Análisis de Redes WiFi con Raspberry Pi Pico 2W

Lina Fernanda Guio Cabrera
est.lina.guio@unimilitar.edu.co

***Resumen*—En la práctica se estudió el funcionamiento del módulo WiFi de la Raspberry Pi Pico W, realizando pruebas de identificación de la dirección MAC, escaneo de redes y configuración del dispositivo como punto de acceso. Se analizaron los valores de RSSI, la distribución de canales y la estabilidad de la señal. Finalmente, se implementó el control remoto del LED desde una página web local, comprendiendo los principios básicos de comunicación inalámbrica mediante MicroPython.**

I. INTRODUCCIÓN

La tecnología WiFi representa una de las principales herramientas de comunicación inalámbrica utilizadas en la actualidad, ya que permite la transmisión de datos entre dispositivos dentro de una red local sin necesidad de cables. Su funcionamiento se basa en el uso de ondas de radio, particularmente en las bandas de 2.4 GHz y 5 GHz, siendo la primera la más común por su amplio alcance y compatibilidad. En este contexto, la Raspberry Pi Pico W —o su versión más reciente, la Pico 2W— incorpora un módulo WiFi que posibilita la conexión con redes inalámbricas y la creación de puntos de acceso, lo que la convierte en un dispositivo versátil para el desarrollo de proyectos educativos y aplicaciones del Internet de las Cosas (IoT).

El presente trabajo tuvo como finalidad comprender los fundamentos prácticos del funcionamiento del WiFi en la Raspberry Pi Pico W mediante una serie de ejercicios experimentales. Se abordaron temas esenciales como la identificación de la dirección MAC, la medición del RSSI (Received Signal Strength Indicator), el análisis de canales de frecuencia, y la configuración del microcontrolador como punto de acceso inalámbrico (AP). Además, se exploró la relación entre la potencia de la señal y la distancia, así como la forma en que los canales se distribuyen en la banda de 2.4 GHz para evitar interferencias.

Por medio del lenguaje MicroPython y los scripts proporcionados en el repositorio oficial, se realizaron pruebas de conexión, escaneo y control remoto, fortaleciendo la comprensión de los conceptos teóricos de redes WiFi aplicados a un entorno embebido. La práctica también permitió evidenciar la importancia de la estabilidad de la señal, la influencia del entorno físico en la intensidad del RSSI y la utilidad de los protocolos de comunicación como HTTP y UDP en la transmisión de datos. En conjunto, la experiencia buscó no solo el desarrollo técnico, sino también la formación de competencias analíticas en el manejo de tecnologías inalámbricas.

II. DESARROLLO

A. Marco teórico

RASPBERRY PI PICO 2 W

Es la actualización de la placa inalámbrica de la Fundación Raspberry Pi, ahora basada en el microcontrolador RP2350, que mejora significativamente al RP2040 al ofrecer mayor rendimiento, más memoria y nuevas funciones de seguridad, manteniendo la compatibilidad con los modelos anteriores de la serie Pico.

Este nuevo chip incorpora doble núcleo y doble arquitectura, permitiendo trabajar con Arm Cortex-M33, que incluye FPU y duplica la velocidad del Cortex-M0+, o con RISC-V Hazard3, ideal para proyectos en arquitecturas abiertas. Además, la memoria se duplicó: 512 KB de SRAM y 4 MB de flash, junto con más periféricos como tres bloques PIO y el periférico HSTX para transmisión de alta velocidad.

En el ámbito de la seguridad, el RP2350 integra Arm TrustZone, arranque seguro, aceleración SHA-256, almacenamiento seguro de claves y un generador de números aleatorios, lo que lo hace confiable para aplicaciones profesionales.

