

Control de un grupo de dispositivos eléctricos con integración a asistentes inteligentes usando un esp32

Control of a group of electrical devices with integration to intelligent assistants using an esp32

Alejandro Manzo Juárez, Igor Guryev, Adán Flores Balderas ¹

¹Departamento de Estudios Multidisciplinarios, División de Ingenierías del Campus Irapuato-Salamanca, Universidad de Guanajuato
a.manzojuarez@ugto.mx¹, guryev@ugto.mx , adanfb@ugto.mx

Resumen

En el proyecto se realiza el desarrollo un dispositivo electrónico para servir de interfaz entre los elementos no compatibles con los asistentes inteligentes y los asistentes inteligentes, así como el desarrollo de la interfaz de control del dispositivo con base en los sensores capacitivos graduales integrados al microcontrolador ESP32. El sistema de control a través del servidor WEB y de los asistentes inteligentes, permitiendo un aumento en la seguridad de la casa en ausencia de los habitantes.

Palabras clave: Asistentes inteligentes, Internet de las cosas(IOT), Servidor Web, ESP32, Sensores Táctiles, Electrónica de Potencia.

Introducción

El uso de asistentes inteligentes ha transformado nuestra vida diaria, facilitando el control remoto de dispositivos electrónicos y mejorando significativamente la calidad de vida de las personas con movilidad reducida (Aguilar, 2020). Sin embargo, existen casos en los que estas tecnologías aún no han alcanzado a todos los usuarios, como aquellos con dispositivos no compatibles o quienes se preocupan por el aumento de la contaminación electrónica. Además, el control de la intensidad lumínica mediante asistentes inteligentes permite optimizar la iluminación de una habitación, contribuyendo al cuidado de la visión del usuario (Ir, 2024).

Aunque el avance tecnológico ha integrado numerosos dispositivos inteligentes, muchas personas todavía no pueden acceder a estos beneficios. Esto se debe a factores como los altos costos, la funcionalidad limitada o el hecho de que aún poseen dispositivos en buen estado que no desean reemplazar (Gutierrez Sanin, 2019). Nuestro proyecto busca ofrecer una solución que permita integrar el control mediante asistentes inteligentes a dispositivos que no cuentan con esta característica, enfocándonos inicialmente en las luminarias.

Actualmente, existen en el mercado dispositivos que permiten el control de luminarias individuales. Sin embargo, en la mayoría de los casos, el sistema de control y la luminaria están integrados en un solo aparato. Esto significa que cuando la luminaria falla, el controlador también se desecha, aumentando así la contaminación electrónica (Aguilera, La basura electronica y la contaminacion ambiental, 2010). Nuestro dispositivo propone una solución a este problema, separando el control de la luminaria. Esto permite que, al final de la vida útil de la luminaria, el controlador siga funcionando, brindando además la posibilidad de controlar luminarias no compatibles con asistentes inteligentes de manera sencilla.

Las personas con movilidad reducida, como ancianos, enfermos o discapacitados, enfrentan dificultades para desplazarse, lo que convierte acciones simples como encender o apagar una luz en una tarea incómoda o incluso riesgosa. Con nuestro dispositivo, tal como lo intenta pineda en (Pineda Fajardo, 2021) estas personas podrán mejorar su estilo de vida solo que, mediante el uso de un asistente inteligente, como Google Asistente o Alexa, desde su celular para controlar la intensidad o apagar la luz sin necesidad de estar presentes en la habitación, aprovechando la tecnología IoT.

Además, la iluminación inadecuada durante nuestras actividades diarias puede causar un sobreesfuerzo visual, ya sea por exceso o falta de luz, lo que puede dañar la visión a largo plazo. Al proporcionar un control eficiente de la iluminación, nuestro dispositivo permitirá a los usuarios ajustar la luz según sus necesidades, reduciendo el daño ocular por fatiga visual.

En resumen, nuestro dispositivo busca reducir la contaminación electrónica, mejorar la calidad de vida de las personas con movilidad limitada, extender el uso de luminarias sin control por asistentes inteligentes y fomentar una iluminación saludable para preservar la visión. A diferencia de las opciones comerciales existentes, que generalmente ofrecen control individual de luminarias y requieren acceso a internet y una membresía, nuestro dispositivo permitirá el control de hasta tres luminarias reemplazables, sin la necesidad de una suscripción y con una interfaz táctil para su uso en caso de fallas de conexión a internet.

