlantes, televisores y teléfonos, además de usar su propio dispositivo Echo. Di su nombre primero y llámalo. En ese momento, el altavoz o dispositivo en el que está integrado comenzará a escuchar y emitirá otro color. Luego tienes que decir el comando por voz. El asistente reconoce lo que estás preguntando y responde. Su asistente virtual siempre escuchará hasta que escuche su nombre. El asistente está entonces listo para recibir órdenes. Los dispositivos Alexa te permiten controlar una amplia variedad de periféricos diseñados por Amazon, no necesariamente de la línea Amazon, a través de ESP-32. Alexa también tiene la capacidad de recordar estos dispositivos, agruparlos por usuario y permitir o habilitar todos los sensores o uno a la vez.

La funcionalidad de Alexa se basa en habilidades que puedes desarrollar, y esta programación se basa en los dos lenguajes más utilizados: JavaScript o Python. El entorno de desarrollo de Alexa se adapta al idioma que elija el usuario de los dos idiomas anteriores.

Aunque los dispositivos Alexa tienen algunas configuraciones predeterminadas que te permiten configurar tu tono de voz y seguir una serie de pasos para configurar los comandos de fábrica, puedes usar la aplicación Alexa para crear tus propias rutinas y comandos que se pueden configurar y crear. Usa tu dispositivo de una manera más personalizada.

Figura 4: Amazon Echo Dot 3generacion

Adaptado:https://www.amazon.com/-/es/Echo-Dot-3ra-Gen-inteligente/dp/B07FZ8S74R

IOT:

Internet de las cosas es un proceso electrónico que permite la conexión de equipos de uso diario con internet, es tan amplio el campo de trabajo que puede ser utilizado en focos de luz, electrodomésticos, bombas de agua, casas inteligentes y hasta en áreas de medicina.

Gracias a varios dispositivos, hoy en día podemos acoplar casi cualquier dispositivo eléctrico y electrónico para que por medio del internet nos envíen datos en tiempo real y pueda ser manipulado su uso.

El IOT es lo que permite controlar por medio del asistente virtual a los sensores y salidas del ESP-32, otra ventaja de trabajar con el iot es que tiene la capacidad de mostrar los datos de los sensores en una página de internet o una aplicación así no tengamos el asistente virtual a mano podremos seguir controlando nuestro hogar.

PROTOCOLO HTTP:

"Protocolo de transferencia de hipertexto" Este es un protocolo diseñado en base al principio cliente-servidor que le permite enviar solicitudes de datos y recursos es el principio de todo el intercambio de información que se procesa en la web. En otras palabras, las solicitudes de datos son iniciadas por elementos. Recibir datos (cliente). Por lo general, un navegador web.

Este es el protocolo utilizado por el ESP-32 para comunicarse con un servidor web que puede mostrar datos para cargarlos en la nube.

COMUNICACIÓN WIFI:

Por las siglas en inglés Wireless o Wireless, es un modo de comunicación que elimina los métodos físicos de propagación y utiliza ondas electromagnéticas para permitir que los dispositivos se comuniquen de extremo a extremo.

SINRIC PRO:

Sinric pro es una aplicación seleccionada que permite que el ESP-32 se conecte a dispositivos Alexa a través de una habilidad desarrollada por la misma aplicación. Su funcionamiento consiste en habilitar una conexión al ESP generando un usuario y contraseña. 32 se fabrica a través de la nube, dando a cada usuario una clave única como modo adicional de seguridad. Esta aplicación no solo se limita a Alexa Assistant, sino que también se puede utilizar con otro tipo de asistentes virtuales como Google Assistant.

El uso que se le dio al Sinric Pro fue como un enlace entre el asistente virtual y el ESP-32 se le escogió porque esta aplicación cuenta con una skill que se encuentra en las librerías de Amazon y según las pruebas realizadas este mostro ser la más compatible con el dispositivo Alexa.

Figura 5: Sinric Pro

Adaptado:

https://www.data.ai/es/apps/ios/app/sinric-pro/

Los sensores que se emplearon para el proyecto son 3 un sensor de temperatura un sensor de presencia y un sensor infrarrojo.

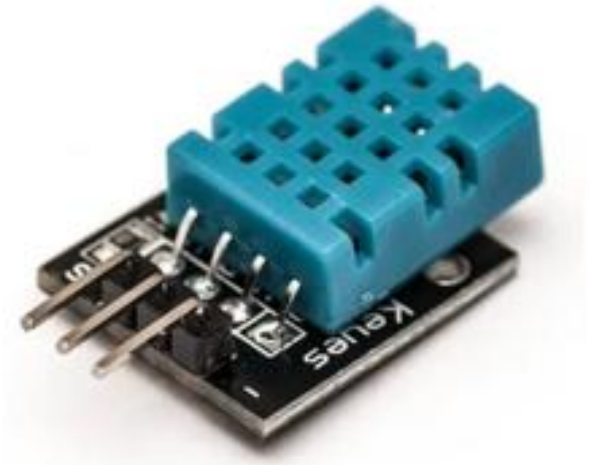
SENSOR DE TEMPERATURA Y HUMEDAD DHT11:

El sensor DHT11 es un sensor electrónico encargado de recolectar los valores de temperatura y humedad del ambiente a su vez posee un bajo costo y un sencillo funcionamiento, este a su vez integra un sensor de humedad capacitivo y a su vez también posee un termistor para sensar la temperatura y humedad del ambiente.

