

Comando remoto de electrodomésticos a través de internet

Ismael L. Becker ^{a,*}, Mateo N. Stehr ^a, Juan P. Gross^{a,b}, Guillermo A. Fernández^{a,b}

^aFacultad de Ingeniería (FI), Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina.

^bGrupo de Investigación y Desarrollo en Electrónica (GIDE), Instituto de Materiales de Misiones (IMAM), Facultad de Ingeniería (FI), Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina.
e-mails: ismaelbecker90@gmail.com, matistehr@gmail.com, gross@fio.unam.edu.ar,
guillermo.fernandez@fio.unam.edu.ar

Resumen

El presente trabajo describe el desarrollo de un sistema embebido cuyo propósito es permitir al usuario controlar de forma remota, por medio de un robot (bot) de *Telegram*, electrodomésticos por medio de señales infrarrojas que imitan al control original. El prototipo desarrollado se compone de un microcontrolador con conectividad Wi-Fi, un sensor IR para aprender los comandos y un emisor IR para comandar los electrodomésticos a controlar. El usuario interactúa con el dispositivo por medio de un chat en *Telegram*, y el microcontrolador recibe los comandos a ejecutar. El trabajo implicó el diseño y validación de la idea por medio de simulación, posteriormente se implementó el prototipo en placa de prueba por último se confeccionó el circuito impreso (PCB). Finalmente se verificó el funcionamiento del prototipo funcional ensamblado, logrando cumplir con el objetivo del trabajo.

Palabras Clave – Automatización del hogar, electrodomésticos, control remoto, internet de las cosas IoT, sistemas embebidos.

1. Introducción

El presente trabajo se desarrolla como actividad integradora de la asignatura “Sistemas Embebidos” de la carrera Ingeniería en Computación, dictada en la Facultad de Ingeniería de la UnaM. El objetivo de dichas actividades es contribuir a desarrollar las habilidades de diseño electrónico y de software al integrar los conocimientos adquiridos en la asignatura junto con el resto de la carrera, de forma acotada y limitada al alcance de una asignatura de grado.

El control de electrodomésticos por medio de señales infrarrojas es una tecnología madura que lleva décadas en el mercado. Pero su limitación es que el control es local, el transmisor debe estar en el mismo ambiente que el dispositivo a comandar. Actualmente la tecnología (y el bajo costo de acceso a la misma) está llevando los mecanismos de control a un entorno conectado a Internet y que permite el control remoto desde cualquier lugar con conectividad a la red. A su vez, se ha desarrollado tecnología de control que permite interactuar con programas por medio de sistema de chat, acortando la brecha de uso de usuarios normales y la curva de aprendizaje de uso de esta.

Por tanto, en este trabajo se propone un sistema embebido diseñado para el comando remoto de electrodomésticos mediante el envío de señales infrarrojas a través de Internet. El sistema permite que los usuarios puedan interactuar con él mediante un bot de *Telegram*, ya sea para enviar comandos de encendido, de apagado, cambios de canal o temperatura, como así también para cualquier comando IR en general. A diferencia de otras soluciones que requieren aplicaciones móviles propietarias, este proyecto busca la simplicidad en el uso, sin sacrificar funcionalidad ni seguridad.

2. Sistema desarrollado

2.1. Relevamiento Técnico

En esta etapa se llevó a cabo un relevamiento técnico orientado a establecer los requerimientos funcionales del dispositivo a diseñar. Se definió que el sistema debía ser capaz de emitir señales infrarrojas compatibles con distintos modelos de electrodomésticos, principalmente televisores y aires acondicionados, replicando fielmente los comandos enviados por los controles remotos originales. Adicionalmente, se estableció que la interfaz de usuario debía estar basada en la plataforma de mensajería Telegram, permitiendo el envío de comandos remotos y la configuración del dispositivo sin la necesidad de aplicaciones externas ni interfaces físicas adicionales. Con estos requerimientos definidos, se diseñó un circuito base y se seleccionaron los componentes electrónicos adecuados al mismo.

Entre los elementos seleccionados para el diseño fueron:

- ESP32: fue seleccionado por su disponibilidad, costo y hardware disponibles, cuenta con acceso a redes Wi-Fi integrado lo que facilitó la conectividad. Se programa por medio del entorno Arduino IDE [1], en lenguaje C.
- Receptor TSOP1738: representado en la Fig. 1, es un módulo integrado (receptor y electrónica de demodulación) para recepción de comunicaciones infrarrojas que tiene capacidad de demodular diferentes protocolos estándar. Se seleccionó por la disponibilidad y la capacidad de soportar múltiples protocolos incrementando su compatibilidad.
- Librería CTBot [2] proporciona una interfaz sencilla para interactuar con la API de Telegram mediante solicitudes HTTP.
- Librería IRremote [3] ofrece una solución completa para la codificación y transmisión de señales IR, soportando una variedad de protocolos comerciales como NEC, Sony, RC-5, entre otros.