Metodología Propuesta

En la Ilustración 1. Se aprecia un diagrama de flujo de la interacción del usuario con el sistema. El usuario puede interactuar directamente con el dispositivo o usando un asistente inteligente para controlar la intensidad lumínica de las cuatro luminarias conectadas a la fase de potencia, que controla el dispositivo.

Cargas Electricas

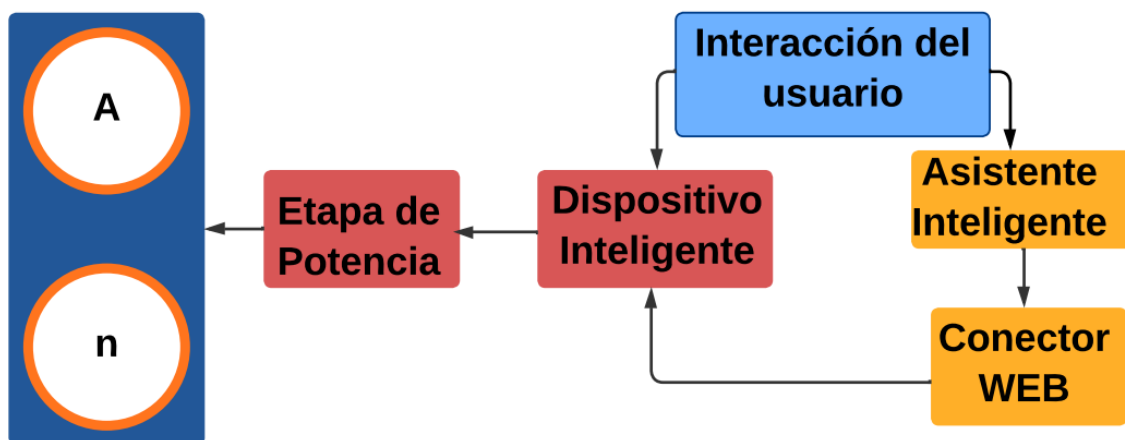


Ilustración 1.- Diagrama de interacción del usuario con el sistema propuesto.
Fuente: Elaboración propia.

Configuración del Hardware

El ESP32 se conecta a dispositivos de electrónica de potencia (triacs, opto acopladores) que permiten controlar la corriente alterna de manera segura y eficiente.

El microcontrolador recibe señales de entrada (lecturas de sensores táctiles o comandos de voz a través de asistentes inteligentes) para ajustar los niveles de potencia o activación de dispositivos a su vez debe de recibir constantemente la señal de cruce por cero debido a que es en base a la misma que se configura el retardo correspondiente.

Desarrollo del Código en C++

El código está escrito en C++ para aprovechar la eficiencia y el control detallado sobre el hardware que este lenguaje proporciona. La elección de C++ permite manejar tanto las operaciones de bajo nivel, como el control directo de pines GPIO, así como la gestión de procesos más complejos.

Configuración de Pines: La primera parte del código se centra en definir y configurar los pines del ESP32. Esto incluye especificar los pines que se utilizarán como entradas para los sensores de cruce por cero, así como de los sensores táctiles y los que actuarán como salidas para controlar los triacs.



Interrupciones y Sincronización: Dado que el control de dispositivos de potencia implica trabajar con señales de corriente alterna, es crucial utilizar interrupciones para detectar el cruce por cero de la señal de corriente. Esto asegura que los triacs se activen en el momento adecuado para regular la potencia suministrada.

Ajuste del Ángulo de Disparo: Después de detectar el cruce por cero, el código en C++ genera un retardo preciso antes de enviar la señal de activación al triac. Este retardo corresponde al ángulo de disparo. Por ejemplo:

Un ángulo de disparo pequeño (cerca del cruce por cero) significa que el triac se activa casi inmediatamente, permitiendo que pase la mayor parte del ciclo de la señal AC y, por lo tanto, entregando máxima potencia a la luminaria (Ornelas, 2019).

Un ángulo de disparo más grande (más cerca del final del ciclo) significa que el triac se activa tarde, permitiendo que pase solo una pequeña parte de la señal AC y, por lo tanto, entregando menos potencia (Ornelas, 2019).