El sensor DHT11 posee las siguientes características:

- Un Voltaje de Operación: 3V - 5V DC
- Rango de medición de temperatura: 0 a 50 °C
- Precisión de medición de temperatura: ±2.0 °C
- Resolución Temperatura: 0.1°C
- Rango de medición de humedad: 20% a 90% RH.
- Precisión de medición de humedad: 5% RH.
- Resolución Humedad: 1% RH
- Tiempo de sensado: 1 seg.
- Interfaz digital: Single-bus (bidireccional)
- Modelo: DHT11
- Dimensiones: 16*12*5 mm
- Peso: 1 gr.
- Carcasa de plástico celeste

Figura 6: Sensor de temperatura y Humedad DTH11



ADAPTADO:

https://naylampmechatronics.com/sensores-temperatura-y-humedad/57-sensor-de-temperatura-y-humedad-relativa-dht11.html

SENSOR DE PRESENCIA HC-SR501:

Este módulo permite la detección de personas mediante sensores PIR (infrarrojo pasivo) muy utilizados en los sistemas de alarma de puertas automáticas.

El sensor HC-SR501 posee las siguientes características:

- Dimensiones aproximadas: 3.2cm x 2.4cm x 1.8 cm
- Circuitos de control incluidos en el módulo
- Tiempo de activación y sensibilidad ajustable
- Voltaje de operación: 4.5 – 20 V
- Corriente de operación: <80uA
- Salida: Pulso lógico 3.3V
- Tiempo de retardo: 5 – 200 Segundos (ajustable)
- Distancia de detección: 3 – 7 Metros (ajustable)

Figura 7: Sensor de presencia HC-SR501

ADAPTADO:

https://www.diarioelectronicohoy.com/blog/sensor-hc-sr501-con-arduino

SENSOR INFRARROJO PIC:

Se disponen lado a lado un LED infrarrojo y una fotorresistencia, el LED es el emisor y el fototransistor el receptor. Es un componente electrónico muy útil para detectar obstáculos y señales de otros dispositivos como mandos a distancia. tomar el control.

El sensor infrarrojo posee las siguientes características:

- Modelo: FC-51.
- Chip de funcionamiento: LM393.
- Voltaje de alimentación: 3.3V – 5V.
- Voltaje de salida digital: 5V.
- Dimensiones: 31 mm x 15 mm x 7 mm.
- Distancia de detección: 20 mm – 300 mm (ajustable)
- Angulo de detección: 35°
- Pines: VCC: Voltaje de alimentación

Figura 8: Sensor infrarrojo

ADAPTADO: https://uelectronics.com/producto/fc-51-sensor-de-obstaculos-reflectivo-infrarojo/-+

6. Marco metodológico.

METODOLOGIA

Se procedió con el desarrollo del banco de pruebas con el ESP-32 el cual se construyó en una caja metálica, el cual cuenta con su respectiva alimentación, y para las salidas se usan relés que irán conectados a leds pilotos que simulan, un ventilador, un motor para puerta de garaje, luces de ambiente para dormitorios, salas, oficina y comedor, también se emplean sensores como: un sensor de temperatura, que será controlado por la asistente Alexa para saber la sensación térmica del lugar, proximidad y además de infrarrojo que al detectar una variación de temperatura emite un pulso que enciende un led piloto, y el dispositivo Alexa también podrá dar la información de estos sensores con comandos de voz previamente programados.

Las prácticas que se pueden realizar son: programación y conexión de nuevos dispositivos, programación de ambientes interconectados mediante la aplicación de Alexa, la medición de temperatura de un cuarto mediante un sensor de temperatura y en caso de que la temperatura sea baja decirle al dispositivo Alexa que encienda la calefacción o si sube la temperatura que se encienda un ventilador y la detección de personas mediante los sensores de infrarrojo y proximidad los cuales con la aplicación pueden emitir diferentes sonidos y palabras previamente configuradas.

Figura 9: Elementos de protección

La primera parte del proyecto se centró en buscar los elementos de protección adecuados para proteger los elementos internos del banco de pruebas de modo que estos no resulten dañados en caso de una subida repentina del voltaje}

Figura 10: medición de la caja para la colocación de los elementos

Se realizo la medición de los elementos tanto de protección, así como los elementos de control, salidas y alimentación y se decide implementar el banco de pruebas en una caja metálica de 40x40x20, donde se procederá a colocar los elementos del tablero los cuales estarán empotrados en el plafón de la caja metálica.

Figura 11: Selector y alimentación.

Para la entrada de alimentación al banco de pruebas se empleó un conector polarizado el cual brinda la energía que el banco de pruebas utilizara y para controlar el paso de energía se empleó un selector el cual al encontrarse en posición de apagado impide el paso de la corriente y al activarse y permitir el paso de esta se enciende un led verde interno que indica que hay paso de energía.

