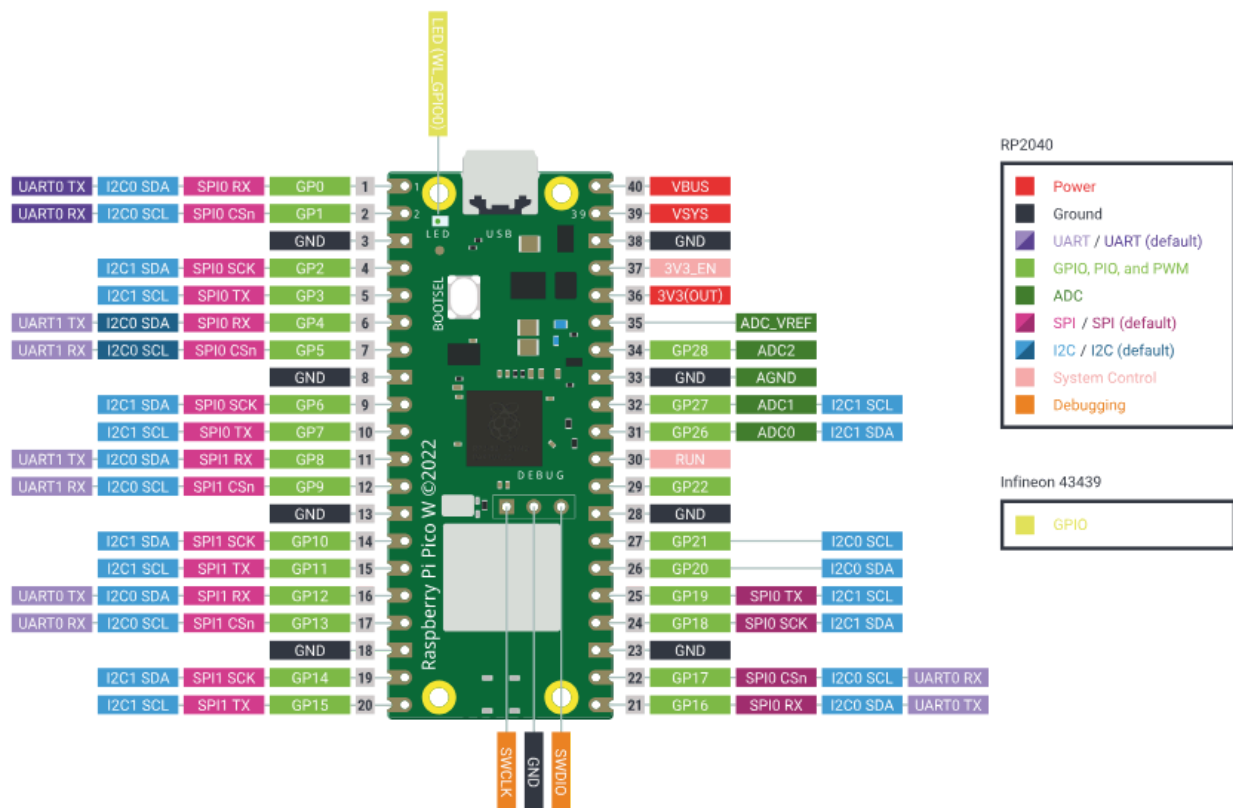


**Wprowadzenie.** Płytki Raspberry Pi Pico W, w porównaniu z Raspberry Pi Pico, jest wyposażona w układ Infineon CYW43439, który pozwala na komunikację w paśmie 2.4GHz (802.11n). Podstawowe parametry techniczne i funkcjonalne są następujące:

- wsparcie dla WPA3;
- wbudowana antena, na licencji firmy ABRACON (poprzednio ProAnt)
- możliwość uruchomienia punktu dostępowego dla max. 4 klientów;
- interfejs Bluetooth 5.2.

Pozostałe parametry takie, jak GPIO, interfejsy przewodowe, rdzenie obliczeniowe, pamięć danych, pamięć programu są identyczne jak w przypadku płytki Raspberry Pi Pico.



Układ do komunikacji bezprzewodowej jest podłączony do mikrokontrolera RP2040 za pomocą interfejsu SPI.