Control de Sensores: Se programan rutinas para leer señales de entrada desde sensores. Esto incluye la recepción de comandos para ajustar automáticamente la potencia según las necesidades del usuario.

Gestión de Conectividad y Comunicación

El ESP32 tiene capacidades de Wifi y Bluetooth, lo que permite que el código también maneje la comunicación con otros dispositivos o servicios en la nube (ESP32, 2024). Por ejemplo, se puede recibir comandos de voz desde un asistente inteligente para ajustar la intensidad de la luz controlada por el sistema de potencia.

El código se asegura de mantener la conectividad estable, y en caso de que se pierda la conexión, se han implementado mecanismos de control manual para que el sistema siga siendo funcional.

Desarrollo del servidor:

Node-RED es una herramienta basada en Node.js la cual facilita la creación de aplicaciones de Internet de las Cosas (IoT) mediante un entorno visual de programación ("Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot", 2018). Para un proyecto de electrónica de potencia que utiliza un ESP32, el desarrollo de un servidor en Node-RED permite gestionar la comunicación, monitoreo y control del sistema de manera eficiente, actúa como punto central de comunicación entre el ESP32 los servicios de google. Recibe comandos de control, procesa datos y distribuye instrucciones de manera adecuada.

Componentes Clave

Nodos MQTT para Comunicación:

El protocolo MQTT es ligero y eficiente, ideal para comunicaciones IoT. Se configuran nodos MQTT en Node para establecer la conexión con el ESP32. Estos nodos permiten recibir datos del microcontrolador y enviarle instrucciones de control.

Por ejemplo, si el ESP32 debe ajustar el ángulo de disparo de un triac para regular la potencia de un dispositivo, el servidor Node puede enviar un comando específico usando MQTT.

Nodos de Funciones para Procesar Datos:

Los nodos de funciones permiten escribir lógica personalizada en JavaScript. Aquí se pueden procesar los datos recibidos del ESP32 para tomar decisiones. Por ejemplo, se puede calcular el ángulo de disparo adecuado en función de la entrada del usuario o sensores externos y luego enviar esta información al ESP32.

Integración con Asistentes Inteligentes:

A través de módulos adicionales, el servidor Node puede conectarse con servicios como Google Asistente o Amazon Alexa, permitiendo controlar el sistema de potencia mediante comandos de voz. Esto se logra enviando comandos a Node que, a su vez, se comunican con el ESP32 para ejecutar la acción deseada (por ejemplo, "Ajustar la luz al 50%").



Flujo de Trabajo en Node

Recepción de Comandos: Los nodos de entrada (por ejemplo, MQTT, HTTP o WebSocket) reciben comandos del usuario, ya sea desde una aplicación web, un asistente inteligente o directamente desde el ESP32.

Procesamiento de Datos: Los datos se procesan a través de nodos de funciones y se toman decisiones en función de la lógica establecida. Si se recibe un comando para ajustar la potencia, el servidor calculará el ángulo de disparo necesario y preparará un mensaje para enviarlo al ESP32.

Envío de Instrucciones al ESP32: Utilizando nodos de salida MQTT, el servidor envía instrucciones claras al ESP32, permitiendo que el microcontrolador ajuste la potencia de los dispositivos conectados.

Monitoreo y Feedback: El ESP32 puede enviar datos de vuelta al servidor.

Conectividad

Gestión de Conexiones: El servidor Node se configura para manejar múltiples conexiones simultáneamente, asegurando que el sistema responda rápidamente a las solicitudes de los usuarios.

Manejo de Errores: Se implementan mecanismos para manejar errores de comunicación y reconectar automáticamente si se pierde la conexión con el ESP32, asegurando la estabilidad del sistema.

Desarrollo del dispositivo:

La conexión de la fase de control:

Debido a que esta parte se compone explícitamente de sensores Táctil se buscan conectores de 3 terminales y conectores de 4 terminales esto debido a que son a los cuales las terminales de los paneles táctiles se acoplaran siendo los tres primeros para la selección de las luminarias a los pines T0, T3 y T8 para la selección de la luminaria 1, 2 y 3 respectivamente. Mientras que los otros cinco se conectarán a los pines T4, T5, T6, T7, T9 que corresponde a los niveles de 0% 25% 50% 75% y 100% respectivamente de la intensidad deseada.