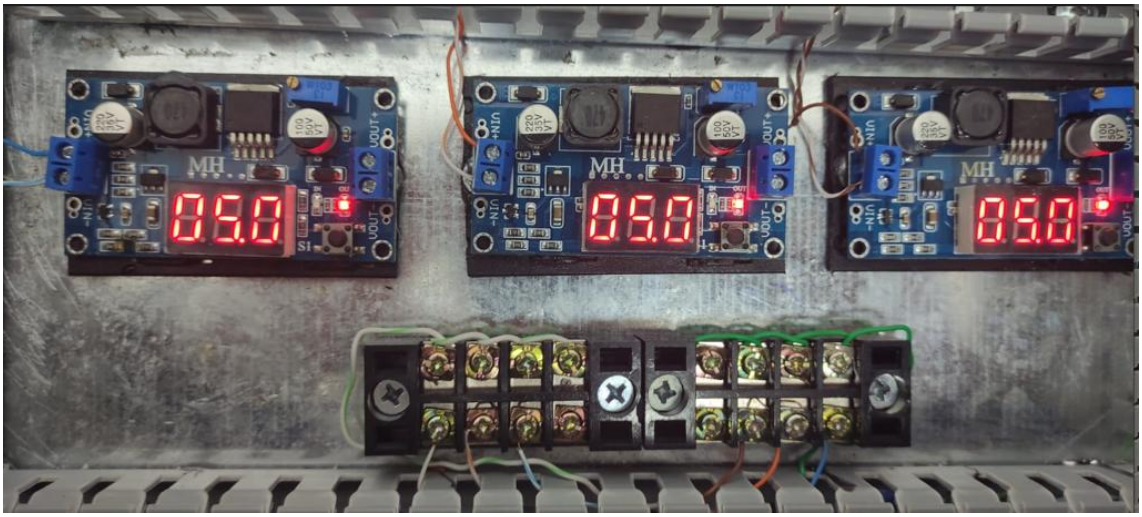
Figura 12: Colocación de los elementos

Se procedió a buscar la mejor posición de los elementos con la cual conseguimos aprovechar la mayor cantidad de espacio posible.

Figura 13: colocación y empotrado de los elementos

Se procedió a empotrar los elementos en el plafón siguiendo la secuencia de que los elementos de alimentación fueron colocados en la parte superior los elementos de control como fueron los relés que controlaran las salidas fueron colocados en la parte central así como el chip esp32.

Figura 14: Probando el funcionamiento de los elevadores de voltaje.



Se decidió por la instalación de elevadores de voltaje para alimentar los sensores debido a que en caso de que los estudiantes deseen conectar otros

tipos de sensores con un consumo más elevado de voltaje puedan ajustar estos mismos elevadores según sea su necesidad.

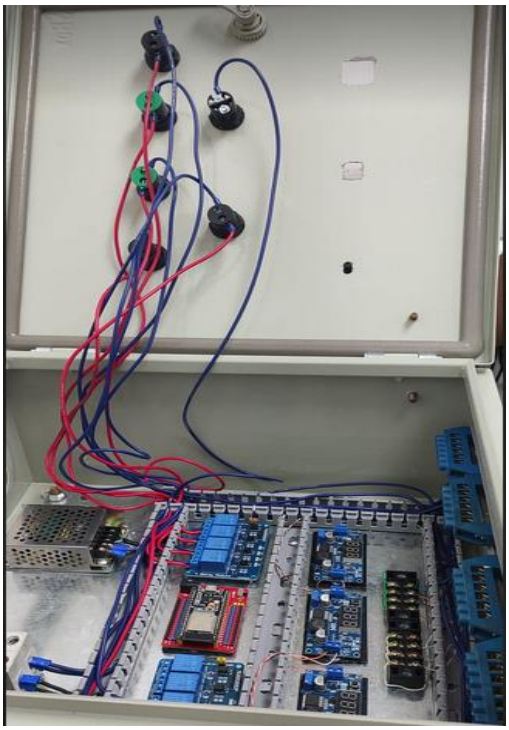
Figura 15: colocación de leds de salida

Se procedió a realizar los agujeros por donde entraran los leds y se los procedió a colocar cada led representa una salida ya se una habitación o un elemento tal como un ventilador o un aire acondicionado.