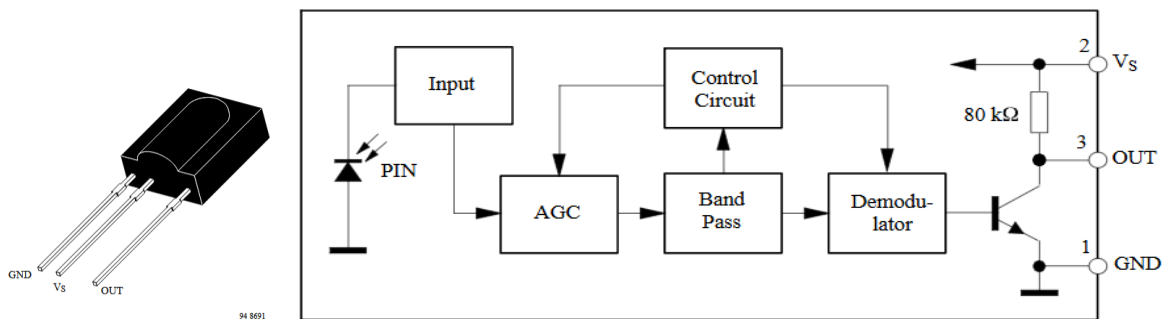


Fig. 1. Componentes y diagrama interno del TSOP 1738 [4].

2.2. Desarrollo del Firmware

El desarrollo del software se separó en etapas

- Validación de funcionamiento de las comunicaciones infrarrojas: a nivel de hardware y software.
- Desarrollo del bot en Telegram

2.2.1. Validación de funcionamiento de las comunicaciones infrarrojas

Se hicieron pruebas de funcionamiento de las librerías IRremote por medio de una implementación sencilla de un transmisor y receptor, como se representa en la Fig. 2, comprobando el correcto envío de datos entre los mismos. Se hicieron pruebas separadas con el ESP32 para validar el funcionamiento en modo de recepción y transmisión de datos de forma individual. Las mismas permitieron confirmar la correcta temporización de la señal IR, la estabilidad de la frecuencia portadora, y la fidelidad en la reconstrucción de las tramas. Se utilizó una placa Arduino UNO de forma accesoria para las pruebas.

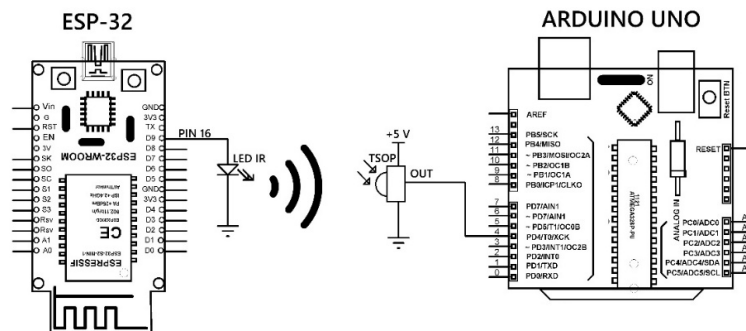


Fig. 2. Diagrama del circuito de prueba de emisión IR del ESP32.

2.2.2. Desarrollo del robot en Telegram

En esta etapa se integró la capacidad de conexión Wi-Fi del ESP32 con la biblioteca CTBot, configurando el *token* del bot, los identificadores de chat y los comandos disponibles para el usuario. Esta biblioteca se utilizó para establecer una interfaz de comunicación entre el ESP32 y Telegram, permitiendo controlar el ESP32 con comandos de texto. La configuración inicial del bot incluye el *token* de autenticación y los datos de red Wi-Fi para habilitar la conexión. El ESP32 es capaz de interpretar mensajes entrantes y asociarlos a distintas acciones, como el envío de señales almacenadas, la captura de nuevos códigos IR y su registro con nombres personalizados. El sistema también gestiona estados para guiar la interacción con el usuario de forma dinámica y secuencial.

Se diseñó una lógica de procesamiento que interpreta los mensajes entrantes desde la plataforma de mensajería, asociándolas a funciones específicas del sistema como la transmisión de señales infrarrojas, la carga de nuevos códigos IR y la consulta del estado del dispositivo. El diagrama de flujo del código desarrollado, presentado en la Fig. 3, representa la lógica de funcionamiento.