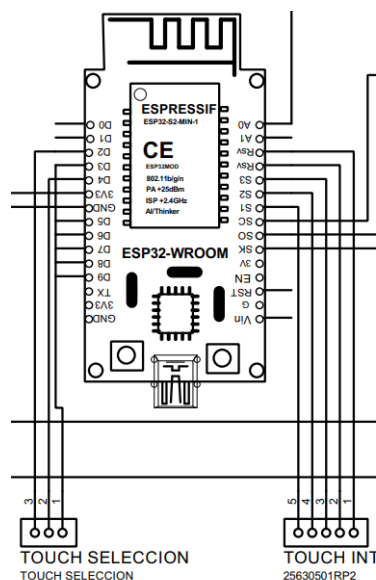


Ilustración 2 esquemático de conexión del panel táctil
Fuente: Elaboración propia.

Fase de detección de cruce por cero

Pasamos a la fase de detección de cruce por cero la cual consta de dos resistencias de potencia, un módulo de detección de cruce por cero H11AA1 y una resistencia de 0.5 watts.

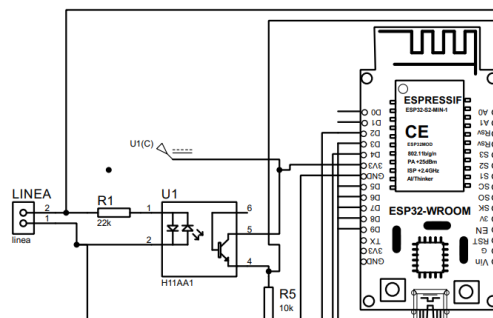


Ilustración 3 esquemático de conexión de control de cruce por cero
Fuente: Elaboración propia.

Para la conexión se utiliza un conector de dos terminales debido a que de esta sección se energizará el prototipo, una terminal se conecta a la resistencia y la segunda a la segunda resistencia, las otras partes se conectan a la terminal 1 y 2 respectivamente, a la terminal 5 se conecta a la terminal de 3.3v del módulo, la terminal 4 se conecta en serie con la resistencia de 1 k ohm la cual a su vez se conecta a la terminal de tierra del módulo mientras que se conecta la terminal 4 del h11a1 a la terminal del pin 21 del esp32.

Fase de Potencia

En la conexión de potencia se ocuparán 3 resistencias, tres opto acopladores, 3 triacs y un conector de 5 terminales

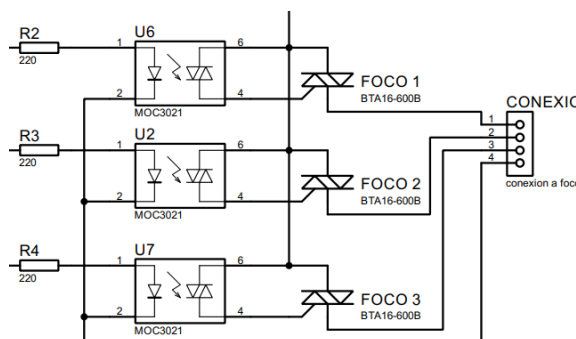


Ilustración 4 esquemático conexión de fase de potencia
Fuente: Elaboración propia.

Las tres resistencias se conectarán primero a los pines 19, 25 y 26 del módulo respectivamente, a su vez cada resistencia se conecta a la terminal 1 del opto acoplador, la terminal 2 de cada uno de los opto acopladores se conectan a la terminal de tierra de módulo esp32, la terminal 6 del opto acoplador se conecta a la terminal común del conector a fase de entrada y a su vez con la terminal de entrada del triac, el terminal 4 del opto acoplador se conecta a la terminal de control del triac; para el triac la última terminal se conecta al conector de 4 terminales el ultimo conectores conecta al neutro de la fase.

los tres primeros terminales del conector corresponden a las luminarias mientras que el cuarto es correspondiente al neutro común de las luminarias.

Resultados

La ilustración 5 muestra el retardo generado de 7500 us mostrado en la señal azul en comparativa con la señal detectada en amarillo como la señal de cruce por cero, a su vez en la ilustración 6 se muestra la luminaria a su correspondiente porcentaje de 0% o cual se comprueba con la ilustración 7 donde el luxómetro solo está mostrando la iluminación del ambiente.