Finalmente, la Pico 2W sigue siendo programable en C/C++, MicroPython y CircuitPython, manteniendo su carácter accesible y versátil para educación y desarrollo profesional. No obstante, el chip RP2350 A2 presenta la errata E9, que afecta algunos GPIO, por lo que en ciertos casos se requieren resistencias externas para garantizar un funcionamiento correcto. [1]

Tecnología WiFi y su Aplicación en la Raspberry Pi Pico 2W

La tecnología WiFi constituye uno de los medios más utilizados para la transmisión inalámbrica de datos en redes locales (WLAN). Su funcionamiento se basa en el uso de ondas de radio que operan, principalmente, en las bandas de 2.4 GHz y 5 GHz. La Raspberry Pi Pico 2W, al incorporar conectividad inalámbrica, utiliza la banda de 2.4 GHz, la cual se encuentra dividida en 13 canales de frecuencia en la mayoría de las regiones del mundo. Cada canal tiene un ancho de banda de aproximadamente 5 MHz y se encuentra centrado en una

frecuencia específica, como se muestra en la tabla de canales y frecuencias. Esta segmentación permite la coexistencia de múltiples redes WiFi en un mismo entorno, aunque el solapamiento de canales puede afectar la calidad de la comunicación.

Dentro de una red WiFi, existen diversos conceptos fundamentales. El BSSID (Basic Service Set Identifier) es un identificador único que distingue a cada punto de acceso (Access Point, AP) dentro de una red. Usualmente, este identificador corresponde a la dirección MAC (Media Access Control) del dispositivo transmisor, lo que permite diferenciar redes que pueden compartir el mismo nombre o SSID (Service Set Identifier). Por otra parte, el RSSI (Received Signal Strength Indicator) representa la intensidad de la señal recibida desde un punto de acceso. Se mide en decibelios-milivatios (dBm) y proporciona una referencia de la calidad del enlace: valores más cercanos a cero indican mejor recepción. Por ejemplo, un RSSI de -40 dBm implica una señal fuerte y estable, mientras que valores inferiores a -80 dBm reflejan una conexión débil o inestable.

En el entorno de programación MicroPython, la Raspberry Pi Pico 2W ofrece un conjunto de funciones que permiten gestionar la conectividad WiFi de manera sencilla. El módulo network se encarga del control de las interfaces inalámbricas. Para operar en modo cliente (estación), se utiliza la instrucción wlan = network.WLAN(network.STA_IF), con la cual el dispositivo puede conectarse a redes disponibles. Posteriormente, la función wlan.active(True) activa la interfaz, mientras que el comando wlan.scan() realiza un escaneo de los puntos de acceso cercanos, devolviendo una lista con información sobre SSID, BSSID, canal y nivel de señal. La conexión a una red específica se establece mediante wlan.connect('SSID', 'clave'), donde se indican el nombre de la red y la contraseña correspondiente.

De manera complementaria, la Pico 2W también puede funcionar como punto de acceso (AP), permitiendo que otros dispositivos se conecten a ella. Para ello, se utiliza la instrucción ap = network.WLAN(network.AP_IF) y se activa con ap.active(True). A través del método ap.config(), es posible definir parámetros como el nombre del AP, el canal de transmisión y la clave de acceso. Finalmente, el comando wlan.isconnected() permite verificar si la interfaz WiFi en modo estación ha establecido una conexión exitosa con una red.

B. Procedimiento}

La primera parte del laboratorio consistió en identificar la dirección MAC del módulo WiFi de la Raspberry Pi Pico W mediante la ejecución del programa **mac_wifi.py**, disponible en el repositorio https://github.com/jrugeles/wifipico. Para ello, primero se descargó el archivo desde el repositorio y se abrió en el entorno de desarrollo **Thonny**. Luego, se cargó el script en la raspberry Pico 2W y se ejecutó, observando en la consola la salida del programa, la cual mostró la dirección MAC **2C:CF:67:DD:9D:8F**

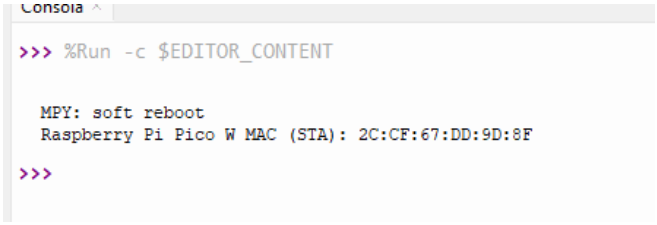


Ilustración 1

Posteriormente, se reinició el microcontrolador varias veces y se volvió a ejecutar el programa en cada ocasión, con el fin de comprobar si la dirección MAC cambiaba o permanecía constante entre los reinicios. Los resultados mostraron que la dirección MAC **no variaba**, permaneciendo igual en todas las ejecuciones. Esto confirmó que la dirección MAC es un identificador único e invariable**,** asignado por el fabricante al módulo WiFi del dispositivo.

