

## Interfaz gráfica de usuario para el control de sistemas en la nube Graphical user interface for the could control of systems

G. H. López-Mera <sup>a</sup>, E. A. Padilla-García <sup>b</sup>, R. D. Cruz-Morales <sup>a</sup>, D. Tinoco-Varela <sup>a</sup>

<sup>a</sup>Departamento de Ingeniería. Facultad de Estudios Superiores Cuautitlán - UNAM. Cuautitlán Izcalli 54714, Edo. de México, México.

<sup>b</sup>Academia de Ingeniería en Robótica. Universidad Politécnica de Atlacomulco. Atlacomulco 50465, Edo. de México, México.

### Resumen

El crecimiento acelerado de la tecnología de la información ha extendido el panorama de diferentes aplicaciones que han sido desarrolladas para el Internet de las cosas (IoT), por ejemplo, aplicaciones de monitoreo o manipulación remota de sistemas. Por este motivo, es importante desarrollar sistemas de información enfocadas a aplicaciones remotas donde los usuarios puedan interactuar, controlar dispositivos o replicar diversas tareas. En este trabajo se presenta el desarrollo de una interfaz gráfica de usuario (GUI) en Python, dicha interfaz adopta la filosofía de trabajo de código abierto y utiliza widgets de la librería Tkinter. La interfaz es utilizada en un sistema de control en lazo abierto y monitoreo de señales con un enfoque maestro-esclavo para un sistema electromecánico. Como validación experimental, se detalla el funcionamiento de la GUI en la interacción del entorno virtual con los elementos físicos del sistema.

*Palabras Clave:* diseño, monitoreo, control, interfaz gráfica de usuario, IoT.

### Abstract

The accelerated growth of the information of technology has been extended the overview of different applications that have been developed for the Internet of Things (IoT), for example, monitoring applications or remote management of systems. For this reason, it is important to develop information systems focused on remote applications where users can interact, control devices or replicate remote systems for other tasks. This paper presents the development of a graphical user interface (GUI) in Python, this interface adopts the open source work philosophy and uses widgets from the Tkinter library. The interface is used in an open-loop control system and signal monitoring with a master-slave approach for an electromechanical system. As an experimental validation, the operation of the GUI is shown with the interaction between the virtual environment and the physical elements of the system.

*Keywords:* Design, monitoring, control, graphical user interface, IoT.

### 1. Introducción

Con la aparición del IoT, el control remoto de sistemas a través de la nube, ha tenido gran interés en las áreas de domótica, automatización industrial, proyectos académicos, entre otros. Sin embargo, la necesidad de usar una GUI surge como una herramienta esencial y cómoda, para que el usuario tenga a su mando el monitoreo y control remoto de alguna actividad que se esté realizando en la nube, por ejemplo, las variables de posición de sistemas rotativos, el encendido y/o apagado de sistemas electrónicos son algunas de ellas.

En este trabajo, el uso de la tecnología IoT se aborda mediante el diseño de una GUI en Python que involucra el monitoreo de variables físicas y el control de motorreductores en un

sistema esclavo a través de la comunicación de la plataforma Blynk. Algunos sistemas de monitoreo que utilizan este enfoque se han estudiado en (Muñoz, 2021) donde se propone un diseño y simulación de un sistema de monitoreo para un proceso de control de temperatura mediante IoT a nivel empresarial. En (Muñoz, 2021) se presenta una solución de monitoreo mediante el uso de la puerta de enlace IoT2040 de Siemens, el entorno de programación Node-RED y la plataforma Ubidots, que integran un sistema IoT. De dicha plataforma, surge esta propuesta, de utilizar como criterio principal el ser un software libre y gratuito; para monitoreo y control de sistemas eléctricos mediante IoT. En el trabajo (Aya *et al.*, 2018) se desarrolla una aplicación móvil en Appinventor utilizando una tarjeta WeMos

\*Autor para correspondencia: gonzalosieben@gmail.com

**Correo electrónico:** gonzalosieben@gmail.com (Gonzalo Hedain López-Mera), erick.padilla@upatlatomulco.edu.mx (Erick Axel Padilla-García), rd-cruz@comunidad.unam.mx (Raúl Dalí Cruz-Morales) y dativa19@hotmail.com (David Tinoco-Varela).

**Historial del manuscrito:** recibido el 26/10/2022, última versión-revisada recibida el 26/10/2022, aceptado el 07/11/2022, publicado el 11/11/2022. **DOI:** <https://doi.org/10.29057/icbi.v10iEspecial5.10123>



D1 mini que les permite medir, procesar, controlar y enviar la medición de variables a la plataforma Thingier.io, los autores consideraron esa plataforma por el servicio gratuito que ofrece para conectar hasta dos dispositivos, evidenciando esta característica que es conveniente para el desarrollo del proyecto. Asimismo, se destaca que esta plataforma ofrece una buena tasa de transferencia de datos por minuto, y el uso de tarjetas embebidas basadas en el microcontrolador ESP8266 que ofrece la minimización de costos en el hardware, generando un nivel de competitividad sobre otras marcas.

En el área de robótica la investigación realizada en (Fauso *et al.*, 2018) presenta un diseño de una interfaz para la adquisición de datos en robots móviles, describiendo una interfaz hombre-máquina desarrollada en Python para la visualización de los datos adquiridos de la tarjeta Sense HAT, misma que provee sensores de temperatura, humedad y presión, cuya tarjeta de adquisición está montada sobre el robot móvil Danchex v2.0. Para este manuscrito, la interfaz gráfica utiliza la librería Tkinter propia de Python, la cual se elige por ser de software abierto, permitiendo el uso de su propio diseño de GUI. Se destaca que el uso de software libre reduce los costos en comparación con programas comerciales como LabView.