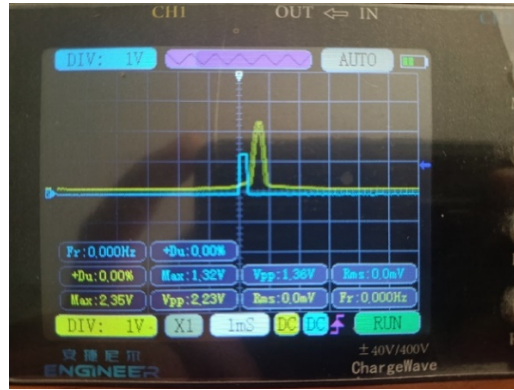


Ilustración 5 RETARDO de 7500 us mostrado en el osciloscopio en comparativa con la detección de cruce por cero
Fuente: Elaboración propia.

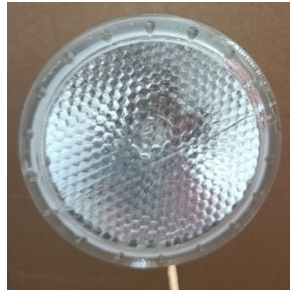


Ilustración 6 intensidad de la luminaria al 0 %
Fuente: Elaboración propia.



Ilustración 7 intensidad luminica en lux detectada a un 0 %
Fuente: Elaboración propia.

La ilustración 8 muestra el retardo generado de 6000 us mostrado en la señal azul en comparativa con la señal detectada en amarillo como la señal de cruce por cero, a su vez en la ilustración 9 se muestra la luminaria a su correspondiente porcentaje de 25% o cual se comprueba con la ilustración 10 donde el luxómetro muestra una iluminación de 93 lux.



Ilustración 8 RETARDO DE 6000 US MOSTRADO EN EL OSCILOSCOPIO EN COMPARATIVA CON LA DETECCIN DE CRUCE POR CERO.
Fuente: Elaboración propia.

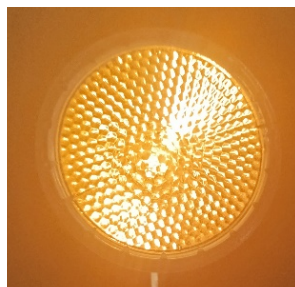


Ilustración 9 INTENSIDAD DE LA LUMINARIA AL 25 %
Fuente: Elaboración propia.



Ilustración 10 INTENSIDAD LUMINICA EN LUX DETECTADA A UN 25%
Fuente: Elaboración propia.

La ilustración 11 muestra el retardo generado de 3000 us mostrado en la señal azul en comparativa con la señal detectada en amarillo como la señal de cruce por cero, a su vez en la ilustración 12 se muestra la luminaria a su correspondiente porcentaje de 50% o cual se comprueba con la ilustración 13 donde el luxómetro muestra una iluminación de 1093 lux.



Ilustración 11 RETARDO DE 3000 US MOSTRADO EN EL OSCILOSCOPIO EN COMPARATIVA CON LA DETECCIN DE CRUCE POR CERO
Fuente: Elaboración propia.



Ilustración 12 INTENSIDAD DE LA LUMINARIA AL 50 %
Fuente: Elaboración propia.



Ilustración 13 INTENSIDAD LUMINICA EN LUX DETECTADA A UN 50%
Fuente: Elaboración propia.

La ilustración 14 muestra el retardo generado de 1500 us mostrado en la señal azul en comparativa con la señal detectada en amarillo como la señal de cruce por cero, a su vez en la ilustración 15 se muestra la luminaria a su correspondiente porcentaje de 100% o cual se comprueba con la ilustración 16 donde el luxómetro muestra una iluminación de 5290 lux.



Ilustración 14 RETARDO DE 1500 US MOSTRADO EN EL OSCILOSCOPIO EN COMPARATIVA CON LA DETECCIN DE CRUCE POR CERO
Fuente: Elaboración propia.



Ilustración 15 INTENSIDAD DE LA LUMINARIA AL 100 %
Fuente: Elaboración propia.



Ilustración 16 INTENSIDAD LUMINICA EN LUX DETECTADA A UN 100%
Fuente: Elaboración propia.

