

Podstawy Systemów Mikroprocesorowych

Laboratorium

Laboratorium 0: Uruchomienie stanowiska i podstawy pracy z Raspberry
Pi Pico 2 W

Autorzy: Dariusz Tefelski, Angelika Tefelska

Spis treści

1	Wprowadzenie	1
2	Wyposażenie stanowiska	1
3	Zadanie 0: Instalacja MicroPythona i konfiguracja Thonny	3
4	Zadanie 1: Dioda LED	4
5	Zadanie 2: Przycisk	6
6	Zadanie końcowe: Czyszczenie pamięci flash	7
7	Podsumowanie	7

1 Wprowadzenie

Cel ćwiczenia



Celem pierwszych zajęć jest przygotowanie stanowiska do dalszej pracy z mikrokontrolerem Raspberry Pi Pico 2 W oraz środowiskiem MicroPython. Po wykonaniu ćwiczenia student powinien:

- potrafić poprawnie zasilić i podłączyć stanowisko,
- uruchomić tryb BOOTSEL i wgrać interpreter MicroPython z pliku UF2,
- skonfigurować środowisko Thonny do pracy z Raspberry Pi Pico 2 W,
- rozróżniać cyfrowe wejścia i wyjścia mikrokontrolera,
- sterować diodą LED i odczytać stan przycisku,
- wykonać podstawową obserwację sygnału elektrycznego za pomocą Analog Discovery II,
- wyczyścić pamięć flash mikrokontrolera po zakończeniu zajęć.

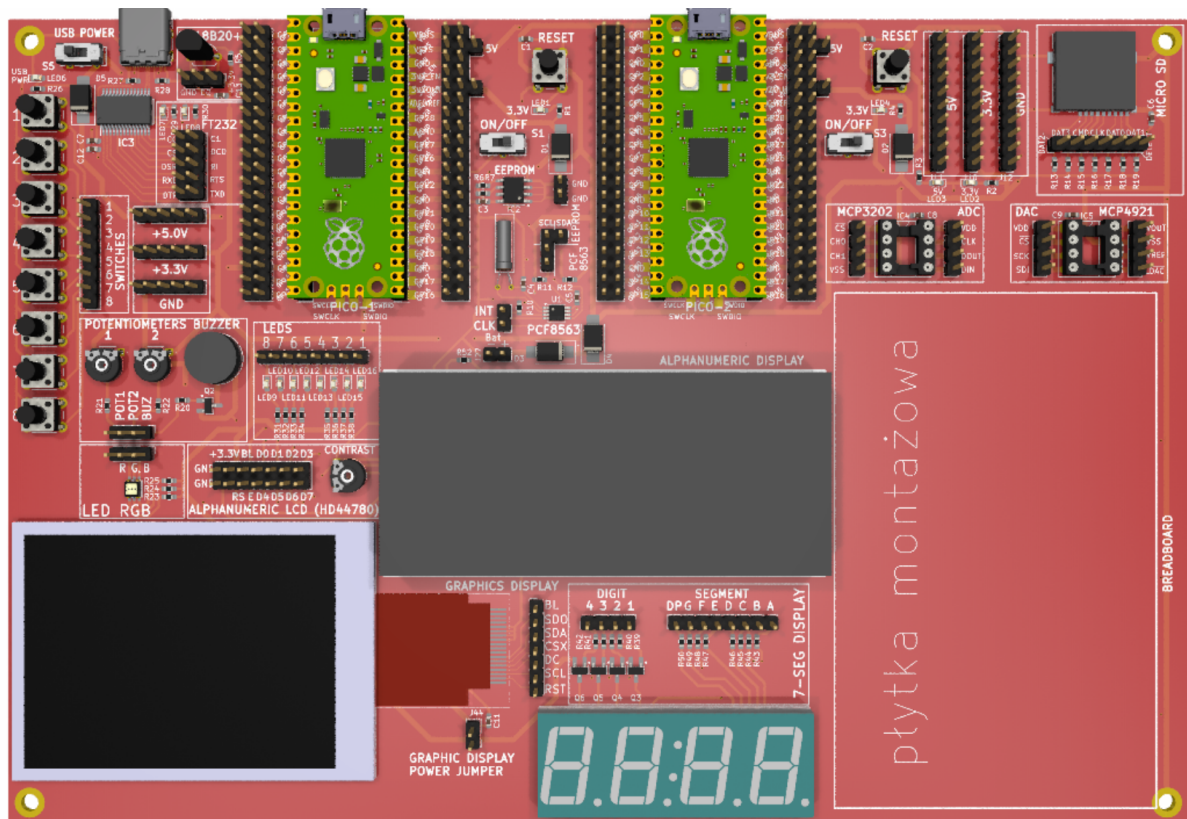
Stanowisko dydaktyczne wyposażone jest w dwa moduły Raspberry Pi Pico 2 W oraz szereg elementów peryferyjnych, między innymi diody LED, przyciski, diodę RGB, potencjometry, buzzer, czujnik DS18B20, zegar RTC, pamięć EEPROM, gniazdo microSD, wyświetlacze oraz zewnętrzne przetworniki A/C i C/A. W kolejnych laboratoriach elementy te będą wykorzystywane do poznawania interfejsów i mechanizmów typowych dla systemów mikroprocesorowych.