En este documento se muestra un panorama de la tecnología IoT en la integración de un diseño propio de una GUI en Python, la plataforma Blynk y el uso de microcontroladores ESP. El diseño de la GUI se valida mediante el monitoreo y control de un sistema esclavo electromecánico, el cual es controlado por medio de acciones remotas en el sistema y acciones virtuales sobre la misma GUI. En la Sección II, se detallan los principales elementos de la tecnología IoT. La Sección III presenta los dispositivos utilizados para la comunicación con la plataforma Blynk. Con base en la propuesta del sistema de monitoreo y control, en la Sección IV se describe el sistema esclavo diseñado para probar diversas salidas y entradas usando la GUI propuesta. En la Sección V se muestra el diseño de la GUI en Python. En la Sección VI se tienen las pruebas realizadas para la validación del control y envío de señales utilizando la GUI diseñada. Finalmente, en la Sección VII se presentan las conclusiones.

## 2. Panorama del internet de las cosas

En este momento la cuarta revolución industrial no sólo precisa que los procesos industriales sean eficientes, sino que, además sean sistemas inteligentes, autónomos y puedan medir las diferentes variables de su entorno, con el fin de controlar al sistema y mejorar el proceso que esté realizando. Dejando a un lado los procesos de automatización basados en microcontroladores como por ejemplo el presentado en (del Brío y de Tecnología Electrónica, 1999), donde estos cumplían su propósito de manera aislada o en el mejor de los casos con la interacción de una interfaz visual a base de algún protocolo de comunicación, surge Banzi (Severance, 2014) en el año de 2005 con la propuesta del desarrollo de una tarjeta económica, de fácil acceso y con la propuesta de ser utilizada por personal con mínimo conocimiento en electrónica, lo que hoy se conoce como tarjeta Arduino. De tal manera que actualmente existen una gran variedad de tarjetas embebidas compatibles con Arduino y diferentes extensiones, como módulos (llamados shields) de WiFi

para la aplicación de sistemas IoT.

Se entiende que la tecnología IoT tiene como finalidad la interacción de entornos virtuales y elementos físicos, permitiendo así que cada elemento físico sea capaz de proporcionar información y esta sea almacenada en alguna plataforma virtual a través del internet. Con esta información se pueden determinar posibles anomalías en los procesos de fabricación, determinar estados o simplemente filtrar datos en tiempo real desde cualquier lugar (Bonilla-Fabela *et al.*, 2016). En los fundamentos se pueden mencionar tres elementos básicos del IoT que interactúan entre sí, según (Lee y Lee, 2015) estos son:

- Hardware**, como sensores y actuadores, siendo dispositivos que permiten el control de los sistemas;
- Plataforma**, el software que permite el intercambio de información entre las aplicaciones;
- Herramientas**, que permitan la visualización e interpretación de la información.

Como consecuencia principal según (Segura, 2016), el IoT podría permitir la conectividad de las personas en cualquier momento con cualquier cosa utilizando cualquier servicio o red.

## 3. Características del sistema electromecánico de prueba

En la implementación de IoT es necesario tener el hardware que permita la manipulación de actividades o procesos que permitan el cambio de estado de las variables a controlar como se menciona en la Sección II. Para la implementación de la GUI diseñada, se desarrolla un sistema electromecánico para el monitoreo y control, el cual se describe a continuación.

De las placas utilizadas, se seleccionaron las placas: ESP32 y ESP8266, con el fin de probar la diversidad de conexión entre los puntos de acceso al sistema. La placa ESP32 y ESP8266 tienen características similares, pero la primera cuenta con las cualidades de tener una unidad central de procesamiento de doble núcleo de 160 Mhz a 240 MHz mientras que el ESP8266 tiene un procesador de un solo núcleo que funciona a 80 MHz (Santos, 2021). La Figura 1 muestra el esquema característico del dispositivo ESP32.

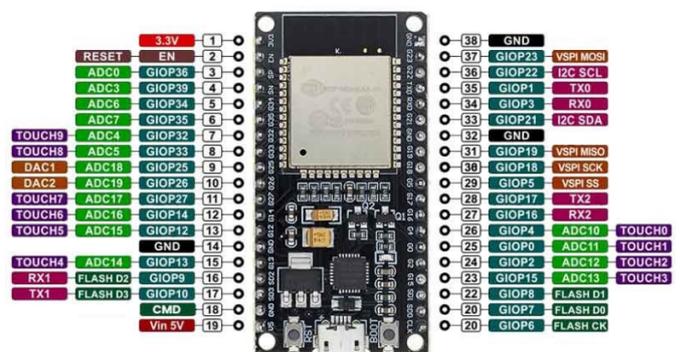


Figura 1: Disposición de los pines del ESP32 de 38 pines. Fuente: (Electronics, 2022)

Para el dispositivo definido como sistema maestro se cuenta con el ESP8266 NodeMCU, usado principalmente para enviar y recibir señales de control entre la GUI y el dispositivo a controlar. Para el sistema esclavo se utiliza el dispositivo ESP32 de

38 pines, siendo el ESP32 el dispositivo con mayor capacidad de procesamiento, es decir dedicado al control del dispositivo a monitorear o con interacción directa del hardware deseado a controlar.

Ambos sistemas maestro y esclavo están conectados a través de la plataforma Blynk. En la Figura 2 se muestra un esquema el procesamiento paralelo de los tres sistemas (maestro, esclavo, Blynk). En dicho esquema se muestra en forma general el flujo de programación en paralelo para la correcta funcionalidad y son la base de la programación de los microcontroladores ESP y la programación del diseño del GUI en Python. El sistema maestro es el encargado de realizar la conexión con la plataforma Blynk, realizar la lectura de los puertos virtuales, comunicarse con el sistema esclavo, realizar la habilitación del puerto serial para la comunicación con la GUI en la computadora y tener la recepción de datos de la GUI que permiten realizar las acciones de control para encender el led, accionar los botones.