Figura 16: Vista frontal del tablero

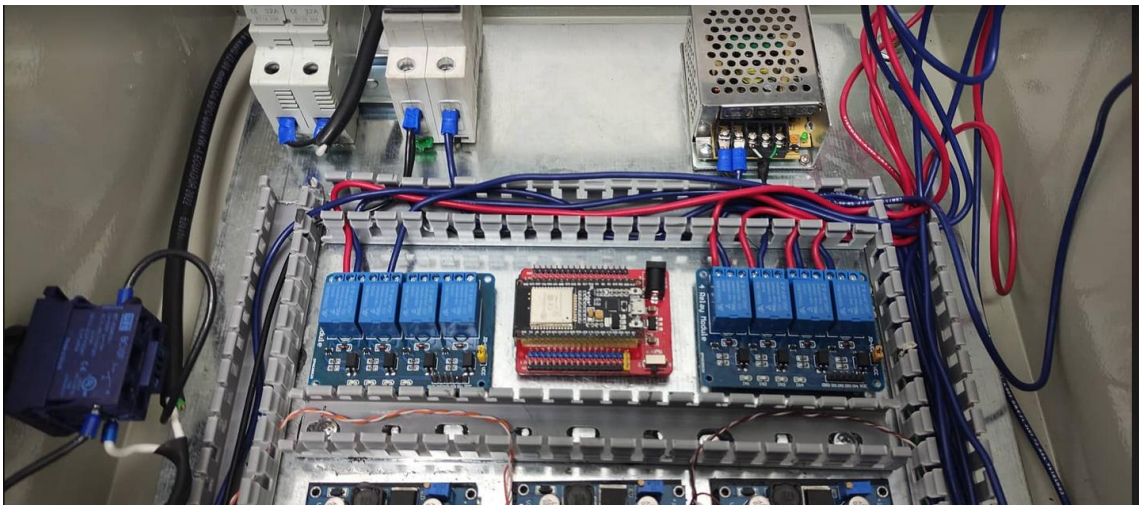
Se procedió a colocar canaletas ranuradas para el paso de los cables, así como la alimentación que se encargará de darle energía al esp32 y los sensores estará a cargo de una fuente conmutada de 5v, posteriormente se colocaron borneras a los cuales ira conectado los elementos de salida.

Figura 17: Cableado de los elementos



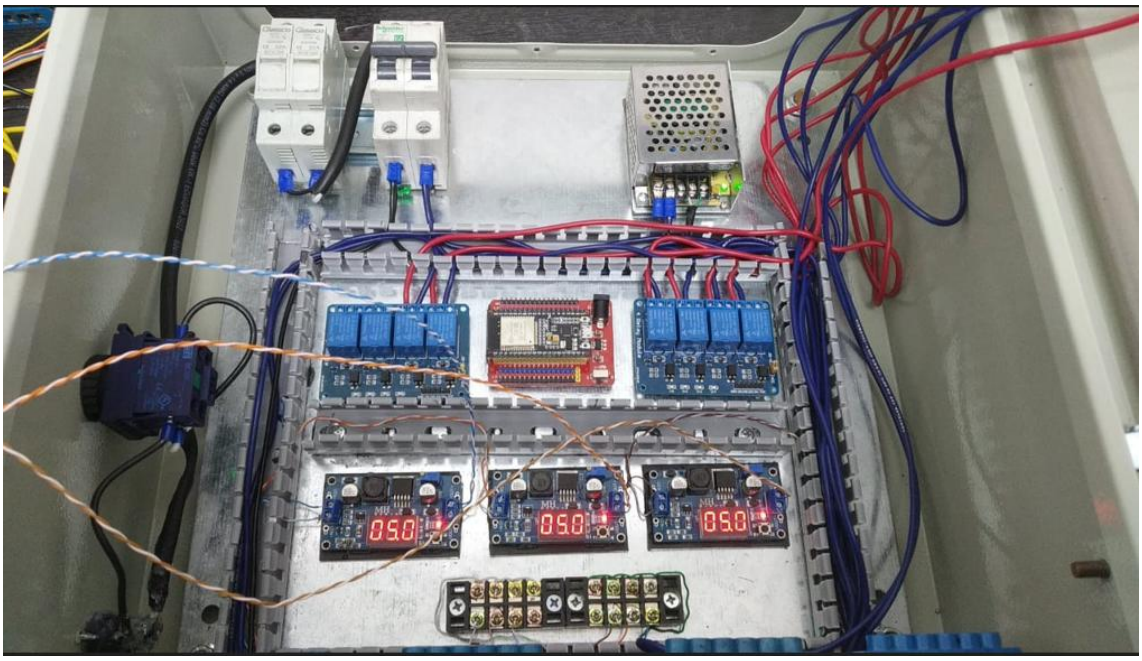
Se realizo el cableado de los elementos, así como su respectiva conexión y posterior prueba del funcionamiento con la programación y el Alexa.

Figura 18: Puesta del chip ESP 32



Se procedió a colocar el chip esp32 de modo que este quedara en el centro y facilite la conexión a los elementos circundante y no causara problemas por el largo de los cables.

Figura 19: Prueba de todos los elementos conectados



Una vez todos los elementos se colocaron en el tablero se procedió a realizar una prueba de los elementos con corriente, el funcionamiento fue satisfactorio

no se presentó ningún inconveniente a la hora de probarlo se realizaron las mediciones respectivas para verificar que la fuente recibía el voltaje correcto, así como comprobar que esta suministrara el voltaje adecuado.

7. Resultados

- Se logro establecer la conexión entre el ESP-32 y el dispositivo Alexa de una manera satisfactoria
- Se realizo la construcción del banco de pruebas y su correcto funcionamiento
- Se logro la conexión de sensores al dispositivo Alexa mediante la aplicación Sinric pro

8. Cronograma

Plan de trabajo de proyecto de tésis	abr-22				may-22				jun-22				jul-22				ago-22			
	Semanas																			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Realizar reuniones con el tutor para sugerencias e ideas del proyecto.																				
Investigar todo lo referente al proyecto para despejar dudas y hallar conclusiones.																				
Pruebas en protoboard del circuito de control para el sistema domótico.																				
Pruebas con la circuitería en gabinete.																				
Elaboración del documento para sustentación de la tésis.																				
Culminación del proyecto de tésis para sustentación.																				

