

Uso de ESP32-CAM para seguridad y vigilancia controlado por Telegram

Julio Dario Solis Leyes, Roberto M. Murdocca, Sergio F. Hernandez Velazquez
 Departamento de Electrónica
 Universidad Nacional de San Luis
 San Luis, Argentina
 e-mail: {leyessolis, mmurdocc, sergio.sfhv}@gmail.com

Resumen - En este trabajo se presenta la implementación de un sistema de seguridad y vigilancia controlado remotamente mediante la plataforma Telegram. El sistema permite gestionar funciones como la captura y envío de imágenes ante la detección de movimiento, la rotación de la cámara, el encendido y apagado de luces y la activación de una sirena utilizando un ESP32-CAM.

Palabras clave: Seguridad y Vigilancia, Telegram, MEF, ESP32-CAM.

I. INTRODUCCIÓN

En un mundo cada vez más digitalizado, la seguridad es una preocupación clave para hogares y negocios. Los métodos tradicionales de vigilancia tienen limitaciones en cuanto a accesibilidad remota y control en tiempo real. Si bien existen sistemas de monitoreo remoto para seguridad y vigilancia modernos con muchas prestaciones, el costo es excesivo para muchos usuarios. Por este motivo se propone el diseño de un sistema modesto, de bajo costo y con posibilidades de ampliaciones futuras.

La demanda de soluciones de seguridad accesibles y avanzadas ha impulsado el desarrollo de sistemas que combinan hardware eficiente con plataformas de comunicación modernas. Los avances tecnológicos en comunicación inalámbrica, sensores inteligentes y procesamiento de datos han permitido la creación de sistemas integrados que operan en tiempo real, ofreciendo vigilancia dinámica y respuestas automáticas a eventos, como la detección de movimiento. Asimismo, la popularización de plataformas de comunicación como Telegram ha abierto nuevas posibilidades en el control remoto de sistemas de seguridad.

Utilizar aplicaciones móviles para gestionar dispositivos conectados ha demostrado ser un enfoque eficiente y conveniente, alineándose con la tendencia hacia la domótica y el Internet de las Cosas (IoT). Estas herramientas ofrecen una interfaz básica e intuitiva, que permite a los usuarios monitorear y controlar sus sistemas de seguridad desde cualquier parte del mundo, reforzando la sensación de control y seguridad.

Este proyecto aborda esa necesidad mediante el desarrollo de un sistema de seguridad y vigilancia remoto, controlado íntegramente a través de la popular plataforma de mensajería Telegram. La solución presentada se enfoca en ofrecer a los usuarios un control total sobre la seguridad de sus entornos, permitiéndoles gestionar de manera remota diversas funcionalidades como el encendido y apagado de luces, la captura de imágenes en tiempo real, y la rotación precisa de una cámara de vigilancia. Además, se incorporan dos modos de operación: el modo PIR, que utiliza un sensor de infrarrojos pasivo para la detección de movimiento y la captura inmediata de imágenes, y el modo alarma, que refuerza la seguridad activando una sirena y luces intermitentes.

II. DESCRIPCIÓN

El sistema propuesto está conformado por una placa de desarrollo ESP32-CAM, un motor paso a paso para posicionar la cámara, un sensor PIR para la detección de movimiento, una sirena, un módulo de luces LED, un módulo de potencia con relé, un modulo DC-DC Step Down y demás componentes auxiliares. En la Fig. 1 podemos ver un diagrama en bloques del sistema y en la Fig. 2 se muestra el circuito completo del sistema.

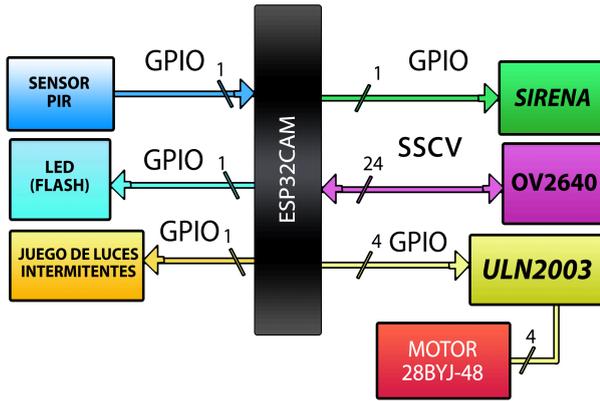


Fig. 1. Diagrama en bloques del sistema

A. HARDWARE

El núcleo del sistema está formado por una placa ESP32-CAM la cual consiste en una placa de desarrollo compacta que integra un microcontrolador ESP32 doble núcleo con Wi-Fi y Bluetooth, una cámara OV2640 capaz de capturar imágenes de alta resolución. Cuenta además, con una ranura para tarjeta microSD, 9 pines GPIO disponibles, y es programable con Arduino IDE y otras plataformas. Para el movimiento de la cámara se utiliza un motor paso a paso modelos 28BYJ-48 y como driver para este último un ULN2003. La sirena se excita desde un módulo rele. Las luces se implementaron con un módulo LED. Todo el sistema se energiza con 12V y se utiliza un convertor DC-DC Step Down para regular los 5V para la ESP32, motor paso a paso y sensor PIR. Todo este conjunto se montó en una caja impresa en 3D para este trabajo. En la Fig. 3 podemos ver el housing diseñado para albergar todo el hardware.