UWAGA. Ze względu na ograniczenia w ilości pinów, niektóre piny, które są wykorzystane do interfejsu bezprzewodowego mają ograniczoną funkcjonalność. CLK jest współdzielony z VSYS, co oznacza, że odczyt VSYS jest możliwy w przypadku braku przesyłu danych z wykorzystaniem SPI. Linia DIN/DOUT układ Infineon CYW43439 jest podłączona do układu RP2040 poprzez GPIO24. Oznacza to, że przerwania związane z tym GPIO mogą nie działać poprawnie w przypadku transmisji z wykorzystaniem interfejsu SPI.

**Zad 1.** Skonfiguruj płytkę Raspberry Pi Pico W do pracy ze środowiskiem Thonny IDE. Niezbędne informacje znajdują się na stronie:

`projects.raspberrypi.org/en/projects/get-started-pico-w/1`

Wszystkie kody są dostępne w repozytorium GitHub, w pliku - `ex05_wifi.zip`.

**Task 2.** Podłącz płytkę Raspberry Pi Pico W do bezprzewodowej sieci ASUS.

```
import time
from network import WLAN, STA_IF
import gc

ssid = "ASUS"
password = "Doktorzec1"

gc.collect()

wlan = WLAN(STA_IF)
wlan.active(True)
wlan.connect(ssid,password)

max_wait = 10
while max_wait > 0:
```

```
if wlan.status() < 0 or wlan.status() >= 3:
    break
max_wait -= 1
print("Connecting to ",ssid)
time.sleep(1)

if wlan.status() != 3:
    raise RuntimeError("Network connection failed")
else:
    print("Connected")
    status = wlan.ifconfig()
    print("IP adres " + status[0])
wlan.disconnect()
```

Kody, określające status połączenia, są opisane na stronie 16, w dokumentacji *Connecting to the Internet with Raspberry Pi Pico W*:

[www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html](http://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html)

### **Zad 3.** Funkcja `wifi_connect()`.

```
from time import sleep
from network import WLAN, STA_IF
from machine import reset
import gc

gc.collect()
ssid = "ASUS"
password = "Doktorzec1"

def wifi_connect():
    wlan = WLAN(STA_IF)
    wlan.active(True)
    wlan.connect(ssid,password)
    while wlan.isconnected() == False:
        print("Connection to " + ssid)
        sleep(1)
    print(wlan.ifconfig())
    print(f'Connected to {ssid} network, IP {wlan.ifconfig()[0]}')
```

```
try:
    wifi_connect()
except KeyboardInterrupt:
    reset()
```

### **Zad 4. MQTT - publikacja.**

```
import time
import network
from machine import Pin
from umqtt import MQTTClient
import gc
import random

gc.collect()

ssid = "ASUS"
password = "Doktorzec1"

# Connect to WiFi
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect(ssid, password)
print(f'Connecting to the {ssid} network')
while wlan.isconnected() == False:
    print('Waiting for connection...')
    time.sleep(0.5)
print(f'Connected to the {ssid} network')
print(f'IP adres is {wlan.ifconfig()[0]}')

mqtt_server = "192.168.1.234"
mqtt_username = "" # username
mqtt_password = "" # user password
mqtt_publish_topic = "number"
mqtt_client_id = "rpi_pico_w"

mqtt_client = MQTTClient(
    client_id=mqtt_client_id,
    server=mqtt_server,
    # user=mqtt_username,
    # password=mqtt_password
```

```
)

mqtt_client.connect()

try:
    message_cnt = 0
    while message_cnt < 20:
        val = random.randint(0, 100)
        print(f'Publish {val:.2f}')
        mqtt_client.publish(mqtt_publish_topic, str(val))
        message_cnt += 1
        time.sleep(2)
except Exception as e:
    print(f'Failed to publish message: {e}')
finally:
    mqtt_client.disconnect()
    print("MQTT client disconnected")
    wlan.disconnect()
```