9. Presupuesto

MATERIALES	Unidades	Precio Unitario	Envío	Precio Total
Módulos relay x8	2	11	5	27
Cable UTP categoría 6, 20 metros	1	15	5	20
ESP32+5dls de envío	3	12	5	41
Alexa Echo Dot Gen 3	1	40	5	45
Tarjeta perforada	3	8	5	29
Sensor de humedad para arduino	1	4	5	9
sensor de temperatura para arduino	1	12	5	17
Sensor de movimiento para arduino	1	3	5	8
Sensor de presión para arduino	1	10	5	15
Modem wifi	1	10	0	10
focos led a 110v	3	1.50	5	9,5
motor a 5vdc	2	5	5	15
Boton de encendido general	2	1	5	7
fuente 5vdc 1a	1	5	5	10
6 metros de cable flexible número 14	1	1	5	23
Gabinete Metálico Eléctrico 40x40x20	1	85	5	90
Electroválvula 5vdc	2	21	5	26
Bornera de 12 polos	1	1,5	0	1,5
Canaletas eléctricas	2	1	0	2
Breaker schneider 2p 10a	1	5	0	5
Portafusibles camsco 32a	1	10	0	10
Boquilla para foco	3	3	0	9
Mostrador de voltaje	3	7	0	21
Total:				450

10. Conclusiones

- Se logró enlazar la asistente virtual Alexa con el chip Esp32 mediante programación en el IDE de Arduino y librerías específicas del chip para poder controlar diferentes equipos como luces, alarmas, motores, resistencias térmicas, sensores con su debida programación y así implementar el banco de pruebas para uso de los estudiantes y cuerpo docente de la Universidad Politécnica Salesiana.
- Se logro desarrollar un banco

11. Recomendaciones

- Para un mejor enlace con diferentes tipos de sensores, se recomienda el uso de todas las opciones de la página de Sinric.
- En caso de que la programación no se cargue en el ESP-32 es necesario tener presionado el botón load de este error se suele presentar raramente.
- Tener cuidado al momento de establecer los nombres de la salida tal como se lo escribe en el código se lo debe decir al dispositivo Alexa caso contrario no lo reconoce.

12. Referencias Bibliográficas

Aguirre, R. E. (2018). DISEÑO DE UN SISTEMA DOMÓTICO DE BAJO COSTO EN UNA PLATAFORMA OPENSOURCE BAJO EL CONCEPT DEL IOT. Bogota.

Alexander, I. G. (2021). Análisis, diseño y desarrollo de un sistema inteligente y control de cultivos con IoT. Quito.

Amazon. (s.f.). https://developer.amazon.com/. Obtenido de https://developer.amazon.com/es-ES/alexa

Balibrea, R. H. (2012). Tecnología domótica para el control de una vivienda. Cartagena.

Carmenate, J. G. (s.f.). programarfacil.com. Obtenido de https://programarfacil.com/esp8266/esp32/

developer.mozilla.org. (22 de Enero de 2022). Obtenido de https://developer.mozilla.org/es/docs/Web/HTTP/Overview

Eduardo. (30 de Abril de 2019). Domotica en Casa. Obtenido de Domotica en Casa: https://domoticaencasa.es/esp32-primer-contacto/

Eldridge, S., Rose, K., & Chapin, L. (2015). La internet de las cosas - Una breve reseña. Internet Society.

Espressif Systems. (5 de septiembre de 2016). ESP32 SERIES. Obtenido de espressif.com: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Fonseca Yupa, J. D., & Soria Badillo, D. A. (2020). DISEÑO E IMPLEMENTACIÓN DE CONTROL DOMÓTICO CON APLICACIONE ADAPTATIVAS BASADAS EN IOT. Quito.

formacion profesional. (s.f.). Obtenido de formacionprofesional: https://www.formacionprofesional.info/los-modulos-esp32-y-lora-mas-vendidos/

Mendez de la Torre, R. E. (2022). DISEÑO E IMPLEMENTACION DE UN MODULO DE ENTRENAMIENTO UTILIZANDO EL PROCESADOR ESP32 PARA APLICACIONES DE DOMOTICA. Guayaquil.

PENTADOM. (2020). PENTADOM. Obtenido de https://pentadom.com/google-assistant-alexa-o-siri-que-asistente-de-voz-elegir-para-mi-vivienda-domotica/

