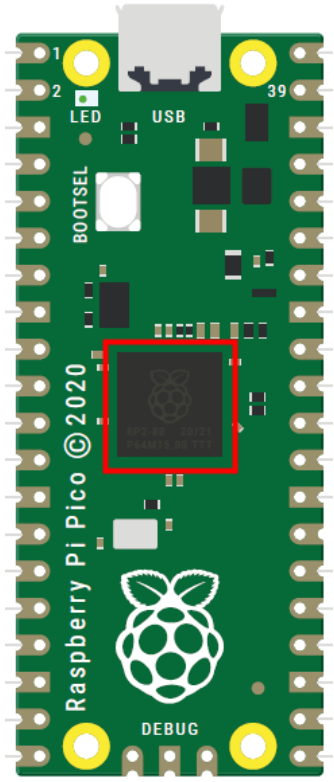
3.2 Onboard temperature sensor

1. Learning Objectives

In this course, we will learn how to read the temperature of the onboard temperature sensor.

2. About Hardware

This course requires no additional hardware and directly utilizes the temperature sensor on the Raspberry Pi Pico board.



The Raspberry Pi Pico's RP2040 microcontroller is a digital device, like all mainstream microcontrollers: it's made up of thousands of transistors, tiny switching devices that either turn on or off. So your PICO can't really understand an analog signal - it can be anything in the spectrum between fully off and fully on - without relying on extra hardware: an analog-to-digital converter (ADC).

An ADC has two key characteristics: its resolution, measured in digital bits, and its channels, or how many analog signals it can accept and convert at a time. The ADC in your PICO has a resolution of 12 bits, which means it can convert an analog signal to a digital signal with numbers ranging from 0 to 4095 - although this is handled in MicroPython to convert from 0 to 65 , a 16-bit number of 535, so it behaves like ADCs on other MicroPython microcontrollers. It has three channels brought to the GPIO pins: GP26, GP27, and GP28, which are also called GP26_ADC0, GP27_ADC1, and GP28_ADC2 for analog channels 0, 1, and 2. There is also a fourth ADC channel, which is connected to a temperature sensor built into the RP2040.

3. About Code

Code path: Code -> 1.Basic course -> 2. On board temperature sensor.py

```
import machine
import time
sensor_temp = machine.ADC(4)
conversion_factor = 3.3 / (65535)
while True:
    reading = sensor_temp.read_u16() * conversion_factor
    temperature = 27 - (reading - 0.706)/0.001721
    print(temperature)
    time.sleep(2)
```

import machine

The machine library contains all the instructions MicroPython needs to communicate with Pico and other MicroPython-compatible devices, extending the language of physical computing.

import time

The "time" library. This library handles everything time related, from measuring it to inserting delays into programs. The unit is seconds.

sensor_temp = machine.ADC(4)

Using ADC channel 4, it is connected to a temperature sensor built into the RP2040.

conversion_factor = 3.3 / (65535)

The level of this pin is 3.3V, and the upper limit of the conversion value is 65535, so the ratio of voltage and number is calculated according to this formula.

reading = sensor_temp.read_u16() * conversion_factor

Calculate the voltage value read by the ADC.

temperature = 27 - (reading - 0.706)/0.001721

Calculates the temperature value of the built-in temperature sensor based on the voltage value.

time.sleep(2)

This calls the sleep function from the utime library, which makes the program pause for any number of seconds you type - in this case 2 seconds.

4. Experimental Phenomenon

After the code is downloaded, we can observe the temperature value through Thony's Shell window.

```python
import machine
import time
sensor_temp = machine.ADC(4)
conversion_factor = 3.3 / (65535)
while True:
    reading = sensor_temp.read_u16() * conversion_factor
    temperature = 27 - (reading - 0.706)/0.001721
    print(temperature)
    time.sleep(2)
```

Shell

```
18.61781
18.61781
18.61781
18.61781
18.61781
18.61781
18.61781
18.61781
```