El hecho de que la dirección MAC se mantuviera constante indicó que este valor está grabado en el hardware y sirve para identificar de manera única al dispositivo dentro de una red inalámbrica. En condiciones normales, este resultado es el esperado, ya que la MAC permite que routers y puntos de acceso reconozcan a cada equipo individualmente. En caso de que la dirección hubiera cambiado entre reinicios, podría haberse debido a un error del firmware o a la asignación temporal de direcciones MAC aleatorias por motivos de privacidad.

La segunda parte del laboratorio consistió en realizar un escaneo de redes WiFi cercanas utilizando el programa scanner_wifi.py, se ejecutó el script desde el entorno de desarrollo Thonny. Una vez iniciado, el programa comenzó a detectar los puntos de acceso (Access Points – AP) disponibles en la banda de 2.4 GHz, mostrando los resultados en la consola por iteraciones y ordenando las redes de acuerdo con su nivel de señal (RSSI), de mayor a menor.

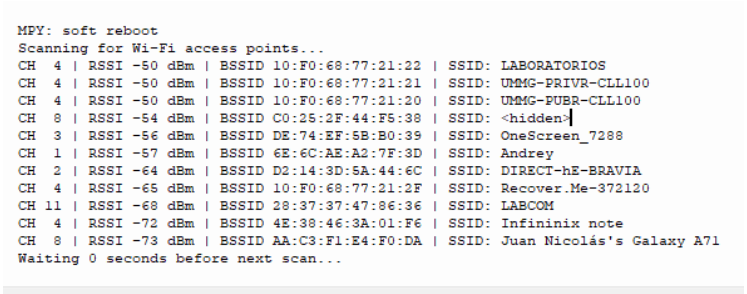


Ilustración 2. Tiempo en 0 segundos

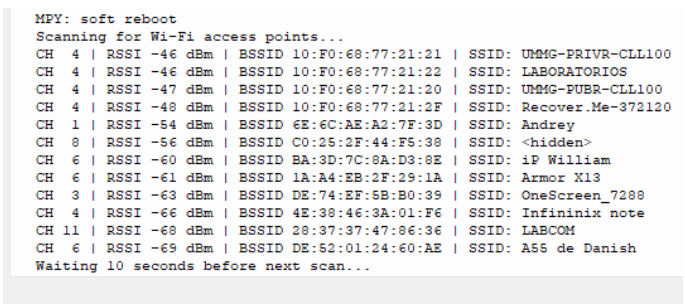


Ilustración 3. Tiempo en 10 segundos

```
Consola
Scanning for Wi-Fi access points...
CH  4 | RSSI -47 dBm | BSSID 10:F0:68:77:21:2F | SSID: Recover.Me-372120
CH  4 | RSSI -50 dBm | BSSID 10:F0:68:77:21:22 | SSID: LABORATORIOS
CH  4 | RSSI -50 dBm | BSSID 10:F0:68:77:21:21 | SSID: UMMG-PRIVR-CLL100
CH  4 | RSSI -53 dBm | BSSID 10:F0:68:77:21:20 | SSID: UMMG-PUBR-CLL100
CH  6 | RSSI -55 dBm | BSSID DE:52:01:24:60:AE | SSID: A55 de Danish
CH  8 | RSSI -55 dBm | BSSID C0:25:2F:44:F5:38 | SSID: <hidden>
CH  4 | RSSI -57 dBm | BSSID 4E:38:46:3A:01:F6 | SSID: Infininix note
CH 11 | RSSI -59 dBm | BSSID 10:F0:68:77:20:C0 | SSID: UMMG-PUBR-CLL100
CH 11 | RSSI -60 dBm | BSSID 10:F0:68:77:20:C2 | SSID: LABORATORIOS
CH 11 | RSSI -60 dBm | BSSID 10:F0:68:77:20:C1 | SSID: UMMG-PRIVR-CLL100
CH  2 | RSSI -61 dBm | BSSID D2:14:3D:5A:44:6C | SSID: DIRECT-hE-BRAVIA
CH  3 | RSSI -63 dBm | BSSID DE:74:EF:5B:B0:39 | SSID: OneScreen_7288
CH  1 | RSSI -63 dBm | BSSID 6E:6C:AE:A2:7F:3D | SSID: Andrey
CH 11 | RSSI -64 dBm | BSSID 28:37:37:47:86:36 | SSID: LABCOM
CH  6 | RSSI -66 dBm | BSSID BA:3D:7C:8A:D3:8E | SSID: iP William
CH  6 | RSSI -70 dBm | BSSID 1A:A4:EB:2F:29:1A | SSID: Armor X13
CH 11 | RSSI -82 dBm | BSSID 10:F0:68:77:1D:11 | SSID: UMMG-PRIVR-CLL100
CH  6 | RSSI -89 dBm | BSSID 2C:CF:67:DD:9C:33 | SSID: Grupo dinamita
Waiting 20 seconds before next scan...
```