13. Anexos

1) Código de leds pilotos

Código de Arduino con librerías de Esp32 para leds piloto:

```
//UNIVERSIDAD POLITECNICA SALESIANA
//Sede Guayaquil
//Tesis de Grado
//Previa a la obtención del título de Ingeniero Electrónico
//Carrera: Ingeniería en Electrónica y Automatización
//Autores: Hugo Xavier Alvarez Saltos
//          Alexis Paúl Loor Torres
//Tutor:    Ingeniero Luis Antonio Neira Clemente, MSC.

//Inicialización

#ifdef ARDUINO_ARCH_ESP32
#include <WiFi.h>
#else
#include <ESP32WiFi.h>
#endif

#include <Espalexa.h>

int Comedor = 15;
int Dormitorios = 0;
int Ventilador = 2;
int Salas = 17;
int Garaje = 4;
int Oficina = 16;
```

```
const char* ssid = "Claro_ALVAREZ0000856402";
const char* password = "7558478878496";

Espalexa alexita;

void FuncionDormitorios(uint8_t brightness);
void FuncionComedor(uint8_t brightness);
void FuncionVentilador(uint8_t brightness);
void FuncionSalas(uint8_t brightness);
void FuncionGaraje(uint8_t brightness);
void FuncionOficina(uint8_t brightness);


void setup() {
  Serial.begin(115200);
  pinMode(Comedor, OUTPUT);
  pinMode(Dormitorios, OUTPUT);
  pinMode(Ventilador, OUTPUT);
  pinMode(Salas, OUTPUT);
  pinMode(Garaje, OUTPUT);
  pinMode(Oficina, OUTPUT);
  ConectarWifi();
  alexita.addDevice("Dormitorios", FuncionDormitorios);
  alexita.addDevice("Comedor", FuncionComedor);
  alexita.addDevice("Ventilador", FuncionVentilador);
  alexita.addDevice("Salas", FuncionSalas);
  alexita.addDevice("Garaje", FuncionGaraje);
  alexita.addDevice("Oficina", FuncionOficina);
  alexita.begin();
```

```
}

void loop() {
  ConectarWifi();
  alexita.loop();
  delay(1);
}

void ConectarWifi() {
  if (WiFi.status() != WL_CONNECTED) {
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    Serial.println("");
    Serial.println("Connecting to WiFi");
    while (WiFi.status() != WL_CONNECTED) {
      digitalWrite(Comedor, 1);
      delay(500);
      digitalWrite(Dormitorios, 1);
      delay(500);
      digitalWrite(Ventilador, 1);
      delay(500);
      digitalWrite(Salas, 1);
      delay(500);
      digitalWrite(Garaje, 1);
      delay(500);
      digitalWrite(Oficina, 1);
      delay(500);
      Serial.print(".");
    }
    Serial.print("Connected to ");
```

```
    Serial.println(ssid);
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());
  }
}

void FuncionDormitorios(uint8_t brightness) {
  Serial.print("Funcion Dormitorios - ");

  if (brightness) {
    digitalWrite(Dormitorios, 1);
    Serial.println(" Encender ");
  }
  else {
    digitalWrite(Dormitorios, 0);
    Serial.println(" Apagar ");
  }
}

void FuncionComedor(uint8_t brightness) {
  Serial.print("Funcion Comedor -");

  if (brightness) {
    Serial.println(" Encender ");
    digitalWrite(Comedor, 1);
  }
  else {
    Serial.println(" Apagar ");
    digitalWrite(Comedor, 0);
  }
```

```
}
void FuncionVentilador(uint8_t brightness) {
  Serial.print("Funcion Ventilador -");

  if (brightness) {
    Serial.println(" Encender ");
    digitalWrite(Ventilador, 1);
  }
  else {
    Serial.println(" Apagar ");
    digitalWrite(Ventilador, 0);
  }

}
void FuncionSalas(uint8_t brightness) {
  Serial.print("Funcion Salas -");

  if (brightness) {
    Serial.println(" Encender ");
    digitalWrite(Salas, 1);
  }
  else {
    Serial.println(" Apagar ");
    digitalWrite(Salas, 0);
  }

}
void FuncionOficina(uint8_t brightness) {
  Serial.print("Funcion Oficina -");
```

```
  if (brightness) {
    Serial.println(" Encender ");
    digitalWrite(Oficina, 1);
  }
  else {
    Serial.println(" Apagar ");
    digitalWrite(Oficina, 0);
  }

}

void FuncionGaraje(uint8_t brightness) {
  Serial.print("Funcion Garaje -");

  if (brightness) {
    Serial.println(" Encender ");
    digitalWrite(Garaje, 1);
  }
  else {
    Serial.println(" Apagar ");
    digitalWrite(Garaje, 0);
      }

}
```