Na tych zajęciach korzystamy tylko z niewielkiej części możliwości płytki. Najważniejsze jest poprawne uruchomienie środowiska oraz wykonanie pierwszych pomiarów.

2 Wyposażenie stanowiska

Do wykonania ćwiczenia potrzebne są:

- płytka dydaktyczna z Raspberry Pi Pico 2 W,
- przewód USB do połączenia Pico z komputerem,
- przewody połączeniowe,
- Analog Discovery II,
- komputer z programami Thonny i WaveForms.



Rysunek 1: Płytką dydaktyczną z Raspberry Pi Pico 2W.

3 Zadanie 0: Instalacja MicroPythona i konfiguracja Thonny

Raspberry Pi Pico 2 W będzie programowane w języku MicroPython. Jest to implementacja języka Python przeznaczona dla mikrokontrolerów. Interpreter MicroPythona musi być zapisany w pamięci flash mikrokontrolera. Na początku zajęć samodzielnie wgrasz go z pliku w formacie UF2 (*USB Flashing Format*).

Tryb BOOTSEL



Raspberry Pi Pico 2 W posiada zapisany w pamięci ROM bootloader. Przytrzymanie przycisku **BOOTSEL** podczas włączania zasilania uruchamia specjalny tryb programowania. W tym trybie mikrokontroler jest widoczny przez komputer jako pamięć masowa USB i można wgrać do niego firmware w formacie UF2.

Zadanie 0.1 — Wgranie MicroPythona



Wykonaj kolejno następujące czynności:

1. Pobierz aktualną stabilną wersję MicroPythona przeznaczoną dla **Raspberry Pi Pico 2 W** ze strony:

https://micropython.org/download/RPI_PICO2_W/

Wybierz plik z rozszerzeniem `.uf2` z sekcji stabilnych wydań.

2. Ustaw przełącznik zasilania używanego modułu Pico w pozycji **OFF**.
3. Naciśnij i przytrzymaj przycisk **BOOTSEL** znajdujący się na module Raspberry Pi Pico 2 W.
4. Trzymając przycisk **BOOTSEL**, włącz zasilanie modułu przełącznikiem **ON**. Po pojawieniu się w systemie nowego dysku USB zwolnij przycisk **BOOTSEL**.
5. Skopiuj pobrany plik UF2 na nowo wykryty dysk. Po zakończeniu programowania Pico automatycznie uruchomi się ponownie, a dysk USB zniknie.
6. Uruchom program Thonny. W prawym dolnym rogu okna wybierz interpreter:

MicroPython (Raspberry Pi Pico)

oraz właściwy port szeregowy.

7. Sprawdź działanie interpretera, wpisując w powłocze:

```
1 print("Hello Pico!")
```

Uwaga



Plik UF2 musi być przeznaczony dla modelu **Raspberry Pi Pico 2 W**. Nie używaj firmware dla Raspberry Pi Pico, Raspberry Pi Pico W ani Raspberry Pi Pico 2.