### **Zad 5. MQTT - subskrypcja.**

```
import network
import time
from umqtt import MQTTClient
import gc

gc.collect()

ssid = "ASUS"
password = "Doktorzecl"

# Connect to WiFi
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect(ssid, password)
print(f'Connecting to the {ssid} network')
while wlan.isconnected() == False:
    print('Waiting for connection...')
    time.sleep(0.5)
print(f'Connected to the {ssid} network')
print(f'IP adres is {wlan.ifconfig()[0]}')
```

```
# Print out of what was received over MQTT
def mqtt_subscription_callback(topic, message):
    print(f'Topic {topic}')
    print(f'Received message {message}')

mqtt_server = "192.168.1.234"
mqtt_publish_topic = "number"
mqtt_client_id = "rpi_pico_w"
mqtt_client = MQTTClient(
    client_id=mqtt_client_id,
    server=mqtt_server)
mqtt_client.set_callback(mqtt_subscription_callback)
mqtt_client.connect()

# Subscribe to the MQTT topic
mqtt_topic = 'mqtt_num'
mqtt_client.subscribe(mqtt_topic)
print(f'Connected and subscribed to the {mqtt_topic}')

try:
    while True:
        print(f'Waiting for messages on {mqtt_topic}')
        mqtt_client.wait_msg()
except Exception as e:
    print(f'Failed to wait for MQTT messages: {e}')
finally:
    mqtt_client.disconnect()
```

### **Zad 6. MQTT - publikacja JSON.**

```
import time
import network
from machine import Pin
from umqtt import MQTTClient
import gc
import random
import json

gc.collect()
```

## Ćwiczenie nr 5: Komunikacja bezprzewodowa

---

```
ssid = "ASUS"
password = "Doktorzec1"

data = {
    "device" : "Rpi Pico W",
    "dev_value" : 10
}

# Connect to WiFi
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect(ssid, password)
print(f'Connecting to the {ssid} network')
while wlan.isconnected() == False:
    print('Waiting for connection...')
    time.sleep(0.5)
print(f'Connected to the {ssid} network')
print(f'IP adres is {wlan.ifconfig()[0]}')

mqtt_server = "192.168.1.234"
mqtt_username = "" # username
mqtt_password = "" # user password
mqtt_publish_topic = "mqtt_json"
mqtt_client_id = "rpi_pico_w"

mqtt_client = MQTTClient(
    client_id=mqtt_client_id,
    server=mqtt_server,
    # user=mqtt_username,
    # password=mqtt_password
)

mqtt_client.connect()

try:
    message_cnt = 0
    while message_cnt < 20:
        data['dev_value'] = random.randint(0, 100)
        json_str = json.dumps(data)
        print(f'Publish {json_str}')
        mqtt_client.publish(mqtt_publish_topic, json_str)
        message_cnt += 1
```

```
        time.sleep(2)
except Exception as e:
    print(f'Failed to publish message: {e}')
finally:
    mqtt_client.disconnect()
    print("MQTT client disconnected")
    wlan.disconnect()
```

### **Zad 7. (1 pkt.)** Komunikacja MQTT.

- (0.1 pkt.) uruchomić serwer MQTT na płytce Raspberry Pi;
- (0.1 pkt.) podłączyć wybrany czujnik i przesyłać wyniki pomiarów w czasie rzeczywistym na dashboard Node-RED;
- (0.1 pkt.) podłączyć 2 czujnik i przesyłać z niego dane na dashboard Node-RED;
- (0.3 pkt.) opisy wszystkich elementów na interfejsie powinny być przesyłane z płytki Raspberry Pi Pico;
- (0.4 pkt.) format wymiany danych JSON.

**Zad 8. (1.5 pkt.)** Zaimplementować wszystkie funkcjonalności z zadania 7. Podłączyć wybrany moduł wyświetlacza *LCD-Pico*. Na wyświetlaczu należy wyświetlić wyniki pomiaru, status połączenia sieciowego oraz komunikaty wysyłane z poziomu interfejsu użytkownika. Format wymiany danych JSON.



**Dla zainteresowanych:**

1. Raspberry Pi Pico W vs. Pico. Jakie są różnice?  
[core-electronics.com.au/guides/raspberry-pi-pico-w-vs-pico-whats-the-difference/](https://core-electronics.com.au/guides/raspberry-pi-pico-w-vs-pico-whats-the-difference/)
2. Strona projektu MicroPython:  
[micropython.org/download/rp2-pico/](https://micropython.org/download/rp2-pico/)
3. Konfiguracja Raspberry Pi Pico W:  
[projects.raspberrypi.org/en/projects/get-started-pico-w/](https://projects.raspberrypi.org/en/projects/get-started-pico-w/)
4. Dokumentacja do Raspberry Pi Pico oraz Pico W:  
[www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html](https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html)
5. Dokumentacja modułów Pico-LCD:  
[www.waveshare.com/wiki/Pico-LCD-0.96](https://www.waveshare.com/wiki/Pico-LCD-0.96)