2) Sensor Infrarrojo

```
//UNIVERSIDAD POLITECNICA SALESIANA

//Sede Guayaquil

//Tésis de Grado

//Previa a la obtención del titulo de Ingeniero Electrónico

//Carrera: Ingeniería en Electrónica y Automatización

//Autores: Hugo Xavier Alvarez Saltos

//        Alexis Paúl Loor Torres

//Tutor:   Ingeniero Luis Antonio Neira Clemente, MSc.

//Código de sensor infrarrojo

//Inicialización


#ifdef ENABLE_DEBUG

 #define DEBUG_ESP_PORT Serial

 #define NODEBUG_WEBSOCKETS

 #define NDEBUG

#endif


#include <Arduino.h>

#ifdef ESP8266

 #include <ESP8266WiFi.h>

#endif

#ifdef ESP32

 #include <WiFi.h>

#endif


#include "SinricPro.h"
```

```
#include "SinricProMotionsensor.h"


#define WIFI_SSID         "POCOX3"

#define WIFI_PASS         "Darkdwarf"

#define APP_KEY           "8530721f-142b-4bb3-8b5c-ff5c5e63b0b3"

#define APP_SECRET        "ca9d341b-89d5-47d5-8909-e4020ee874d1-62076131-458e-4bff-87ce-
3531d357989f"

#define MOTIONSENSOR_ID   "62e0a67c0bb1005386659b31"


#define BAUD_RATE         115200              // Change baudrate to your need


#define MOTIONSENSOR_PIN  36


bool myPowerState = true;

bool lastMotionState = false;

unsigned long lastChange = 0;


void handleMotionsensor() {

  if (!myPowerState) return;


  unsigned long actualMillis = millis();

  if (actualMillis - lastChange < 250) return;


  bool actualMotionState = digitalRead(MOTIONSENSOR_PIN);


  if (actualMotionState != lastMotionState) {
```

```
    Serial.printf("Motion %s\r\n", actualMotionState?"detected":"not detected");
    lastMotionState = actualMotionState;
    lastChange = actualMillis;
    SinricProMotionsensor &myMotionsensor = SinricPro[MOTIONSENSOR_ID];
    myMotionsensor.sendMotionEvent(actualMotionState);
  }
}
bool onPowerState(const String &deviceId, bool &state) {
  Serial.printf("Device %s turned %s (via SinricPro) \r\n", deviceId.c_str(), state?"on":"off");
  myPowerState = state;
  return true;
}
void setupWiFi() {
  Serial.printf("\r\n[Wifi]: Connecting");
  WiFi.begin(WIFI_SSID, WIFI_PASS);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.printf(".");
    delay(250);
  }
  IPAddress localIP = WiFi.localIP();
  Serial.printf("connected!\r\n[WiFi]: IP-Address is %d.%d.%d.%d\r\n", localIP[0], localIP[1],
localIP[2], localIP[3]);
}
void setupSinricPro() {
  // add device to SinricPro
```

```
  SinricProMotionsensor& myMotionsensor = SinricPro[MOTIONSENSOR_ID];



  myMotionsensor.onPowerState(onPowerState);


  SinricPro.onConnected([](){ Serial.printf("Connected to SinricPro\r\n"); });

  SinricPro.onDisconnected([](){ Serial.printf("Disconnected from SinricPro\r\n"); });

  SinricPro.begin(APP_KEY, APP_SECRET);

}


void setup() {

  Serial.begin(BAUD_RATE); Serial.printf("\r\n\r\n");


  pinMode(MOTIONSENSOR_PIN, INPUT);


  setupWiFi();

  setupSinricPro();

}

void loop() {

  handleMotionsensor();

  SinricPro.handle();

}
```

3) Sensor de movimiento

```
//UNIVERSIDAD POLITECNICA SALESIANA

//Sede Guayaquil

//Tesis de Grado

//Previa a la obtención del titulo de Ingeniero Electrónico

//Carrera: Ingeniería en Electrónica y Automatización

//Autores: Hugo Xavier Alvarez Saltos

//        Alexis Paúl Loor Torres

//Tutor:   Ingeniero Luis Antonio Neira Clemente, MSc.

//Código de sensor de movimiento

//Inicialización


#ifdef ENABLE_DEBUG

 #define DEBUG_ESP_PORT Serial

 #define NODEBUG_WEBSOCKETS

 #define NDEBUG

#endif


#include <Arduino.h>

#ifdef ESP8266

 #include <ESP8266WiFi.h>

#endif

#ifdef ESP32

 #include <WiFi.h>

#endif


#include "SinricPro.h"
```

```
#include "SinricProMotionsensor.h"


#define WIFI_SSID         "POCOX3"

#define WIFI_PASS         "Darkdwarf"

#define APP_KEY           "8530721f-142b-4bb3-8b5c-ff5c5e63b0b3"

#define APP_SECRET        "ca9d341b-89d5-47d5-8909-e4020ee874d1-62076131-458e-4bff-87ce-3531d357989f"

#define MOTIONSENSOR_ID   "62e0a67c0bb1005386659b31"


#define BAUD_RATE         115200


#define MOTIONSENSOR_PIN  36



bool myPowerState = true;

bool lastMotionState = false;

unsigned long lastChange = 0;


void handleMotionsensor() {

  if (!myPowerState) return;


  unsigned long actualMillis = millis();

  if (actualMillis - lastChange < 250) return;


  bool actualMotionState = digitalRead(MOTIONSENSOR_PIN);
```

```
  if (actualMotionState != lastMotionState) {
    Serial.printf("Motion %s\r\n", actualMotionState?"detected":"not detected");
    lastMotionState = actualMotionState;
    lastChange = actualMillis;
    SinricProMotionsensor &myMotionsensor = SinricPro[MOTIONSENSOR_ID];
    myMotionsensor.sendMotionEvent(actualMotionState);
  }
}
bool onPowerState(const String &deviceId, bool &state) {
  Serial.printf("Device %s turned %s (via SinricPro) \r\n", deviceId.c_str(), state?"on":"off");
  myPowerState = state;
  return true; // request handled properly
}
void setupWiFi() {
  Serial.printf("\r\n[Wifi]: Connecting");
  WiFi.begin(WIFI_SSID, WIFI_PASS);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.printf(".");
    delay(250);
  }
  IPAddress localIP = WiFi.localIP();
  Serial.printf("connected!\r\n[WiFi]: IP-Address is %d.%d.%d.%d\r\n", localIP[0], localIP[1],
localIP[2], localIP[3]);
}
```

```
void setupSinricPro() {
  SinricProMotionsensor& myMotionsensor = SinricPro[MOTIONSENSOR_ID];

  myMotionsensor.onPowerState(onPowerState);

  SinricPro.onConnected([](){ Serial.printf("Connected to SinricPro\r\n"); });
  SinricPro.onDisconnected([](){ Serial.printf("Disconnected from SinricPro\r\n"); });
  SinricPro.begin(APP_KEY, APP_SECRET);
}

void setup() {
  Serial.begin(BAUD_RATE); Serial.printf("\r\n\r\n");

  pinMode(MOTIONSENSOR_PIN, INPUT);

  setupWiFi();
  setupSinricPro();
}
void loop() {
  handleMotionsensor();
  SinricPro.handle();
}
```