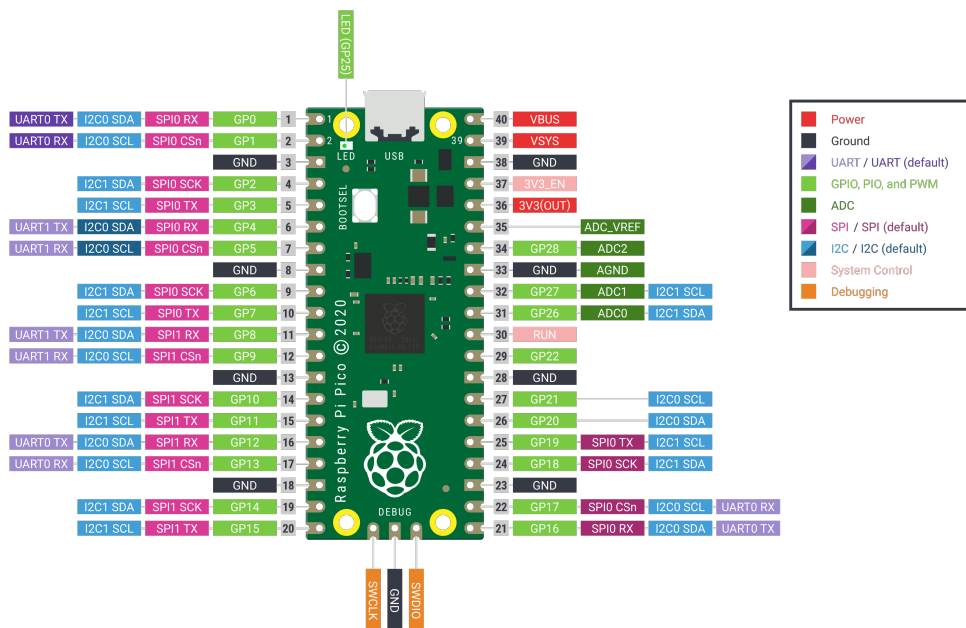
Warto wiedzieć



Powłoka widoczna w dolnej części programu Thonny to **REPL** (Read–Evaluate–Print Loop). Pozwala ona wykonywać pojedyncze polecenia bez zapisywania całego programu. Jest bardzo przydatna podczas uruchamiania sprzętu i szybkiego sprawdzania połączeń.

4 Zadanie 1: Dioda LED

Moduł Raspberry Pi Pico 2 W udostępnia 26 wielofunkcyjnych pinów GPIO (*General Purpose Input/Output*). Na schemacie płytki są one oznaczone jako GPx , gdzie x jest numerem GPIO.



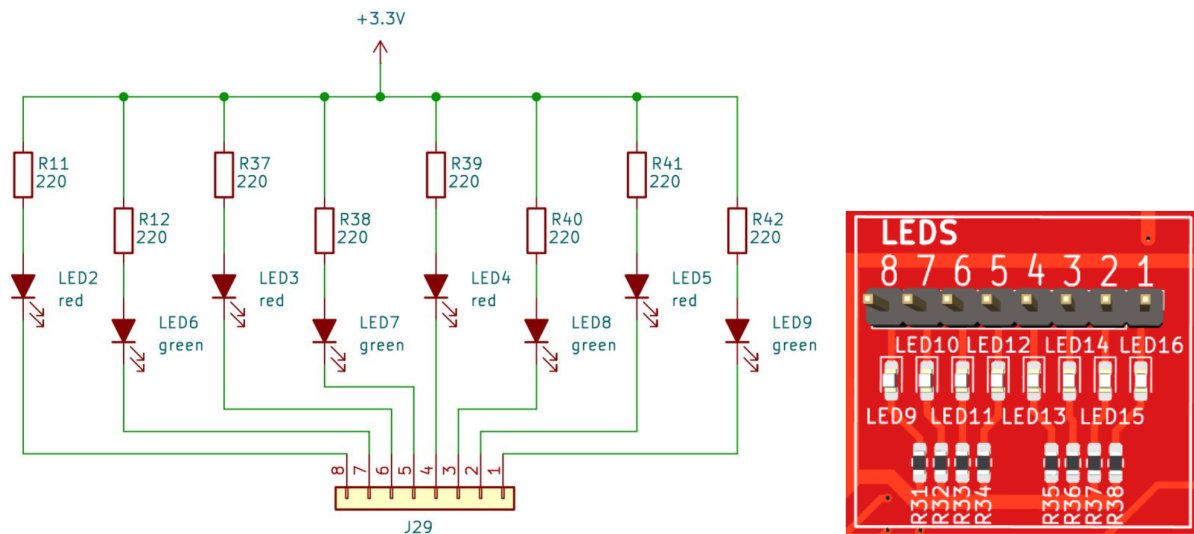
Rysunek 2: Wyprowadzenia modułu Raspberry Pi Pico 2 W.