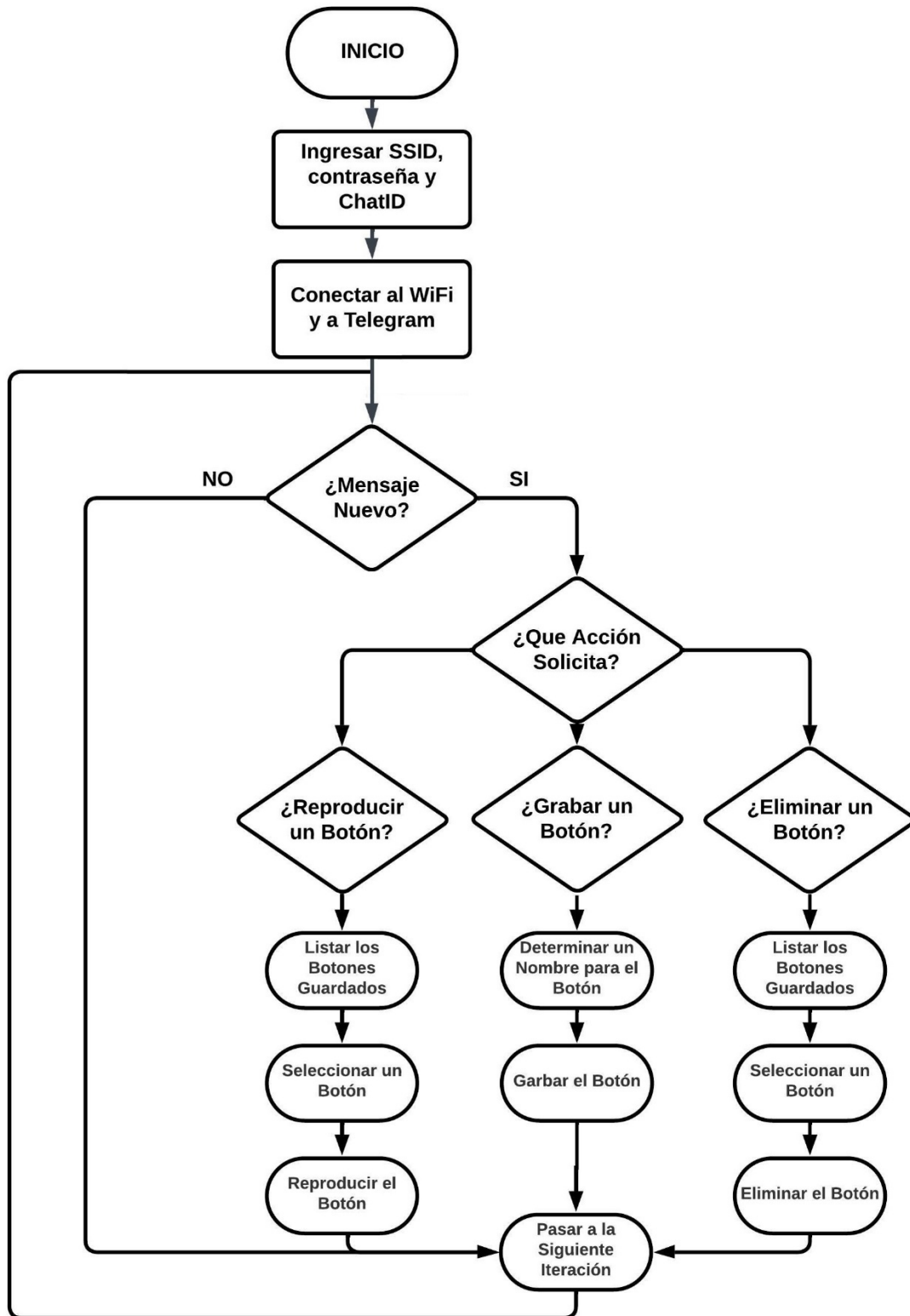


Fig. 3. Diagrama de flujo del *Firmware* desarrollado.

2.3. Funcionamiento del programa

El funcionamiento básico del programa se desglosa en dos etapas:

- Configuración Inicial
- Funcionamiento Normal

2.3.1. Configuración inicial

En esta etapa, el microcontrolador inicia un modo de configuración autónoma, que consiste en generar una red local inalámbrica por medio de su punto de acceso integrado (AP). EL usuario se conecta a la misma, y accediendo a una página web que reside dentro del dispositivo (cuya IP es 192.168.4.1/24), ingresa las credenciales de conexión a la red y el código del chatID del bot Telegram donde se lo usará en modo funcionamiento normal.

2.3.2. Configuración inicial

Luego de la configuración inicial, el sistema entra en el modo de uso normal, donde se conecta a la red Wi-Fi configurada y a la espera de comandos por el chat de Telegram. El flujo normal del programa es esperar a recibir un comando por el chat, en caso de hacerlo, se procesa el mismo.

Los comandos posibles se pueden agrupar en 3 tipos de acciones relacionadas:

- Grabar el código de un nuevo botón, presentado en la Fig. 4(a).
- Reproducir el código de un botón, presentado en la Fig. 4(b).
- Eliminar un botón existente, presentado en la Fig. 4(c).

El programa lleva una lista con los botones grabados, al momento de grabar uno nuevo se le solicita al usuario ingresar un identificador para el registro del botón en la lista interna y evitar superposiciones. Una vez recibido el nombre, el sistema inicia el proceso de grabación, el cual puede implicar la lectura de una señal, la detección de un código, o el almacenamiento de un patrón. Al finalizar esta grabación, el nuevo botón queda disponible para ser utilizado en futuras interacciones.

Al eliminar un botón, se le presenta al usuario una lista de botones existentes para elegir cual borrar, tras lo cual el sistema borra la información correspondiente, retirando ese botón tanto de la memoria como del listado accesible para acciones futuras. Luego de la ejecución de estos procedimientos, se ejecuta la siguiente iteración del bucle.

