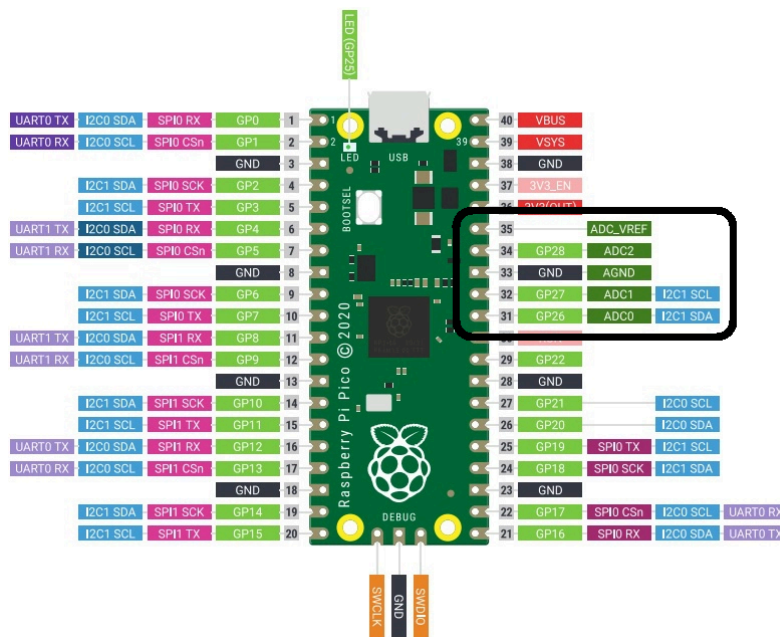


## Ćwiczenie nr 2: Wejścia analogowe

**Wprowadzenie.** Przetwornik A/C w RP2040 ma rozdzielczość 12 bitów, co oznacza, że kolejne próbki sygnału analogowego, podanego na wejście wybranego kanału, zostaną przekształcone w liczbę całkowitą z przedziału od 0 do 4095. Dzięki zastosowanej w module *machine* MicroPython'a technice nadpróbkowania rozdzielczość przetwarzania A/C została zwiększona do 16 bitów, co oznacza wartość liczbową od 0 do 65535. RP2040 posiada przetwornik 5-kanałowy. 4 kanały są dostępne poprzez wybrane linie GPIO: GP26, GP27, GP28 and GP29. Na płytkach Raspberry Pi Pico, pierwsze 3 z nich są dostępne dla użytkownika, a 4 umożliwia pomiar napięcia VSYS. Piąty kanał przetwornika A/C jest podłączony do czujnika temperatury wbudowanego w mikrokontroler RP2040.



Do odczytu wartości analogowej, proporcjonalnej do napięcia na danym kanale, można wykorzystać numer wyprowadzenia (pin) lub numer kanału:

```
adc = machine.ADC(26) # wyprowadzenie - GP26
adc = machine.ADC(0)  # kanał - odpowiada GP26
```

**Zad 1.** Napisz skrypt w MicoPython'ie, który odczyty i wyśle do komputera wartość temperatury mierzonej przez wbudowany w RP2040 czujnik.

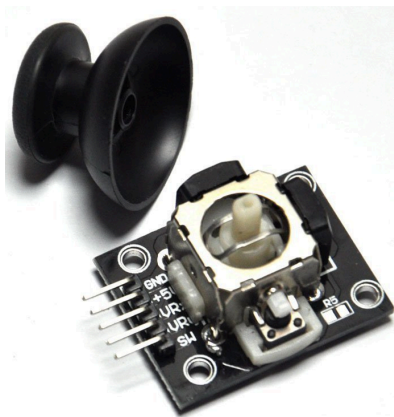
```
from machine import ADC
from utime import sleep

temp_sensor = ADC(4) # channel

while True:
    result = temp_sensor.read_u16()
    temperature = 27 - (result*3.3/65535 - 0.706)/0.001721
    print(temperature)
    sleep(2)
```

Zwykle, 0.706V oznacza 27 stopni Celsjusza, przy wartości współczynnika U/T równej -1.721mV (0.001721V) na stopień Celsjusza.

**Zad 2.** Podłącz joystick do płytki Raspberry Pi Pico.



płytki Pico	joystick
Gnd	Gnd
+3V3	+5V
GP26	VRx
GP27	VRy
GP15	SW

## Ćwiczenie nr 2: Wejścia analogowe

---

Napisz skrypt w MicroPython'ie, który wysyła do komputera wartości proporcjonalne do wychylenia drążka joysticka.

```
from machine import ADC
from utime import sleep

joy_X = ADC(26)
joy_Y = ADC(27)

while True:
    t_string = "X position = " + str(joy_X.read_u16())
    print(t_string)
    t_string = "Y position = " + str(joy_Y.read_u16())
    print(t_string)
    sleep(0.5)
```

**Zad 3.** Napisz skrypt w MicroPython'ie, który wyśle do komputera wartość z przedziału od 0 do 10, proporcjonalnie do wychylenia drążka joysticka.

```
from machine import ADC
from utime import sleep

def analog_map(x, in_min, in_max, out_min, out_max):
    return int((x-in_min)*(out_max-out_min)/(in_max-in_min)+ out_min)

joy_X = ADC(26)

while True:
    result = analog_map(joy_X.read_u16(), 0, 65535, 0, 10)
    t_string = "X position = " + str(result)
    print(t_string)
    sleep(1)
```

**Zad 4.** Podłącz wyświetlacz *Pico-LCD-...* do płytki Raspberry Pi Pico.



Dokumentacja i pliki z przykładami (*Demo code*) znajduje się w zakładce *Resources* na stronie:

[www.waveshare.com/wiki/Pico-LCD-0.96](http://www.waveshare.com/wiki/Pico-LCD-0.96)

**Zad 5. (0.5 punkt)** Napisz skrypt w MicroPython'ie, który wyświetli wartość temperatury otoczenia na wyświetlaczu OLED. Wysokość znaków do najmniej 36px.

**Zad 6. (1 punkt)** Wykorzystaj joystick do kontroli położenia zapalonego segmentu na bargrafie. Przycisk joysticka powinien pozwalać na włączenie i wyłączenie segmentu na bargrafie.

**Zad 7. (1 punkt)** Napisz skrypt w Micropython'ie, który pozwoli na kontrolę położenia znaku na ekranie wyświetlacza OLED za pomocą joysticka.

**Zad 8. (1 punkt)** Napisz skrypt w MicroPython'ie, który wygeneruje wykres temperatury otoczenia na ekranie wyświetlacza OLED.

**Do zaliczenia zadania wymagana jest prezentacja i przesłanie rozwiązania na adres wskazany w stopce instrukcji. Przed wysłaniem plik należy spakować lub zmienić rozszerzenie na txt.**

### **Dla zainteresowanych:**

1. Strona MicroPython:

[micropython.org/download/rp2-pico/](https://micropython.org/download/rp2-pico/)

2. RandomNerd tutorial - wejścia analogowe:

[randomnerdtutorials.com/raspberry-pi-pico-analog-inputs-micropython](https://randomnerdtutorials.com/raspberry-pi-pico-analog-inputs-micropython/)  
[/](#)