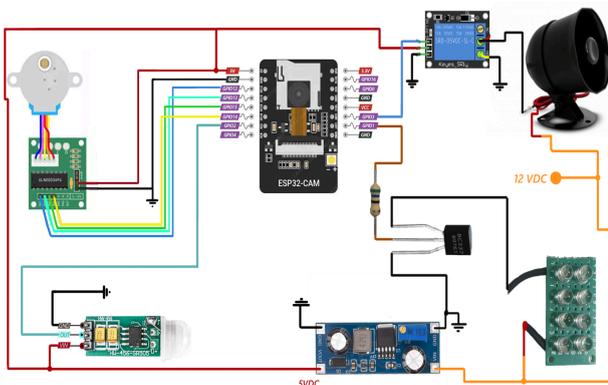


Fig. 2. Diagrama en bloques del sistema

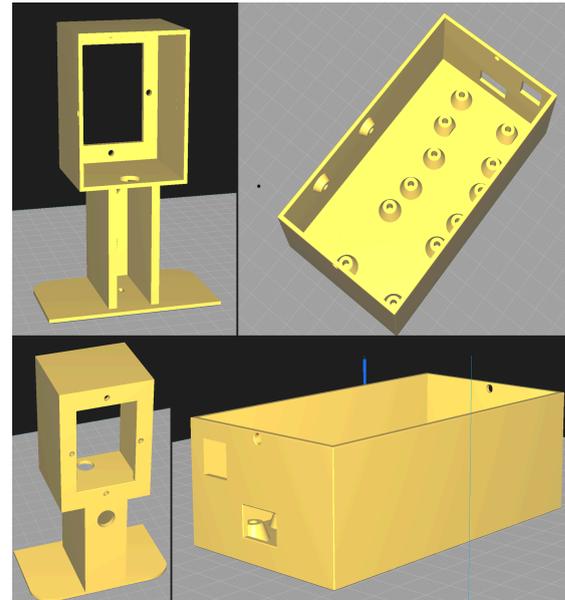


Fig. 3. Housing del proyecto.

B. SOFTWARE

La gestión del software del sistema se modela mediante una Máquina de Estados Finitos (MEF). La MEF detalla cómo se gestionan las tareas como la captura de imágenes, la rotación de la cámara y el cambio de modos de funcionamiento en respuesta a las condiciones y comandos recibidos. En la Fig. 4 podemos ver dicha MEF.

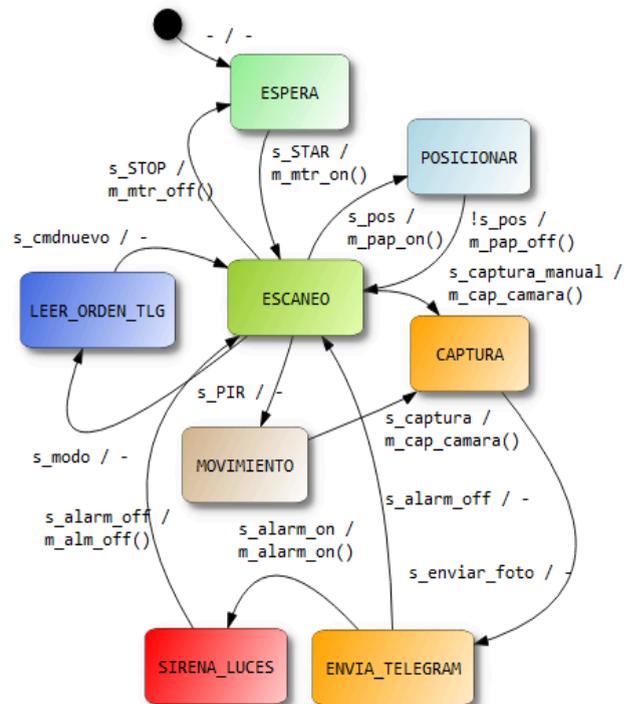


Fig. 4. Máquina de estados finitos (MEF)

En la MEF tenemos los siguientes estados:

- **Escaneo:** El estado principal donde el sistema monitorea continuamente en busca de eventos.
- **Captura:** Cuando se solicita una foto, el sistema toma la captura.
- **Posicionar:** En este estado se ajusta la posición de la cámara y luego vuelve a **Escaneo**.
- **Movimiento:** El sistema detecta movimiento con el sensor PIR y realiza la captura de imagen.
- **Sirena y Luces:** En este estado se activan la sirena y el módulo LED para alertar sobre amenazas detectadas.