El sistema utiliza la clase Preferences del ESP32 para guardar los códigos IR capturados en la memoria *flash*. Cuando se recibe un código IR, el arreglo de datos que contiene la secuencia cruda de la señal, junto con su longitud, se guardan en la memoria bajo una clave única correspondiente al nombre del botón. Adicionalmente, se mantiene una lista de nombres de botones, que se actualiza dinámicamente para incluir nuevos botones si aún no están registrados. Esta lista se almacena como

una cadena separada por comas. Para recuperar un código guardado, se utiliza una función que accede a los datos usando el mismo nombre clave y carga nuevamente el contenido.

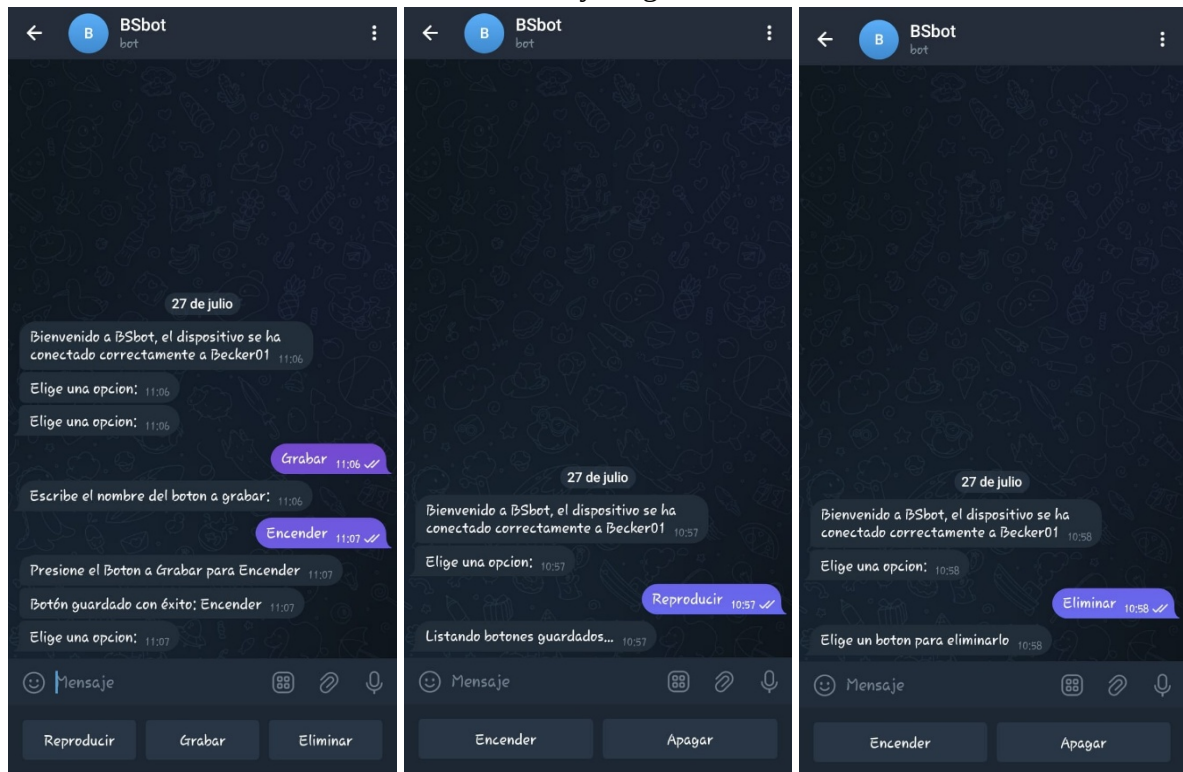


Fig. 4. Diferentes pantallas del chat: (a) Menú Principal, (b) Reproducir, (c) Eliminar.

2.4. Diseño del Circuito

2.4.1. Pantalla OLED

Se decidió incorporar una pantalla OLED al sistema con el fin de mejorar la interacción directa del usuario durante la configuración y el monitoreo local del dispositivo. A pesar de que el control principal se realiza mediante un bot de Telegram, la inclusión de la pantalla permite visualizar información crítica en tiempo real, como el estado de conexión, confirmaciones de grabación o reproducción de códigos infrarrojos, e incluso mensajes de error o diagnóstico. Esta funcionalidad resulta especialmente útil durante la fase de puesta en marcha o ante eventuales fallos de conectividad, permitiendo una identificación rápida del estado del sistema sin necesidad de herramientas externas.

2.4.2. Etapa de Potencia

Si bien en las pruebas preliminares el ESP32 demostró ser capaz de emitir correctamente las infrarrojas de comando, su alcance era limitado. Por este motivo fue necesario incorporar una etapa de potencia con una configuración Darlington, que permitió polarizar los LEDs con una corriente

pulsante de hasta 1 A, varios órdenes de magnitud mayor a la capacidad de corriente de salida del ESP32.