En el sistema esclavo se tiene la conexión con la plataforma Blynk, se asigna la tarea al sistema esclavo, se declara la función que permite accionar el estado inicial del sistema esclavo, se hace la lectura del puerto virtual que acciona los motorreductores y respecto al dato recibido se realizan las acciones correspondientes, se declaran los pines de entrada para los interruptores táctiles y de acuerdo con su manipulación se envía el cambio de estado al sistema maestro. Para la interfaz se tiene la inicialización del puerto serial, la declaración de animación en las tres gráficas, el envío de datos para su correcta publicación y las señales de control que permiten el cambio de estado del sistema esclavo.

Ambas placas, ESP8266 y ESP32, cuentan con un módulo de comunicación inalámbrica Wifi y módulo bluetooth, lo que le permite desarrollar proyectos autónomos de bajo costo en diferentes aplicaciones (Ceja et al., 2017), siendo dispositivos de bajo costo en el mercado, con módulo Wifi. Además, se programan con código abierto compatible con el entorno de desarrollo integrado (IDE) de Arduino, el cual es gratuito y de programación conocida y amigable con el usuario, ya que se basa en el lenguaje C.

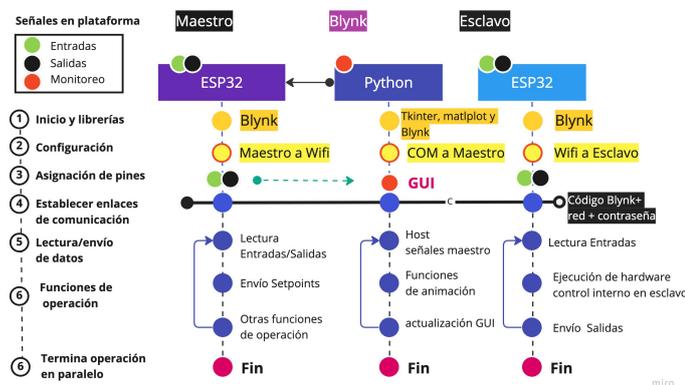


Figura 2: Flujo de programación en paralelo para el sistema propuesto.

#### 4. Sistema esclavo para entradas/salidas de la GUI.

En el sistema esclavo ESP32, se tiene como elemento de enlace a el servidor de la plataforma Blynk, y además es la par-

te que interactúa con el dispositivo a controlar. Por ello en esta propuesta, se desarrolla un sistema de referencia que permita controlar y monitorizar diferentes señales para aplicaciones futuras, como se ve en la Figura 3. Esta propuesta del sistema esclavo cuenta de dos motorreductores de plástico de reducción 1:48, el rango de voltaje de operación es de 3 V a 12 V en corriente directa. La velocidad sin carga de estos motorreductores es de 250 rpm. Como sensor de posición de cada motorreductor se acopla en un extremo de su eje un dispositivo encoder rotativo KY-404. Este tipo de encoder es un codificador incremental con dos canales que indican la dirección en la que se está girando, cuenta con 12 posiciones, cada una de 30 grados (Cameron, 2021). El funcionamiento del codificador se basa en el conocido código Grey de 2 bits. Estos motorreductores serán utilizados más adelante como tren motriz para robots móviles.

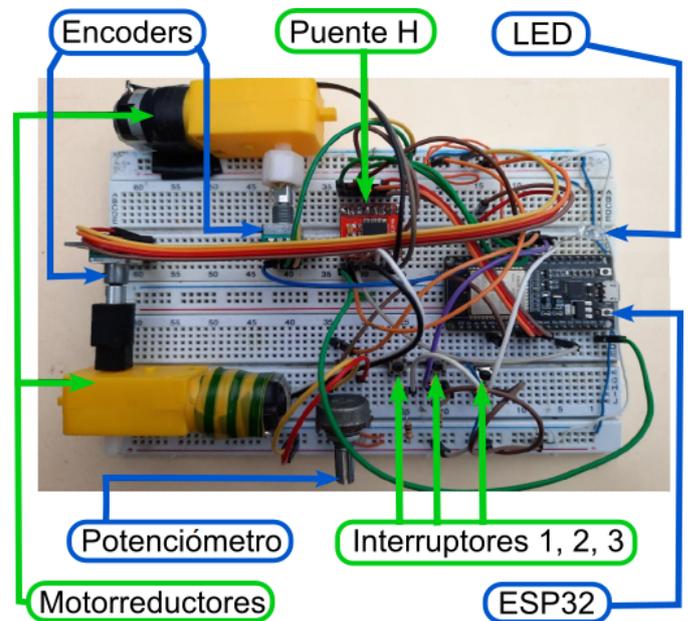


Figura 3: Elementos que componen al sistema esclavo.

Además de las tarjetas, se utiliza un puente H TB6612FNG el cual permite el control de dos motorreductores con una corriente constante de 1,2 A y corriente pico de 3,2 A. Este dispositivo cuenta con dos salidas de control que permiten el control de velocidad de los motorreductores mediante señal PWM de hasta 100 kHz y voltaje de salida de hasta 15 V (Llamas, 2016). El esquema de los pines de salida de este puente H se puede ver en la Figura 4.