III. FUNCIONAMIENTO

Inicialmente, las pruebas de funcionamiento del sistema se centraron en evaluar la eficiencia de la captura de imágenes y su posterior envío a través del servidor de Telegram. Se pudieron identificar y corregir problemas, como los retrasos en la transmisión de imágenes y los envíos erróneos que ocurrían al encender el sistema.

El sistema posee un modo de trabajo denominado PIR. En este modo se combina la funcionalidad de captura de imágenes con la lógica de detección basada en un sensor de infrarrojos pasivo (PIR). Las pruebas en este contexto se orientaron a validar que las imágenes solo se capturaron y enviaran cuando el modo PIR estuviera activado por el usuario y el sensor detectara movimiento de manera simultánea.

Para ello, se energizó el sistema con el modo PIR activo y se activó el sensor PIR en intervalos controlados mediante la detección de presencia, garantizando así la correcta sincronización entre la detección de movimiento y el envío de imágenes.

Para el posicionamiento de la cámara se utilizó un motor paso a paso 28BYJ-48, gestionado a través del controlador ULN2003. Se probó al principio un servomotor ya que se disponía de uno, pero él mismo hacía bastante ruido y esto podría alertar a los intrusos.

Se tuvo que tener en cuenta los límites de giro del motor y se realizó el almacenamiento en memoria no volátil de la última posición del motor para que en caso de corte de energía, al

retomar la misma el motor retome su posición previa, evitando errores derivados de la suposición de una posición inicial neutral o incorrecta, que podrían comprometer la precisión del control de los límites de giro.

La capacidad del sistema para identificar y mantener la posición de giro del motor es esencial para prevenir daños físicos, como la tensión excesiva o el corte de los cables que conectan la placa de desarrollo situada en la carcasa superior con los módulos alojados en la carcasa inferior. Este control preciso asegura la integridad de las conexiones y el correcto funcionamiento del sistema en su conjunto, garantizando la fiabilidad y la longevidad del equipo.

El sistema requiere la conexión con un servidor de Telegram, enlazado con un bot que utiliza encriptación para garantizar la máxima seguridad. Solo los usuarios autorizados, que posean el token correspondiente, tendrán acceso al sistema de seguridad. Una vez que tanto el sistema como el dispositivo del usuario estén conectados a internet, podrán interactuar de manera remota. El usuario tendrá acceso a una mini guía de uso y podrá enviar comandos al sistema desde cualquier lugar, asegurando un control eficiente y seguro.

IV. RESULTADOS

La implementación del sistema se llevó a cabo en una caja diseñada para este fin a través de impresión 3D como se muestra en la Fig. 6. La impresión fue realizada en dos partes. En la parte inferior de esta se montaron los circuitos de alimentación y el driver del motor. En la parte superior se colocó el ESP32, la cámara, el motor PAP y el sensor PIR. Dado que el microcontrolador se encuentra en la parte superior de la carcasa, fue necesario que todos los módulos ubicados en el gabinete inferior estén conectados a este a través de cables de longitud adecuada. Esto es crucial para garantizar que dichos cables no estén sometidos a tensiones mecánicas excesivas durante la rotación de hasta 180 grados que puede efectuar el sistema, evitando así posibles fallas debido al desgaste o desconexión accidental.

En la Fig. 6 se puede ver una foto del prototipo en su versión final. Las pruebas realizadas en este primer prototipo estuvieron a

la altura de los objetivos planteados al principio del trabajo. A modo de ejemplo en la Fig. 7 podemos ver una captura enviada a Telegram ante la detección de intrusos. Además el usuario puede interactuar con el sistema a través de un menú implementado en Telegram mediante un bot. Este bot posee comandos que permiten hacer capturas, encender la sirena, encender las luces y posicionar la cámara.