Piny GPIO mogą pełnić funkcję pinów zarówno wejściowych jak i wyjściowych. Piny wejściowe używamy wtedy gdy za ich pomocą mikrokontroler odczytuje jakąś wartość np. z czujnika temperatury, czujnika ruchu czy przycisku. Piny wyjściowe są używane wtedy gdy za ich pomocą sterujemy innymi elementami np. silnikiem, serwomechanizmem, diodą LED czy wyświetlaczem. Więcej na ten temat na kolejnych zajęciach.

Do sterowania pojedynczym pinem cyfrowym wykorzystujemy klasę `Pin` z modułu `machine`. W pierwszym programie użyjemy jednej z diod LED znajdujących się na płytce i połączymy ją z pinem `GP15`.

Na płytce diody LED są podłączone z jednej strony do 3,3V przez rezystor a z drugiej strony do pinu sterującego (patrz rysunek 3). Oznacza to, że są sterowane w logice **ak-**

tywnej stanem niskim: ustawienie GPIO na 0 powoduje przepływ prądu i zapalenie diody, a ustawienie 1 — jej zgaszenie.



Rysunek 3: Podłączenie diod LED na płytce dydaktycznej.

Zadanie - Migająca dioda

🔗 Połącz wybraną diodę LED z pinem GP15. Następnie uruchom program:

```
1 from machine import Pin
2 from utime import sleep
3
4 led = Pin(15, Pin.OUT)
5
6 while True:
7     led.value(0)          # LED włączona
8     sleep(0.5)
9     led.value(1)          # LED wyłączona
10    sleep(0.5)
```

Zmodyfikuj program tak, aby:

1. dioda migała dwa razy szybciej,
2. po włączeniu programu rozpoczynała pracę od stanu wyłączonego.

Co rzeczywiście znajduje się na pinie GPIO?



Uruchom program WaveForms i wybierz narzędzie **Scope**. Podłącz kanał pomiarowy Analog Discovery II do pinu GP15, a masę urządzenia do GND płytki. Na potrzeby obserwacji ustaw w programie czasy opóźnienia na około 5 ms. Zaobserwuj przebieg prostokątny i oszacuj:

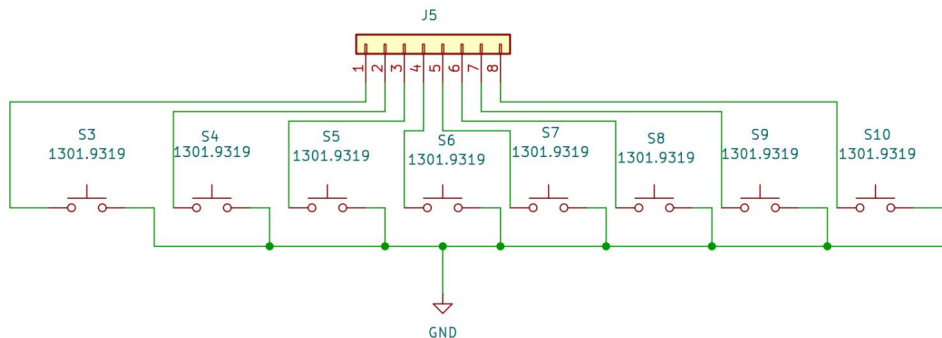
- napięcie stanu niskiego i wysokiego,
- amplitudę sygnału,



- okres i częstotliwość przebiegu.
Porównaj zmierzoną częstotliwość z wartością wynikającą z programu.