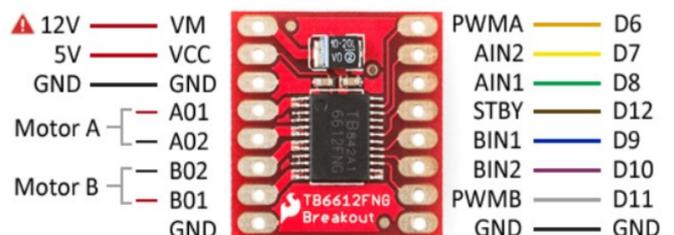


Figura 4: Disposición de los pines de salida en el puente H TB6612FNG. Fuente: (RoboComp.in., 2022)

Como instrumentos adicionales de monitoreo y control en el sistema esclavo, se colocan tres interruptores táctiles de 2 pines que al ser pulsados permiten el flujo de corriente y un led difuso de 5 mm de propósito general que emite luz al ser atravesado por una corriente nominal. La integración de los componentes del sistema esclavo propuesto se muestra en la Figura 3. La programación de las placas ESP8266 (Llamas, 2018) y ESP32 (Xukio, 2021) se realiza con el IDE de Arduino.

## 5. Interfaz gráfica de usuario en Python

En la elección del lenguaje de programación para la GUI diseñada se usa Python, por su creciente utilización para el desarrollo de software en los últimos años, además de tener la posibilidad de ser utilizado en diferentes plataformas y sistemas operativos, como Windows, Mac OS y Linux, sin olvidar que tiene licencia de código abierto. Por sus aplicaciones según (Fernandez, 2013), Python permite desarrollar aplicaciones científicas, comunicaciones de red para GUI, juegos para smartphone y aplicaciones web.

Para el diseño de la GUI se utiliza la librería Tkinter que es una librería nativa de Python. Tkinter cuenta con diferentes widgets de los cuales se utilizaron:

1. **Entry:** Es una caja de texto que permite, tanto mostrar información al usuario como obtenerla del mismo,
2. **Label:** Son etiquetas utilizadas para mostrar información de tipo String,
3. **Button:** Contiene texto o imágenes y se les puede asociar funciones,
4. **tk:** Creación de ventanas,
5. **Frame:** Creación de un espacio dentro de una ventana,
6. **Scale:** Proporciona un objeto de control deslizante gráfico que permite seleccionar valores de una escala específica.

La plataforma IoT seleccionada es Blynk que permite el control de sistemas desarrollados en Arduino, ESP y Raspberry Pi. Blynk solo requiere que el proyecto este enlazado a internet por medio de WiFi. De igual forma el atributo de elección se sustenta en que es una aplicación de código abierto, siendo que esta permite controlar múltiples dispositivos al mismo tiempo (Anónimo, 2018). Para la interfaz propuesta se tiene la inicialización del puerto serial, la declaración de animación en las tres gráficas, el envío de datos para su correcta publicación, las señales de control que permiten el cambio de estado de los botones 1, 2 y 3, las acciones para el botón Iniciar, Pausar y LEDOFF, las acciones para el botón Enviar rpm y del deslizador y las posibles acciones fuera del rango de velocidades. La Figura 5 muestra la GUI final donde las gráficas reciben las señales del sistema esclavo, los botones en color morado, el botón Enviar rpm, y el deslizador envían señales de control al sistema esclavo y los botones rojos son activados por el puerto serial en el sistema maestro.

## 6. Validación experimental

En esta sección se presenta la forma en que se realiza la validación experimental de la GUI. En el sistema esclavo se conecta

a una fuente para el funcionamiento del ESP32 (Figura 6), este sistema tiene un potenciómetro, encoders, los motorreductores e interruptores táctiles. El sistema esclavo se conecta vía WiFi a Blynk utilizando la interfaz de programación de aplicaciones (API) del firmware de la biblioteca de Blynk.

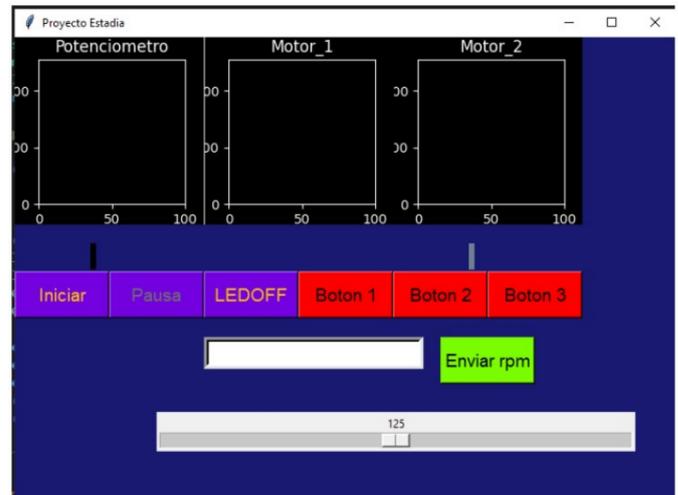


Figura 5: Diseño final de la GUI mostrando su estado inicial.

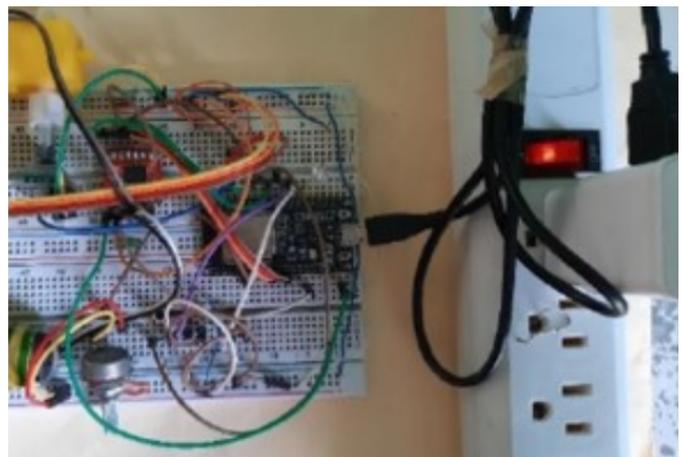


Figura 6: Conexión del sistema esclavo.



Figura 7: Conexión del sistema maestro.

