

MicroPython

Coding / RP2040 examples

GPIO, ADC, PWM and I2C essentials for a Raspberry Pi Pico.

01 Core API

Digital

`Pin(n, Pin.OUT).value(1); Pin(n, Pin.IN, Pin.PULL_UP).value()`.

ADC

`ADC(26).read_u16()`: scaled 0-65535; voltage ~ reading x 3.3 / 65535.

PWM

`PWM(Pin(15)); freq(1000); duty_u16(32768)` for roughly 50% duty.

I2C

`I2C(0, sda=Pin(4), scl=Pin(5), freq=100000); scan()` finds addresses.

02 Sensor patterns

Read registers

`i2c.readfrom_mem(address, register, count)`; decode bytes per sensor datasheet.

Errors

Catch `OSError` for bus failures and retry after a delay.

Cleanup

Call `pwm.deinit()` when finished. Pin choices and peripheral IDs depend on the port.

Code example / python

MicroPython / continued

Coding / RP2040 examples

```
from machine import Pin, ADC, PWM, I2C
from time import sleep_ms

adc = ADC(26)
pwm = PWM(Pin(15))
pwm.freq(1000)
i2c = I2C(0, sda=Pin(4), scl=Pin(5),
          freq=100000)
print("I2C devices:", i2c.scan())

try:
    while True:
        reading = adc.read_u16()
        pwm.duty_u16(reading)
        print(reading)
        sleep_ms(100)
finally:
    pwm.deinit()
```

Before you power on

The example uses original Pico GPIO assignments. Keep ADC signals within 0-3.3 V.

Datasheets & further reading

MicroPython RP2 reference

<https://docs.micropython.org/en/latest/rp2/quickref.html>