5 Zadanie 2: Przycisk

Przyciski znajdujące się na płytce po naciśnięciu zwierają odpowiadający im pin do masy (patrz rysunek 4).



Rysunek 4: Podłączenie przycisków z pinami sterującymi.

Aby stan wejścia był jednoznaczny również wtedy, gdy przycisk nie jest naciśnięty, wykorzystamy wewnętrzny rezystor podciągający *pull-up* mikrokontrolera.

Zadanie - Odczyt stanu przycisku



Połącz jeden z przycisków do pinu GP14 i skonfiguruj go jako element wejściowy z podłączonymi rezystorami *pull-up*:

```
1 from machine import Pin
2 from utime import sleep
3
4 button = Pin(14, Pin.IN, Pin.PULL_UP)
5
6 while True:
7     print(button.value())
8     sleep(1)
```

Sprawdź wartości zwracane dla przycisku puszczonego i naciśniętego.

Następnie rozbuduj program, w taki sposób by dioda LED była zapalana/gaszona po naciśnięciu przycisku analogicznie jak włącznik światła.

Warto wiedzieć



W kolejnych laboratoriach zobaczymy, że częste odpytywanie przycisku w pętli nie zawsze jest najlepszym rozwiązaniem. Poznamy przerwania, które pozwolą mikrokontrolerowi reagować na zdarzenie bez ciągłego sprawdzania stanu wejścia.

6 Zadanie końcowe: Czyszczenie pamięci flash

Po zakończeniu zajęć należy pozostawić mikrokontroler w stanie czystym, tak aby kolejna grupa rozpoczynała pracę od samodzielnego wgrania MicroPythona.

Uwaga



Wykonanie tego zadania usuwa zawartość pamięci flash używanego Raspberry Pi Pico 2 W, w tym interpreter MicroPython oraz pliki zapisane przez użytkownika. W tym laboratorium jest to działanie **zamierzone**.

Zadanie — Wyczyszczenie mikrokontrolera



1. Pobierz plik `flash_nuke.uf2` z sekcji *Resetting flash memory* dokumentacji Raspberry Pi:
<https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html#reset-flash-memory>
2. Ustaw przełącznik zasilania używanego modułu Pico w pozycji **OFF**.
3. Naciśnij i przytrzymaj przycisk **BOOTSEL**, a następnie włącz zasilanie modułu przełącznikiem **ON**.
4. Po pojawieniu się dysku USB zwolnij przycisk **BOOTSEL**.
5. Skopiuj plik `flash_nuke.uf2` na wykryty dysk. Mikrokontroler automatycznie uruchomi program czyszczący i wyczyści pamięć flash.
6. **Nie wgrywaj ponownie MicroPythona.** Następna grupa rozpocznie Laboratorium 0 od Zadania 0.1 i samodzielnie zainstaluje interpreter.

7 Podsumowanie

Po tych zajęciach powinieneś umieć



- uruchomić Pico 2 W w trybie **BOOTSEL** i wgrać firmware z pliku `UF2`,
- skonfigurować Thonny i uruchomić prosty skrypt w MicroPythonie,
- skonfigurować pin jako wejście i wyjście,
- wyjaśnić działanie rezystora *pull-up*,
- wykonać podstawową obserwację sygnału za pomocą Analog Discovery II,
- wyczyścić pamięć flash mikrokontrolera po zakończeniu pracy.

Uwaga



Przed opuszczeniem stanowiska upewnij się, że wykonano zadanie czyszczenia pamięci flash, następnie wyłącz zasilanie płytki i uporządkuj przewody oraz elementy używane podczas ćwiczenia.