4) Sensor de temperatura

```
//UNIVERSIDAD POLITECNICA SALESIANA
//Sede Guayaquil
//Tesis de Grado
//Previa a la obtención del título de Ingeniero Electrónico
//Carrera: Ingeniería en Electrónica y Automatización
//Autores: Hugo Xavier Alvarez Saltos
//        Alexis Paúl Loor Torres
//Tutor:   Ingeniero Luis Antonio Neira Clemente, MSc.
//Código de sensor de temperatura
//Inicialización

#ifdef ENABLE_DEBUG
 #define DEBUG_ESP_PORT Serial
 #define NODEBUG_WEBSOCKETS
 #define NDEBUG
#endif

#include <Arduino.h>
#ifdef ESP8266
 #include <ESP8266WiFi.h>
#endif
#ifdef ESP32
 #include <WiFi.h>
#endif

#include "SinricPro.h"
```

```
#include "SinricProTemperaturesensor.h"

#include "DHT.h" // https://github.com/markruys/arduino-DHT


#define WIFI_SSID         "POCOX3"

#define WIFI_PASS         "Darkdwarf"

#define APP_KEY           "8530721f-142b-4bb3-8b5c-ff5c5e63b0b3"

#define APP_SECRET        "ca9d341b-89d5-47d5-8909-e4020ee874d1-62076131-458e-4bff-87ce-3531d357989f"

#define TEMP_SENSOR_ID    "62e0a6560bb1005386659b08"

#define BAUD_RATE         115200

#define EVENT_WAIT_TIME   60000


#define DHT_PIN           34


DHT dht;


bool deviceIsOn;

float temperature;

float humidity;

float lastTemperature;

float lastHumidity;

unsigned long lastEvent = (-EVENT_WAIT_TIME);


bool onPowerState(const String &deviceId, bool &state) {

  Serial.printf("Temperaturesensor turned %s (via SinricPro) \r\n", state?"on":"off");

  deviceIsOn = state; // turn on / off temperature sensor
```

```
  return true;
}
void handleTemperaturesensor() {
  if (deviceIsOn == false) return;

  unsigned long actualMillis = millis();
  if (actualMillis - lastEvent < EVENT_WAIT_TIME) return;

  temperature = dht.getTemperature();
//  temperature = dht.getTemperature() * 1.8f + 32;
  humidity = dht.getHumidity();

  if (isnan(temperature) || isnan(humidity)) {
    Serial.printf("DHT reading failed!\r\n");
    return;
  }

  if (temperature == lastTemperature || humidity == lastHumidity) return;

  SinricProTemperaturesensor &mySensor = SinricPro[TEMP_SENSOR_ID];
  bool success = mySensor.sendTemperatureEvent(temperature, humidity);
  if (success) {  Serial.printf("Temperature: %2.1f Celsius\tHumidity: %2.1f%%\r\n", temperatura,
humidiad);
  } else {
    Serial.printf("Something went wrong...could not send Event to server!\r\n");
  }
```

```
  lastTemperature = temperature;

  lastHumidity = humidity;

  lastEvent = actualMillis;

}



// setup function for WiFi connection

void setupWiFi() {

  Serial.printf("\r\n[Wifi]: Connecting");

  WiFi.begin(WIFI_SSID, WIFI_PASS);


  while (WiFi.status() != WL_CONNECTED) {

    Serial.printf(".");

    delay(250);

  }

  IPAddress localIP = WiFi.localIP();

  Serial.printf("connected!\r\n[WiFi]: IP-Address is %d.%d.%d.%d\r\n", localIP[0], localIP[1],
localIP[2], localIP[3]);

}

void setupSinricPro() {

  SinricProTemperaturesensor &mySensor = SinricPro[TEMP_SENSOR_ID];

  mySensor.onPowerState(onPowerState);

  SinricPro.onConnected([](){ Serial.printf("Connected to SinricPro\r\n"); });

  SinricPro.onDisconnected([](){ Serial.printf("Disconnected from SinricPro\r\n"); });
```

```
  SinricPro.begin(APP_KEY, APP_SECRET);
}

void setup() {
  Serial.begin(BAUD_RATE); Serial.printf("\r\n\r\n");
  dht.setup(DHT_PIN);

  setupWiFi();
  setupSinricPro();
}

void loop() {
  SinricPro.handle();
  handleTemperaturesensor();
}
```