Cabe mencionar que los LEDs infrarrojos utilizados se polarizan con 2 V y soportan corrientes elevadas de 1 A por periodos de hasta 1 ms, y esto encaja bien dentro de las necesidades del sistema pues las tramas IR utilizados, son trenes de pulsos con duraciones en el orden de los microsegundos [5].

Para la etapa de potencia, cuyo circuito se expone en la Fig. 5, se utilizaron dos transistores, ambos capaces de soportar un voltaje mayor a 45 V, lo que es más que suficiente incluyendo factores de seguridad. Para el primer transistor de la configuración Darlington se utilizó un BC337 [6] que posee una ganancia mínima de 60 y soporta una corriente de colector de hasta 500 mA y para el segundo transistor en la configuración se utilizó un BD139 [7] debido a su mayor soporte de corriente de colector, y ganancia mínima de 25.

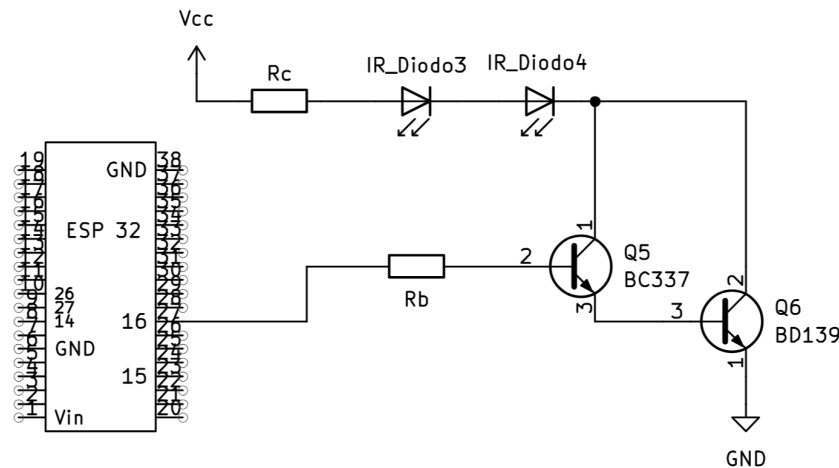


Fig. 5. Circuito de la etapa de potencia.

En primer lugar, se realizó un cálculo para los valores de la Resistencia de colector, donde V_{carga} es la tensión de 2 LEDs infrarrojos en serie:

$$R_c = \frac{V_{cc} - V_{carga} - V_{CE(sat)}}{I_c} \quad (1)$$

$$R_c = \frac{15V - 2V - 2V - 1,2V}{1A} = \frac{9,8V}{1A} = 9,8\Omega \quad (2)$$

Y luego para la resistencia de base:

$$R_b \leq \frac{V_{\square} - V_{BE1(on)} - V_{BE2(on)}}{I_b} = \frac{3V - 1,2V - 1V}{20mA} = 40\Omega \quad (3)$$

Para adaptarse a los valores comerciales se utiliza una resistencia de colector de 10Ω con una disipación de 1 W y para la resistencia de base se utilizaron dos resistencias de 47Ω en paralelo

para lograr una resistencia menor a 40Ω , esto se realizó por disponibilidad por lo que una opción válida también es utilizar una resistencia de 39Ω .

Para verificar el funcionamiento se realizan diversas mediciones con el osciloscopio emitiendo códigos infrarrojos y se demostró que la corriente que circula por los LEDs es de 880 mA. Este valor no llega al valor de 1 A calculado debido a que la fuente es capaz de entregar únicamente 1 A, y la diferencia entonces radica en el consume utilizado para alimentar el ESP32.

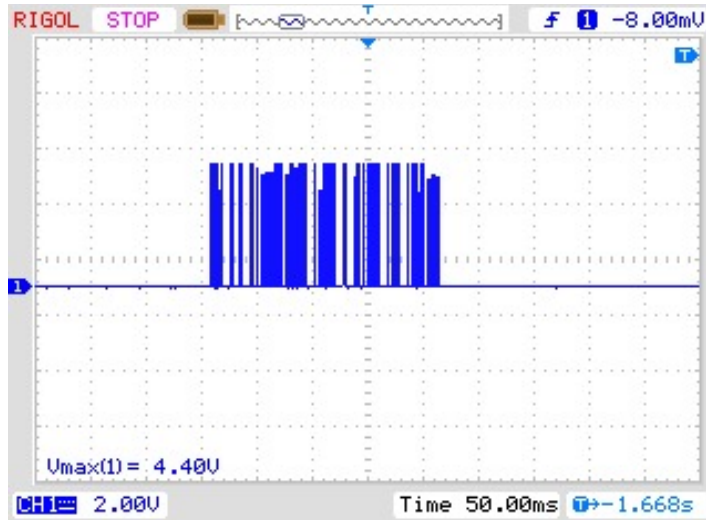


Fig. 6. Medición con el Osciloscopio del Voltaje en la Resistencia de Colector.