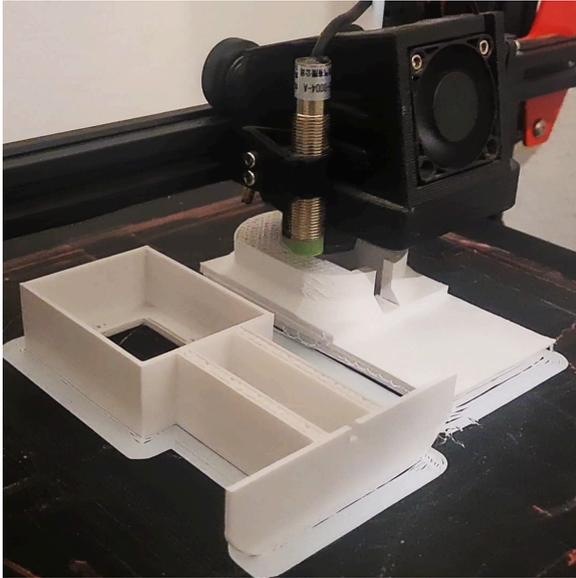


Fig. 5: Detalle de la carcasa.

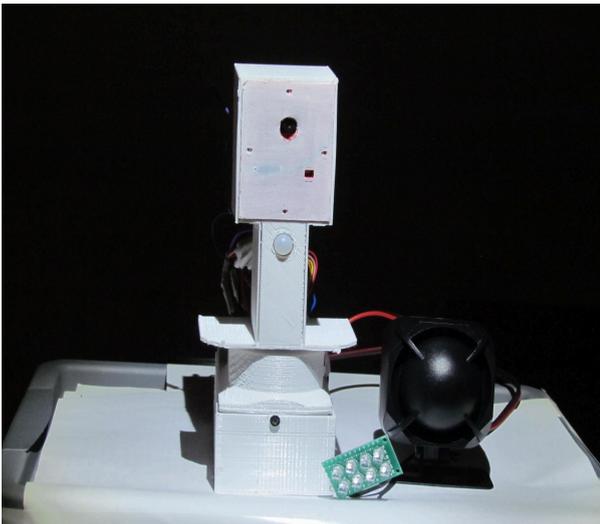


Fig. 6 Dispositivo en su versión final.



Fig. 7: Detección de presencia.

V. CONCLUSIONES

En este proyecto se desarrolló con éxito un sistema integral de seguridad de bajo costo, ofreciendo un rendimiento sólido en términos de vigilancia y seguridad remota. Se logró implementar la detección de movimiento y la captura de imágenes para su envío a través de Telegram. Además, se integró un bot que permite al usuario interactuar con el sistema mediante un menú funcional de opciones.

Los objetivos propuestos al inicio del proyecto fueron alcanzados satisfactoriamente, resolviendo la problemática de seguridad planteada. Durante el proceso, se evidenció el potencial de la ESP32-CAM para realizar capturas y envíos de imágenes, lo cual sugiere nuevas posibilidades para explorar en futuras aplicaciones.

Entre las modificaciones futuras se considera el agregado de un sensor ranurado u óptico para tener una referencia de origen en caso de pérdida de pasos. Además, se prevé investigar sobre la incorporación de un pequeño algoritmo de inteligencia artificial para el seguimiento automático del objeto detectado. También se están realizando pruebas para habilitar la transmisión de video en tiempo real hacia un servidor, con el objetivo de grabar secuencias en caso de detección de presencia no deseada.

REFERENCIAS

- [1] Espressif Systems. (2024). *ESP32 datasheet*. Espressif Systems. Obtenido de https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [2] Manuals+. (s. f.). *ESP32-CAM Module Manual*. Manuals+. Recuperado el 15 de julio de 2024, de <https://manuals.plus/es/electronic-hub/esp32-cam-module-manual>
- [3] STMicroelectronics. (s. f.). *ULN2001–ULN2004: High voltage, high current Darlington transistor arrays [Datasheet]*. STMicroelectronics. Recuperado el 8 de mayo de 2024, de <https://www.st.com/resource/en/datasheet/uln2001.pdf>
- [4] Arduino. (s. f.). *Stepper* [Documentación de la biblioteca]. Arduino. Recuperado el 8 de mayo de 2024, de <https://www.arduino.cc/reference/en/libraries/stepper/stepper/>
- [5] Rapid Electronics. (s. f.). *28BYJ-48 Stepper Motor Datasheet*. Rapid Electronics. Recuperado el 8 de mayo de 2024, de https://static.rapidonline.com/pdf/78-4110_v1.pdf
- [6] SendPulse. (s. f.). *Cómo crear un chatbot para Telegram*. SendPulse. Recuperado el 2 de agosto de 2024, de <https://sendpulse.com/latam/knowledge-base/chatbot/telegram/create-telegram-chatbot>
- [7] Programar Fácil. (s. f.). *ESP32-CAM: Guía completa de uso y programación*. Programar Fácil. Recuperado el 8 de mayo de 2024, de <https://programarfácil.com/esp32/esp32-cam/>
- [8] Witnessmenow. (s. f.). *Universal Arduino Telegram Bot* [Repositorio en GitHub]. GitHub. Recuperado el 2 de agosto de 2024, de <https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot/tree/master>