Ilustración 4. Tiempo en 20 segundos

Se registraron un total de 11 redes WiFi disponibles en el entorno. Al analizar los resultados, se observó que 4 de estas redes se encontraban utilizando el mismo canal, específicamente el canal 4. Este hecho evidenció la existencia de un solapamiento de canales dentro de la banda de 2.4 GHz, ya que el canal 4 se traslapa con los canales adyacentes (1, 2, 3, 5, 6), generando posibles interferencias entre las redes que operan en rangos de frecuencia cercanos.

Este solapamiento puede ocasionar una disminución en la calidad de la conexión, pérdida de paquetes, menor velocidad de transmisión y un aumento en la latencia. En consecuencia, se concluyó que una buena práctica consiste en configurar los puntos de acceso en canales no solapados, como los canales 1, 6 y 11, con el fin de minimizar las interferencias y optimizar el rendimiento de la red inalámbrica.

Asimismo, se notó que el valor del RSSI fluctuaba entre iteraciones, lo cual puede deberse a varios factores, como la distancia entre la raspberry pi Pico 2W y los puntos de acceso, la presencia de obstáculos físicos (paredes, objetos metálicos), la interferencia de otros dispositivos electrónicos o incluso las variaciones naturales del entorno electromagnético.

Luego, en la tercera parte del laboratorio se configuró la Raspberry Pi Pico 2W como un punto de acceso inalámbrico (AP), con el objetivo de permitir la conexión de un dispositivo cliente, como un computador o teléfono móvil, y controlar el estado del LED integrado a través de una página web local.

Para ello, primero se cargaron en la memoria del Pico 2W los archivos APWifipico.py e index.html. Se tuvo en cuenta que al renombrar el archivo principal con el nombre main.py, este se ejecutaría automáticamente cada vez que se alimentara el circuito, lo cual facilitó el arranque del sistema. Luego, se editó el bloque de configuración dentro del código, asignando un nombre personalizado al punto de acceso junto con la contraseña “12345678” y el canal 6 como frecuencia de operación.

```
SSID     = "PicoW-AP"
PASSWORD = "12345678"
CHANNEL  = 6
```

Ilustración 5

Después de ejecutar el programa, se verificó en la consola el mensaje que indicaba la dirección IP asignada y el canal activo. Posteriormente, desde un dispositivo móvil o un computador, se realizó la conexión al SSID creado.

Ilustración 6

Se accedió a la dirección http://192.168.4.1 mediante el navegador. Una vez abierta la página, se probaron los botones LED ON y LED OFF, observando cómo el LED en la Raspberry respondía a las órdenes en tiempo real.

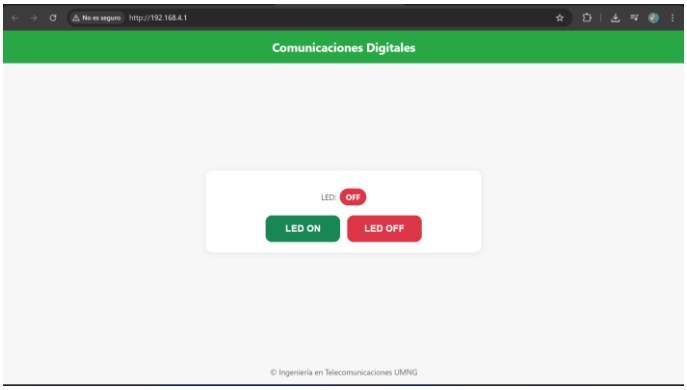


Ilustración 7

Finalmente, se accedió a la dirección http://192.168.4.1/state donde el servidor integrado devolvió un archivo en formato JSON con el estado actual del LED.