Sin embargo, con esta corriente las distancias verificadas son mayores que las utilizadas por la mayoría de los controles remotos tradicionales, por lo que la potencia de emisión obtenida se consideró adecuada, validando la configuración para el diseño final del circuito. Los resultados de las mediciones de tensión sobre el resistor de colector se presentan en la Fig. 6, y permiten determinar tanto la forma de los pulsos emitidos, como también la corriente que atraviesa los LEDs IR.

2.4.3. Fuente de Alimentación

Uno de los requisitos de la actividad integradora es diseñar la fuente de alimentación. Los resultados obtenidos se exponen en la Fig. 7.

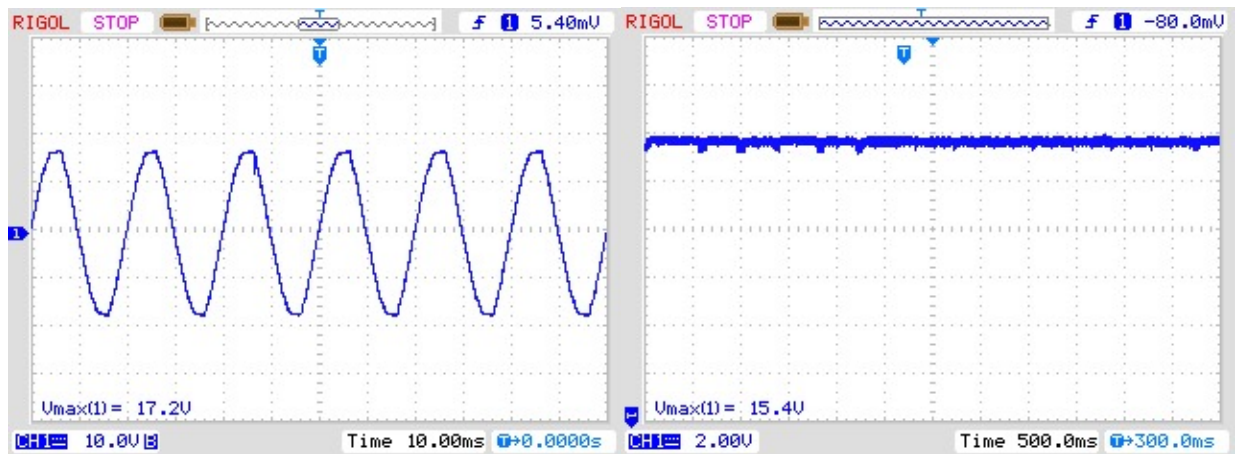


Fig. 7. Medición con el Osciloscopio del Voltaje anterior y posterior al filtro capacitivo.

Se partió de un transformador 220 V/12 V 1 A, cuya tensión de salida es rectificadora por medio de un puente de *Graetz* y se disminuyó lo más posible el rizado utilizando un filtro capacitivo conectando un capacitor de 1000 μ F en paralelo.

El voltaje a la salida del filtro es superior a la alimentación de 5 V, necesaria para alimentar el ESP32. El primer enfoque fue utilizar un regulador de voltaje lineal como un LM7805 [8], sin embargo, esta alternativa fue descartada ya que los requerimientos de corriente del ESP32, (según las mediciones realizadas), implican un total aproximado de 120 mA, lo que representa una disipación de potencia importante sobre el regulador lineal, lo que, sumado a la instalación en un gabinete cerrado y con refrigeración pasiva, demostró no ser una alternativa viable para implementar.

Por este motivo se decidió utilizar un regulador conmutado integrado del tipo step-down (buck converter), en este caso el LM2596 [9], que debido a su alta eficiencia y capacidad para manejar corrientes elevadas disipa una cantidad ínfima de calor en comparación con un regulador lineal.

2.5. Elaboración del PCB y montaje en gabinete

El diseño del circuito completo se expone en la Fig. 8, donde se detalla la composición del mismo.