El sistema maestro se conecta a la PC donde se encuentra la GUI de Python (Figura 7), éste sistema sirve como puerto de comunicación con el GUI diseñado en Python a través del puerto serial y este último, ejecutado desde una computadora para la comunicación entre el usuario y el sistema esclavo. Ver Figura 8.

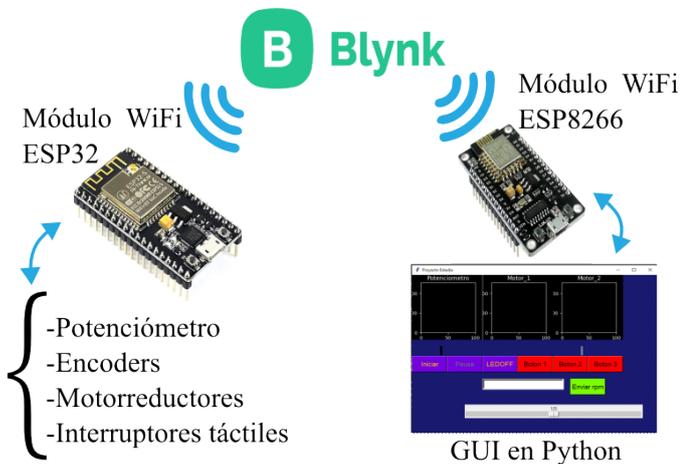


Figura 8: Esquema IoT de control y monitoreo con Blynk.

Para dar comienzo con el funcionamiento de la GUI se inicializa la configuración de red y se pone la contraseña del dispositivo. En el extremo de comunicación, el sistema esclavo se conecta a la fuente de energía. Ambos sistemas empiezan con el estado inicial, inmediatamente después, la GUI interactúa con el sistema esclavo, accionando ambos motorreductores a una velocidad de giro de 125 rpm. Esto fue verificado de manera visual, con una marca colocada en los ejes de los motorreductores. En el estado inicial, la GUI no muestra animación alguna en las gráficas. Para la validación de los diferentes botones, se describe a continuación la puesta en marcha de estos y las actualizaciones de los estados de los elementos del GUI. Al hacer clic en el botón Iniciar como se ve en la Figura 9, el mismo botón cambia de texto a Detener en la Figura 10, y se visualizan las animaciones de las gráficas del Potenciómetro, Motor\_1 y Motor\_2.

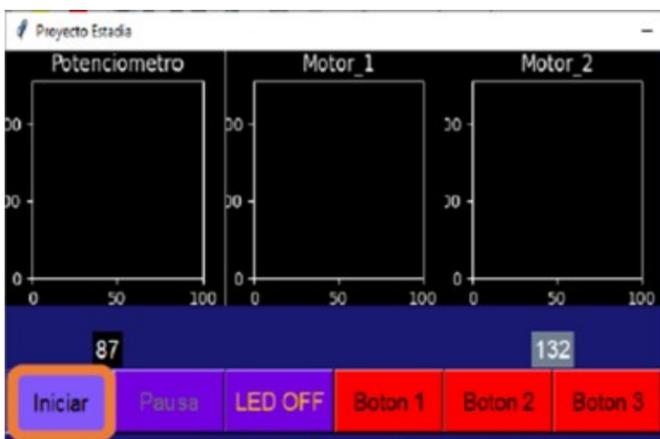


Figura 9: Validación del botón Iniciar.

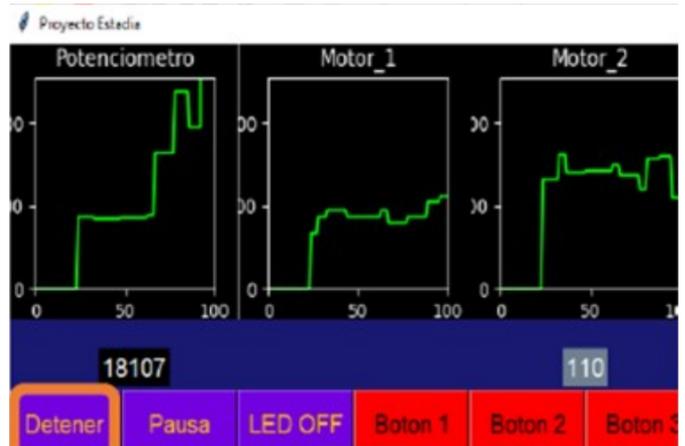


Figura 10: Cambio de estado del botón Iniciar a Detener, se inicializa la animación.

Al accionar el botón Pausa en la Figura 11, este permite visualizar en la Figura 12 la detención de las animaciones en las gráficas y mostrar el histórico de los valores animados, el texto cambia a Continuar. El funcionamiento de este botón valida que la GUI permite la adquisición de una magnitud de tipo analógica por el potenciómetro y la adquisición de un valor discreto por los encoders, con lo cual se tiene la animación para la visualización del usuario y la grabación digital de datos en memoria.

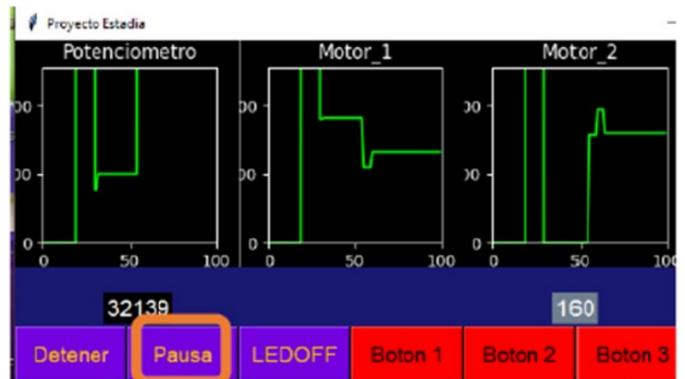


Figura 11: Validación del botón Pausa.