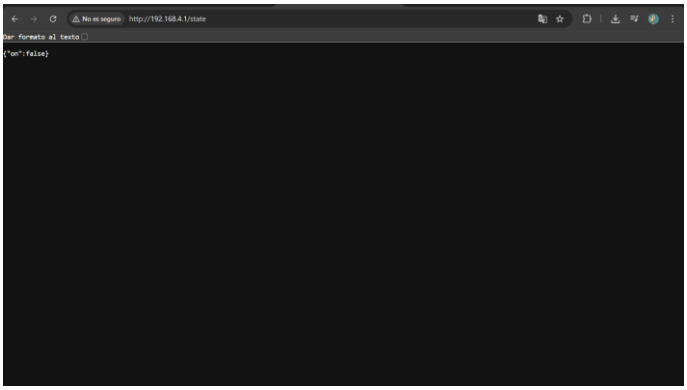


Ilustración 8

En esta parte del laboratorio se detuvo la ejecución del programa y se modificó el canal de operación en el archivo APWifipico.py, cambiándolo al canal 8. Posteriormente, se

verificó el cambio utilizando el escáner scanner_wifi.py desde otro dispositivo, con el fin de confirmar que el punto de acceso efectivamente estaba transmitiendo en el nuevo canal configurado.

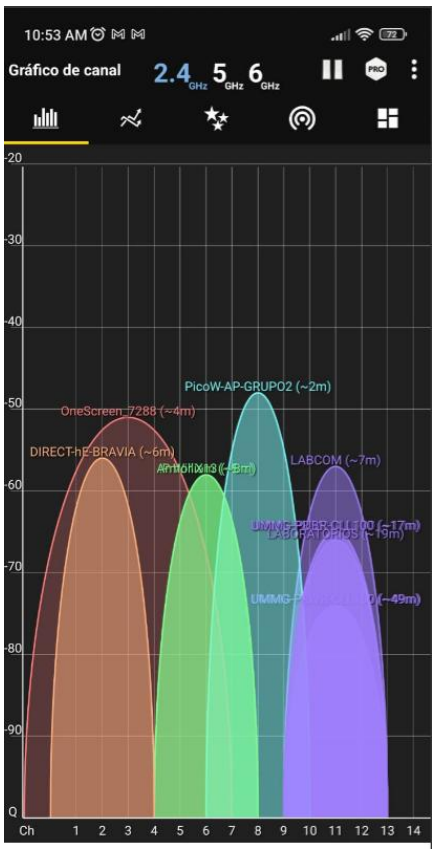


Ilustración 9

Durante la verificación se observó que, al usar el canal 8, sí se produce solapamiento con otros canales cercanos, como el 6 y el 11. Esto se debe a que en la banda de 2.4 GHz los canales están separados solo por 5 MHz, mientras que cada canal ocupa un ancho de aproximadamente 22 MHz. Como resultado, las frecuencias de los canales adyacentes se superponen parcialmente, generando interferencia y disminuyendo la calidad de la señal.

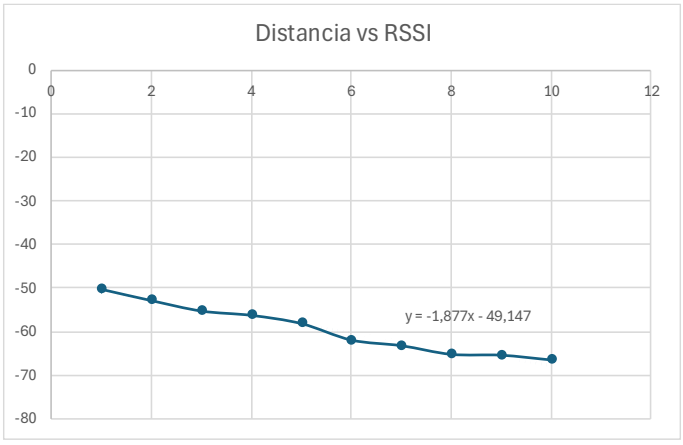
Por último, se realizó la parte final del laboratorio, que consistió en integrar y analizar distintos aspectos del funcionamiento del WiFi en la Raspberry Pi Pico W. En esta etapa se llevaron a cabo tres actividades principales.