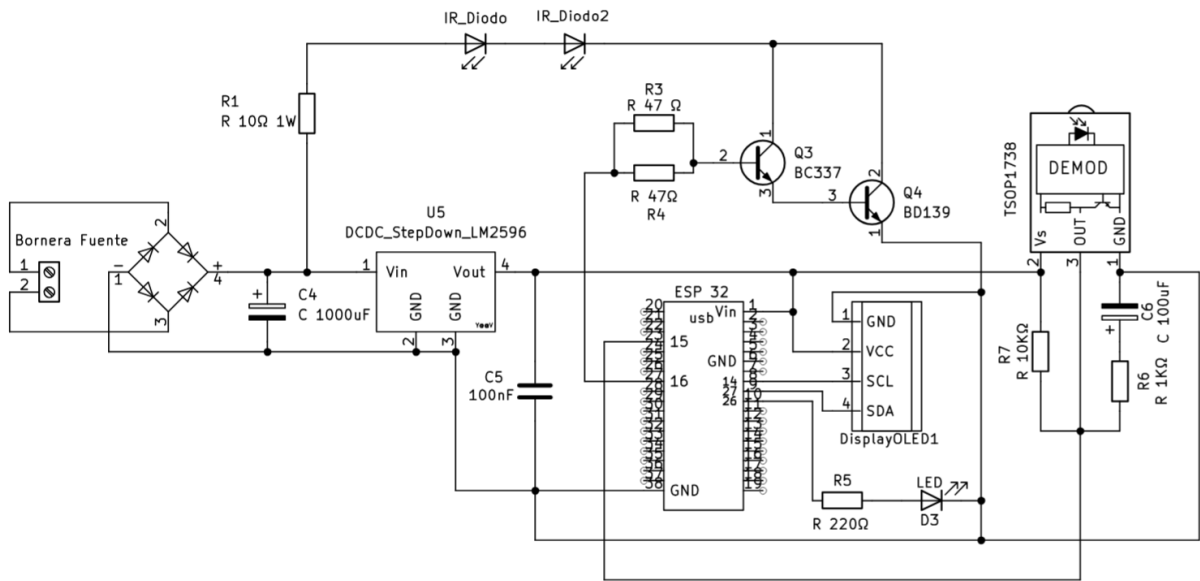


Fig. 8. Circuito completo.

A partir de este circuito se diseñó el circuito impreso de la Fig. 9, utilizando KiCAD [10].

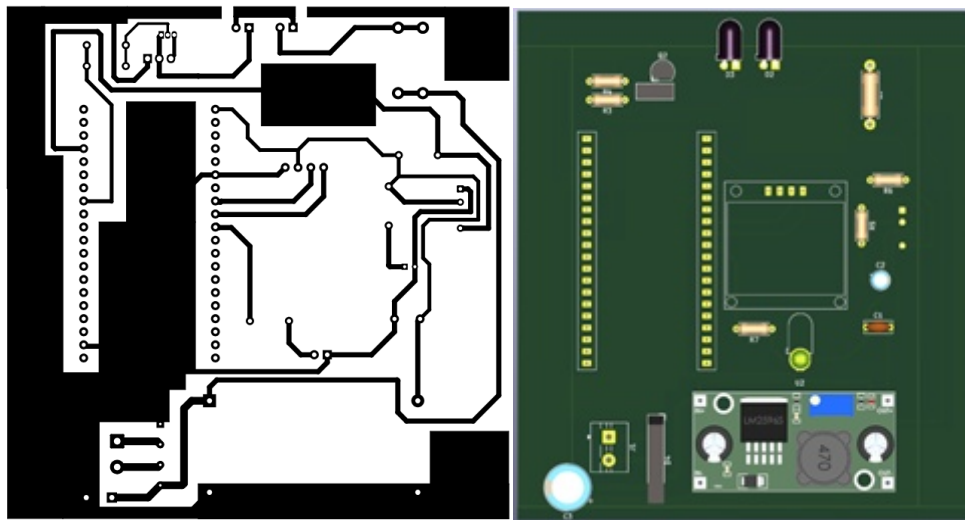


Fig. 9. PCB diseñada en KiCAD.

La placa se diseñó con dimensiones de 10 x 10 cm, a fin de montarla en una caja estanca plástica de 11,5 x 16,5 x 11 cm. Se dejaron los LEDs infrarrojos sobresaliendo de los límites de la placa, ya que, al montar la placa en la caja, estos sobresalen por unas muescas circulares realizadas en la tapa.

3. Resultados

La placa del circuito, con todos los componentes insertados y soldados, se presenta en la Fig. 10.

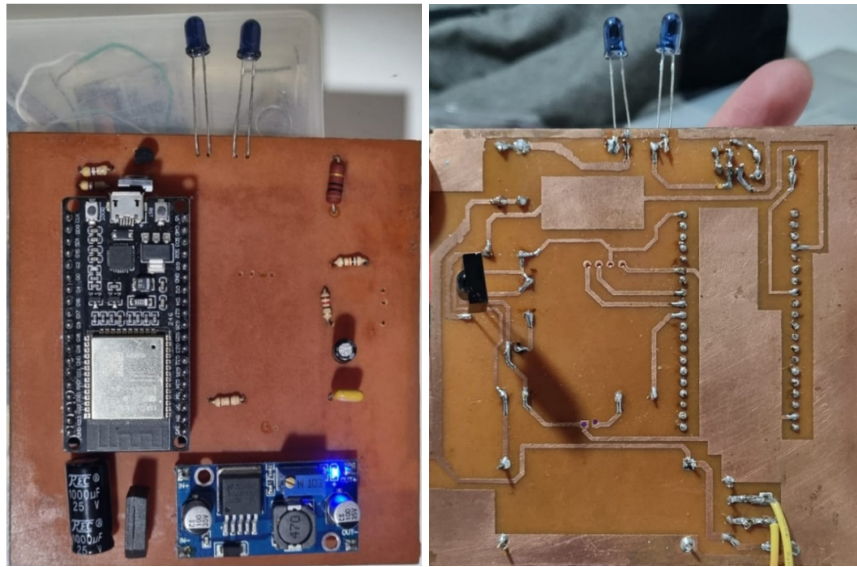


Fig. 10. PCB terminada, con todos los componentes insertados.