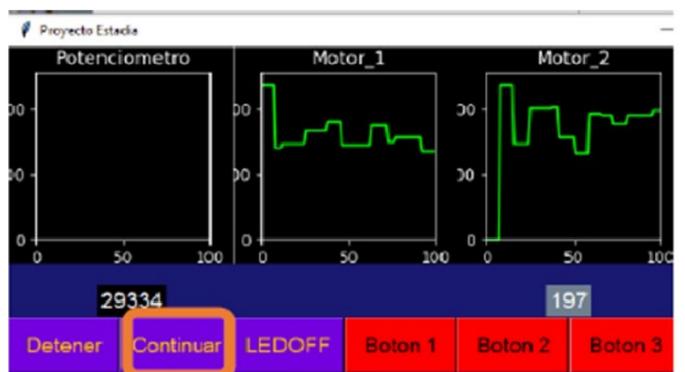


Figura 12: Cambio de estado del botón Pausa a Continuar, se detiene la animación.

El tercer botón de la GUI permite el control del dispositivo led en el sistema esclavo. El led está encendido en el estado inicial como se ve muestra en la Figura 13. Al accionar el botón LED OFF, el led se apaga como se ve en la Figura 14 y el texto cambia a LED ON. Con este botón se valida el control del sistema esclavo desde la GUI.

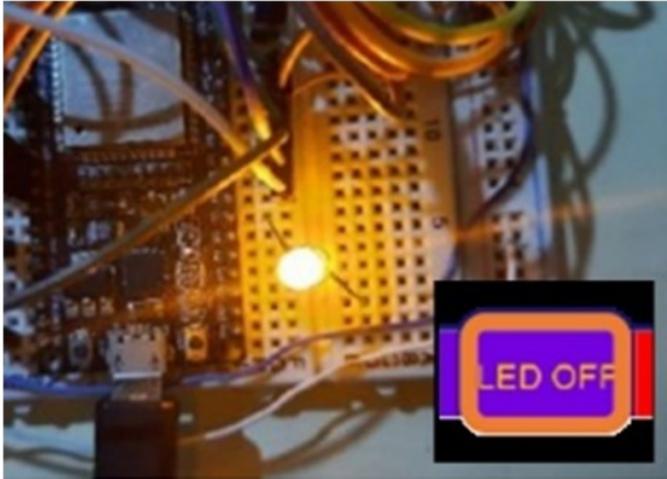


Figura 13: Validación del botón LEDOFF.

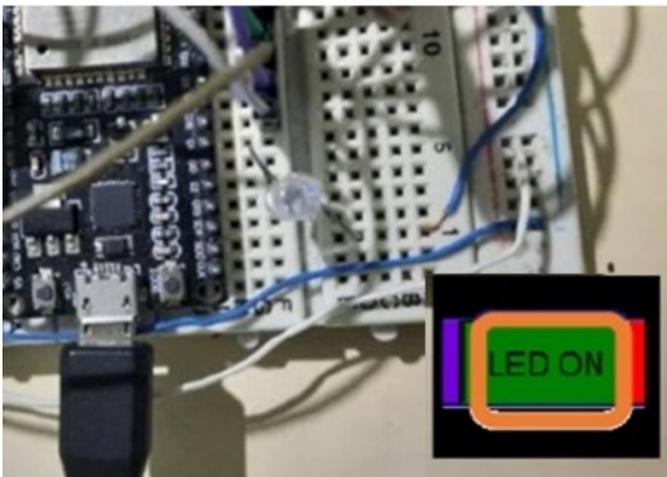


Figura 14: Cambio de estado del botón LEDOFF a LEDON, se apaga el led del sistema esclavo.

En la Figura 15 se muestran los tres interruptores táctiles en el sistema esclavo, mismos que son los accionamientos de interrupción que pueden ser ejecutados en cualquier momento. Para la validación del monitoreo desde la GUI, se presiona el interruptor número dos, visualizando en la Figura 16 el cambio de estado del indicador nombrado como Botón 2, cambiando de estado de color a rojo a verde. Mismo comportamiento se tiene con Botón 1 y Botón 3. Para el cambio de estado de color verde a rojo, solo es necesario presionar nuevamente el interruptor táctil y volverá a su estado anterior.

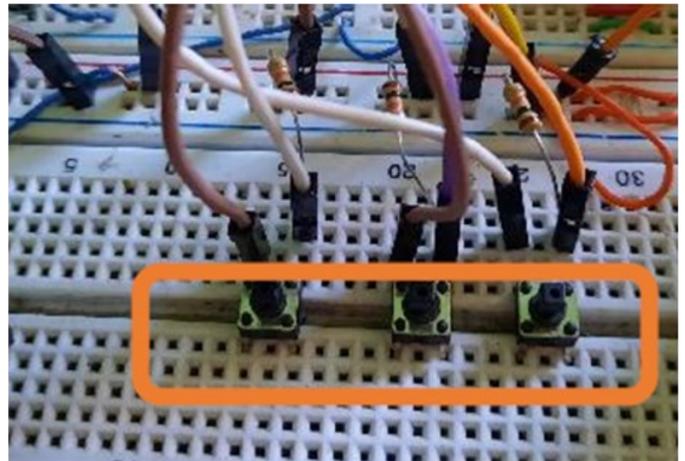


Figura 15: Validación de interruptores en el sistema esclavo.



Figura 16: Cambio de estado de interruptores, cambio a color verde del Botón 2 al ser pulsado el interruptor 2.

Para la validación del control de los motorreductores en el sistema esclavo con el GUI se hace mediante el deslizador y la introducción de valores por medio de teclado en la barra de Enviar rpm. Utilizando el deslizador, los valores entre 125 y 250 permiten el aumento en la velocidad del motor 1 de forma proporcional y de manera inversa en el motor 2, esto es, con un valor de 250 en el deslizador el motor 1 alcanza la velocidad máxima y el motor 2 tiene un paro total, las animaciones en el GUI se muestran en la Figura 17.