Primero, se modificaron los programas APWifipico.py e index.html con el fin de permitir la lectura del valor del conversor analógico-digital ADC0 directamente desde la interfaz web del punto de acceso creado por la Pico W. Esta modificación permitió visualizar en tiempo real el valor analógico medido, expresado tanto en unidades digitales como en voltaje, además de mantener el control del encendido y apagado del LED desde el navegador.

Ilustración 10

Posteriormente, se diseñó y ejecutó un experimento para analizar la variación del nivel de señal recibida (RSSI) con respecto a la distancia. Se realizaron mediciones cada metro utilizando un punto de acceso externo (como un teléfono celular), registrando al menos diez lecturas por posición para calcular un promedio representativo. Los resultados se almacenaron en un archivo CSV dentro del microcontrolador y posteriormente se graficaron, obteniendo así una relación entre el RSSI y la distancia. A partir de estas mediciones se determinó el alcance máximo efectivo de la señal y se ajustó una ecuación que describía el comportamiento de la atenuación en función de la distancia.

Distancia(m)	RSSI (dBm)
1	-50,2
2	-52,8
3	-55,2
4	-56,2
5	-58,1
6	-61,9
7	-63,2
8	-65,2
9	-65,4
10	-66,5



Los datos obtenidos muestran la relación entre la distancia (en metros) y la intensidad de la señal WiFi (RSSI, por sus siglas en inglés Received Signal Strength Indicator), medida en

decibelios-milivatios (dBm). Se registraron mediciones cada metro desde 1 hasta 10 metros, promediando varias lecturas en cada punto para reducir el error y obtener un valor representativo de la potencia de la señal.

El valor de RSSI es una medida negativa que indica cuán fuerte llega la señal al receptor: cuanto más cercano a cero sea el número, mayor será la intensidad de la señal; por el contrario, cuanto más negativo sea, más débil será la conexión.
Al analizar los resultados:

- A 1 metro, la señal fue de -50,2 dBm, lo que indica una conexión fuerte y estable.
- Entre 2 y 4 metros, la potencia disminuyó gradualmente hasta -56,2 dBm, mostrando una pérdida moderada de intensidad a medida que aumentó la distancia.
- Desde 5 hasta 10 metros, la caída se volvió más pronunciada, alcanzando un valor mínimo de -66,5 dBm a 10 metros.
-

Esta tendencia es coherente con el comportamiento esperado de las ondas de radio: a medida que la distancia entre el emisor (punto de acceso) y el receptor (Raspberry Pi Pico W o teléfono celular) aumenta, la potencia recibida disminuye de manera aproximadamente logarítmica.

HTTP es un protocolo que opera sobre TCP (Transmission Control Protocol), el cual establece una conexión confiable mediante un proceso de tres pasos conocido como “three-way handshake”. Este mecanismo garantiza que los datos enviados lleguen completos y en el orden correcto al receptor. Por esa razón, HTTP se utiliza en aplicaciones donde la exactitud y la fiabilidad son esenciales, como el control de dispositivos, la transferencia de archivos o el acceso a páginas web. Sin embargo, esta confiabilidad conlleva una mayor latencia y consumo de recursos, debido a las confirmaciones necesarias en cada transmisión.

UDP (User Datagram Protocol), en cambio, funciona sin establecer conexión previa. Envía los datos en forma de datagramas directamente al destino, sin verificar si fueron recibidos correctamente. Este método permite una comunicación mucho más rápida y eficiente, aunque con la posibilidad de pérdida de paquetes o errores en la entrega. UDP es ideal para aplicaciones donde la velocidad es prioritaria sobre la fiabilidad, como en transmisiones de video, audio en tiempo real o lectura continua de sensores.

El análisis de estos dos protocolos permite comprender que ambos son complementarios dentro del entorno IoT: HTTP es apropiado para procesos que requieren control y precisión, mientras que UDP resulta más adecuado en situaciones que demandan rapidez, menor consumo de tiempo y tolerancia a pequeñas pérdidas de información. En el caso de la Raspberry Pi Pico W, la elección entre uno u otro depende del tipo de aplicación que se desarrolle y de los requerimientos específicos de comunicación.

III. REFERENCIAS

[1] “Pico 2 W Download.” Accessed: Aug. 17, 2025. [Online]. Available: https://circuitpython.org/board/raspberry_pi_pico2_w/