La relación entre el retardo y la iluminación generada por la luminaria queda mostrada en la ilustración 17.

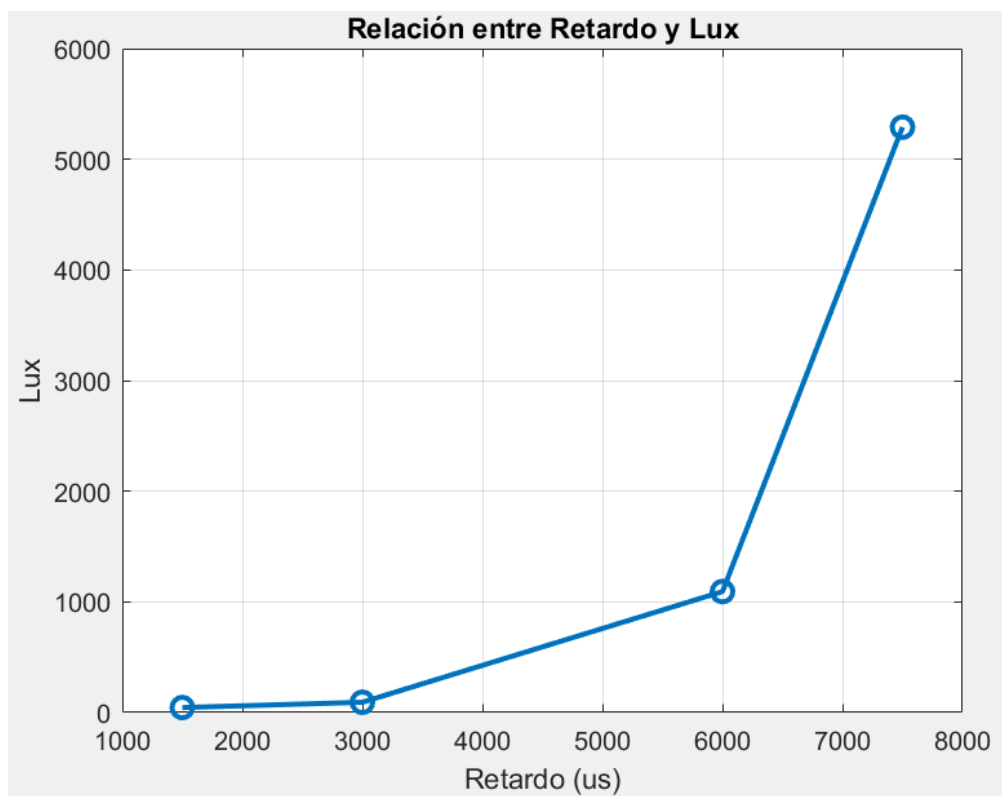


Ilustración 17 GRAFICA de relación lux-retardo
Fuente: Elaboración propia.

Referencias

- aguilar, e. (2020). *Los sistemas domoticos controlados por voz. El caso de los asistentes inteligentes*. Zaragoza.
- Aguilera, H. (2010). La basura electronica y la contaminacion ambiental. *Enfoque UTE*, 46-61.
- "Desarrollo backend para aplicaciones web, Servicios Web Restful:Node.js vs Spring Boot". (2018). *Sist. e Tecnologias de Informa*, 309-321.
- ESP32. (20 de octubre de 2024). Obtenido de ESPRESSIF: <https://www.espressif.com/en/products/socs/esp32>
- Gutierrez Sanin, F. M. (2019). ¿Obsolescencia programada? La implementacion de la sustitucion y sus inconsistencias. *Analisis Politico*, 136-160.
- Ir, G. v. (24 de 5 de 2024). *DYNA*. Obtenido de Luz y Salud: <https://revista-dyna.com/index.php/DYNA/article/view/5673>
- Ornelas, j. c. (2019). Variacion del angulo de disparo de un sistema senoidal para desarrollar un sistema electrico de ahorro de potencia. Toluca: Ciateq.
- Pineda Fajardo, L. M. (16 de 7 de 2021). *Desarrollo de un prototipo de sistema mecanico para el desplazamiento de una persona con movilidad reducida*. Obtenido de <http://hdl.handle.net/20.500.12749/20175>