Figura 17: Validación del deslizador con valor de 250.

De forma similar ocurre para valores entre 0 y 125, en este caso el incremento de forma proporcional en velocidad es para el motor 2 y la disminución de velocidad es para el motor 1, la animación en la GUI se muestra en la Figura 18.



Figura 18: Validación del deslizador con valor de 1.

Cabe mencionar que los valores animados tienen un umbral de oscilación entre las transiciones de los valores en las gráficas de Motor\_1 y Motor\_2, pero en valores máximos y mínimos estos tienen estabilidad en animación después de cierto tiempo. Se acredita esta imprecisión de monitoreo en la forma en que se realiza la lectura de la señal en los encoders del sistema esclavo, la cual puede tener mejoras. Para finalizar la validación de la GUI, se introduce de manera directa el valor numérico para el control de los motorreductores y se da clic en Enviar rpm. Los resultados obtenidos son semejantes que con el deslizador cuando los valores están en el rango de 0 a 250, en el caso de introducir valores mayores como el mostrado en la Figura 19, la función establecida a los motorreductores es el paro inmediato de ambos.



Figura 19: Control directo mediante el botón Enviar rpm.

### 6.1. Experimento: Activación de dispositivos externos

En este experimento, se prueba el accionamiento de un robot móvil diferencial, cuyo chasis se muestra en la Figura 20.

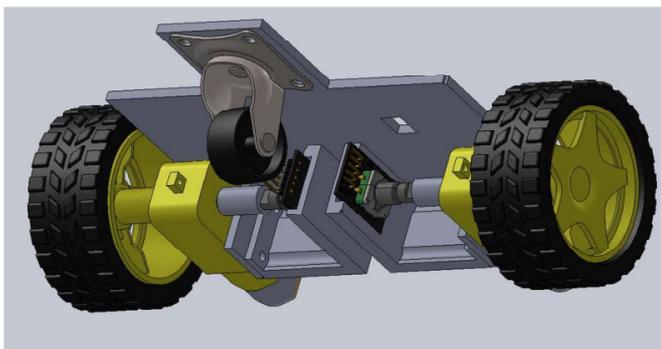


Figura 20: Estructura de robot móvil diferencial para experimento.

El robot consta de los mismos elementos que la plataforma esclava mostrada al inicio de la sección 6. Se consideran 2 encoders rotativos, 2 motorreductores de plástico y el puente H TB6612FNG. Sin embargo, el robot ya cuenta con un programa de control interno PID el cual ya está validado y programado de manera independiente con un tiempo de muestreo de 10 ms. El comportamiento en simulación y experimental del robot en una trayectoria se muestra en las gráficas de la Figura 21. Así, el robot se considera como un dispositivo independiente externo, que puede activarse o desactivarse a distancia.

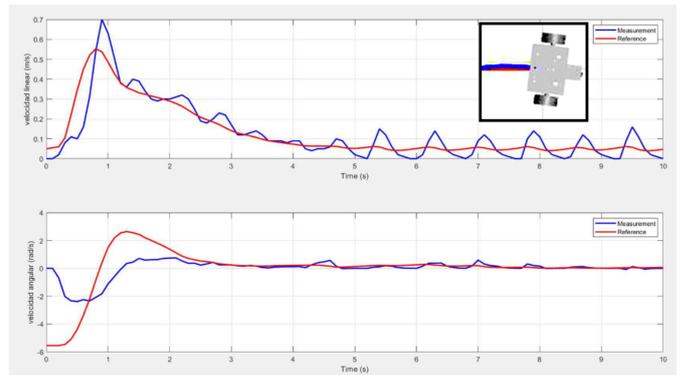


Figura 21: Desempeño del controlador interno del robot.

En este experimento, el robot ahora cuenta con interacción con el sistema esclavo para activar (indicador LED ON) o desactivar (indicador LED OFF) el controlador del robot. Primero, el robot se mantiene a la espera de la señal de arranque del sistema esclavo mediante una señal digital virtual llamada V0, dicha señal pertenece al sistema esclavo virtual registrado en la plataforma de Blynk, la cual puede cambiar de estado mediante la señal GUI o mediante los widgets de la plataforma web. Entonces, el robot vía remota espera la señal de arranque, como se muestra en la Figura 22.

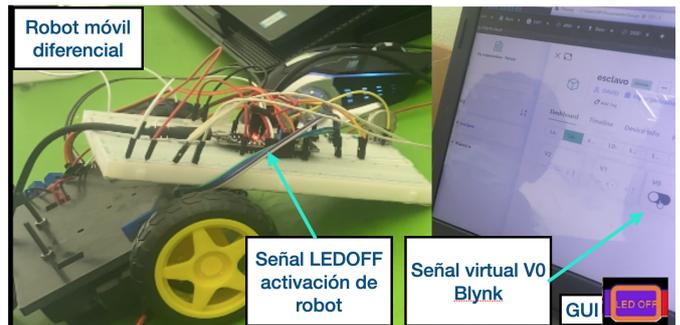


Figura 22: Control de evento discreto de robot en OFF.

Segundo, el robot inicia el PID interno, y realiza la trayectoria programada. Se activa la señal del sistema esclavo, y se muestra esa activación de señal virtual V0 tanto en la plataforma de Blynk como en la GUI, como se muestra en la Figura 23. Entonces, el robot comienza la tarea pre-programada interna, manteniendo la señal V0 en alto y mantiene LED-ON en la GUI.