El circuito montado sobre el gabinete se expone en la Fig. 11.



Fig. 11. Prototipo final encapsulado.

Se realizó un video de demostración en el cual se muestra el funcionamiento del dispositivo desarrollado.

Este puede consultarse en <https://youtu.be/Fu6TFzSXKNs> [11].

4. Conclusiones

El presente trabajo integrador logró el desarrollo exitoso de un sistema embebido orientado al control remoto de electrodomésticos mediante la emisión de señales infrarrojas, operado a través de

un bot de Telegram como interfaz principal. A partir de la integración entre el microcontrolador ESP32, las bibliotecas IRremote y CTBot, y una etapa de potencia optimizada, fue posible alcanzar un sistema robusto, funcional y con un alcance superior al de controles remotos tradicionales.

Las pruebas realizadas permitieron verificar la fiabilidad de la transmisión IR y la correcta interpretación de comandos por parte del microcontrolador, así como la estabilidad en la comunicación con el usuario mediante Internet. La utilización de un regulador step-down en reemplazo de reguladores lineales permitió una mejora significativa en la eficiencia térmica del sistema, garantizando una alimentación segura en condiciones de uso continuo.

En el diseño se contemplaron múltiples aspectos aprendidos a lo largo de la carrera, como la selección adecuada de componentes, la simulación previa en software especializado, el diseño y elaboración del PCB, y la implementación de una carcasa funcional. Se destacó, además, la importancia de interpretar correctamente hojas de datos y dimensionar correctamente los componentes críticos del circuito, en especial aquellos involucrados en la etapa de potencia y disipación.

La posibilidad de operar el sistema de forma remota, sin necesidad de aplicaciones propietarias ni interfaces complejas, representa una solución accesible, versátil y escalable dentro del ámbito de la domótica, con potencial para adaptarse a distintos entornos y usuarios.

Como mejora futura, se propone optimizar la interfaz de usuario del bot, incluyendo un menú interactivo o sistema de respuestas guiadas que facilite aún más la interacción. También sería deseable incorporar una base de datos de comandos IR ya preconfigurados para distintos dispositivos, simplificando el proceso de aprendizaje del sistema.

En definitiva, el sistema desarrollado cumple con los objetivos planteados y refleja la correcta aplicación de los conocimientos adquiridos, constituyendo una valiosa experiencia formativa y un aporte concreto en el área de sistemas embebidos aplicados al hogar inteligente.

Referencias

- [1] **Arduino**, “Arduino IDE,” *Arduino.cc*, [Software] [Online]. Available: <https://www.arduino.cc/en/software/>
- [2] **S. Cirillo**, *CTBot: Telegram Bot library for ESP8266*, *GitHub*, [Online]. Available: <https://github.com/shurillu/CTBot>
- [3] **Armin Joachimsmeier**, *Arduino-IRremote*, *GitHub*, [Online]. Available: <https://github.com/Arduino-IRremote/Arduino-IRremote>
- [4] **Vishay Telefunken**, *TSOP1730 – Photo Module for PCM Remote Control Systems* [Imagen], Rev. 10, Document No. 82030, Apr. 2, 2001. [Online]. Available: <https://www.alldatasheet.com/datasheet-pdf/view/26589/VISHAY/TSOP1738.html>
- [5] **Everlight Electronics Co, Ltd.**, *IR333-A Infrared Emitting Diode*, *Datasheet*, [Online]. Available: https://cdn-shop.adafruit.com/datasheets/IR333_A_datasheet.pdf
- [6] **Mouser Electronics**, *BC-337 NPN Amplifier Transistor Datasheet*, Fairchild Semiconductor, rev. 1.5, published ca. 2015. Available at: <https://www.mouser.com/datasheet/2/149/BC337-193546.pdf>

- [7] **STMicroelectronics**, BD139 NPN Epitaxial Silicon Transistor Datasheet Rev. 5, May 2008. Available at: <https://www.st.com/resource/en/datasheet/bd139.pdf>
- [8] **Texas Instruments**, *LM340, LM340A and LM78xx Wide VIN 1.5-A Fixed Voltage Regulators*, Rev. K, Jul. 2016. [Online]. Available: <https://www.alldatasheet.com/datasheet-pdf/view/838007/TI1/LM7805.html>
- [9] **Texas Instruments**, *LM2596 SIMPLE SWITCHER® Power Converter 150 kHz, 3-A Step-Down Voltage Regulator Datasheet*, Rev. G, SNVS124G, Mar. 2023. [Online]. Available: <https://www.ti.com/lit/ds/symlink/lm2596.pdf>
- [10] **KiCad EDA**, “KiCad – A Cross Platform and Open Source Electronics Design Automation Suite,” [Software]. [Online]. Available: <https://www.kicad.org/>
- [11] M. Stehr e I. Becker, *Video de demostración del funcionamiento del dispositivo*, YouTube, Jul. 2025. [Online]. Available: <https://youtu.be/Fu6TFzSXKNs>