Una vez finalizada la tarea, se desactiva la señal virtual V0, y envía LED-OFF en la GUI, esperando a que vuelva a pedir activación de nuevo o que reciba otro tipo de instrucción.

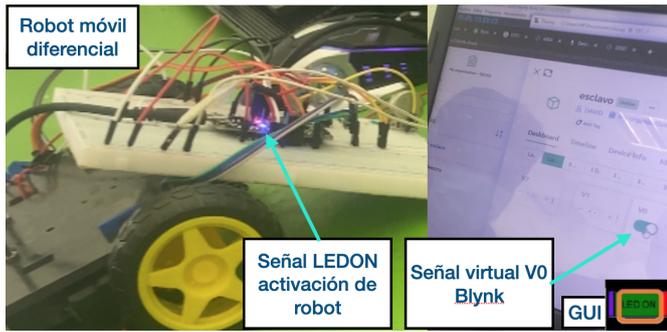


Figura 23: Control de evento discreto de robot en ON.

A pesar de que llegan todas la señales a la animación de la GUI, existen aún diversos retardos que impiden de manera eficiente un sistema de control donde el flujo de señales en la nube pueda ser parte de la realimentación del sistema. Sin embargo, sería posible diseñar un control supervisor de eventos discretos donde los retardos puedan manejarse dentro del inicio y fin de cada dispositivo externo, como en este experimento propuesto.

Como parte del desarrollo futuro, se propone analizar en detalle los diversos retardos ocasionados por el sistema, el control, la conexión de internet, la velocidad de la red, entre otros, para tener un acercamiento directo al control dentro del proceso.

## 7. Conclusiones

Por medio de la GUI diseñada en Python se logra el control y monitoreo de un sistema esclavo a través del módulo wifi del microcontrolador ESP32, con la cual se permite ejecutar funciones y recibir los valores de velocidad de los motorreductores, además de la recepción del valor de la tensión en el potenciómetro, activación de señales digitales, leds y animar estas señales en las gráficas de la interfaz. Se entiende que en estas animaciones hay un retraso de tiempo en la actualización de los valores obtenidos, debido al tráfico de datos entre envío del control en los motorreductores, la recepción de datos y la velocidad de ambas redes (esclavo/maestro), observando un retraso máximo de 1.5 segundos durante la validación de esta propuesta.

Se logró el monitoreo de los estados de conmutación de tres interruptores, permitiendo visualizar el cambio de estado en la interfaz de manera gráfica. Se logró la recepción de datos cambiando el dispositivo esclavo de un ESP8266 a un ESP32, y a su vez se realizó la conexión exitosa entre un dispositivo ESP8266

utilizado como dispositivo maestro y el dispositivo ESP32 utilizado como dispositivo esclavo. Se utilizó exitosamente la plataforma Blynk, permitiendo desarrollar el entorno de IoT en el proyecto propuesto, con la ventaja del uso de código de seguridad para evitar el jaqueo de usuarios externos a dicho sistema de control y monitoreo.

## Agradecimientos

Los autores agradecen a la Universidad Nacional Autónoma de México por su parcial financiamiento para lograr este proyecto, mediante PICI 2254 y PAPIME PE105822. También, agradecemos a Enrique Reyes Morales, egresado de Ingeniería en Robótica de la Universidad Politécnica de Atlacomulco, por su valioso apoyo para realizar las pruebas experimentales durante su proyecto de estancia.

## Referencias

- Anónimo (2018). Blynk, plataforma de internet de las cosas en la red.
- Aya, A. R., Luna, J. F., y García, J. C. (2018). Sistema de control y telemetría de datos mediante una aplicación móvil en android basado en iot para el monitoreo de datos. *Espacios*, 39:22–30.
- Bonilla-Fabela, I., Tavizon-Salazar, A., Morales-Escobar, M., Guajardo-Muñoz, L. T., y Laines-Alamina, C. I. (2016). Iot, el internet de las cosas y la innovación de sus aplicaciones. *Vinculatégica efan*, 2(1):2313–2340.
- Cameron, N. (2021). Rotary encoder control.
- Ceja, J., Rentería, R., Ruelas, R., y Ochoa, G. (2017). Módulo esp8266 y sus aplicaciones en el internet de las cosas. *Revista de Ingeniería eléctrica*, 1(2):24–36.
- del Brío, B. M. y de Tecnología Electrónica, A. (1999). *Sistemas electrónicos basados en microprocesadores y microcontroladores*. Prensas Universitarias de Zaragoza.
- Electronics, U. (2022). Esp32 38 pines esp wroom 32.
- Fausto, M. S. R., Jonathan, V. T., Martínez, V., y Salvador, J. (2018). Diseño de una interfaz hombre-máquina para la adquisición de datos en robots móviles.
- Fernandez, A. (2013). *Python 3 al descubierto*. Alfaomega Grupo Editor.
- Lee, I. y Lee, K. (2015). The internet of things (iot): Applications, investments, and challenges for enterprises. *Business horizons*, 58(4):431–440.
- Llamas, L. (2016). Controlar dos motores dc con arduino y driver tb6612fng.
- Llamas, L. (2018). Cómo programar esp8266 con el ide de arduino.
- Muñoz, C. A. (2021). Diseño y simulación de un sistema de monitoreo para un proceso de control de temperatura mediante plc e iot para la empresa educativa.
- RoboComp.in. (2022). Tb6612fng dual motor driver module 1a for arduino.
- Santos, S. (2021). Esp32 vs esp8266 – pros and cons.
- Segura, A. A. (2016). Arquitectura de software de referencia para objetos inteligentes en internet de las cosas. *Revista latinoamericana de ingeniería de software*, 4(2):73–110.
- Severance, C. (2014). Massimo banzi: Building arduino. *Computer*, 47(1):11–12.
- Xukio (2021). Programación de un esp32 nodemcu con el ide de arduino.