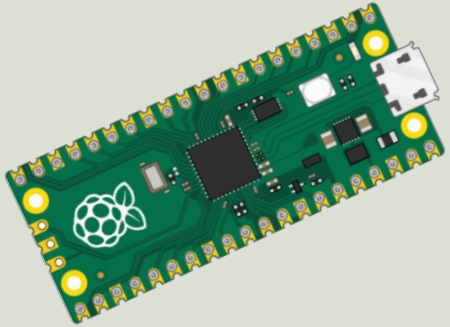
https://www.halvorsen.blog



Raspberry Pi Pico

Using onboard Temperature Sensor

Hans-Petter Halvorsen

Introduction

- In this Tutorial we will use Raspberry Pi Pico
- We will use the built-in Temperature Sensor on the Raspberry Pi Pico hardware
- We will use MicroPython
- We will use the Thonny Python Editor

Raspberry Pi Pico

- Raspberry Pi Pico is a microcontroller board developed by the Raspberry Pi Foundation
- Raspberry Pi Pico has similar features as Arduino devices
- Raspberry Pi Pico is typically used for Electronics projects, IoT Applications, etc.
- You typically use MicroPython, which is a downscaled version of Python, in order to program it

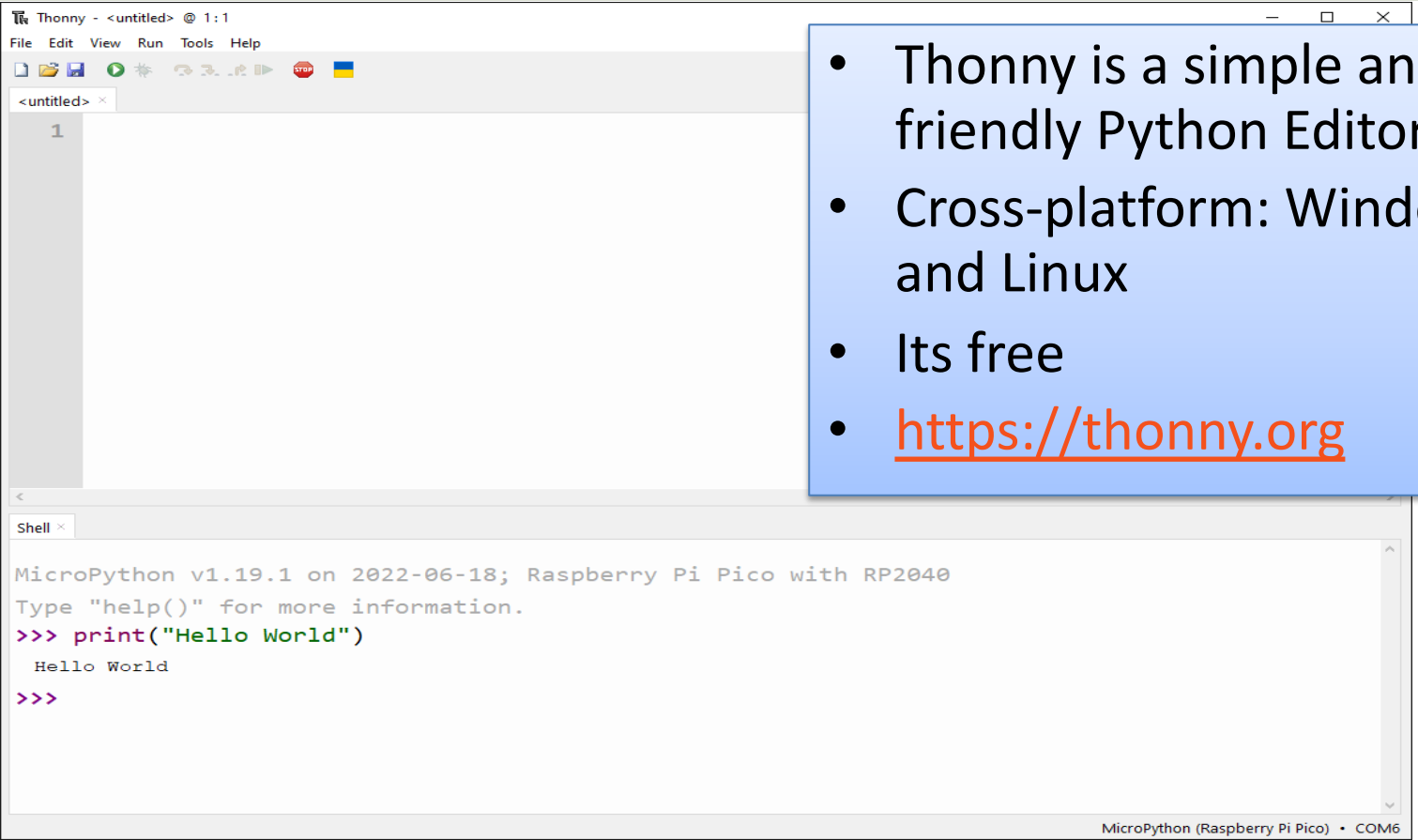
https://www.raspberrypi.com/products/raspberry-pi-pico/

https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico

MicroPython

- MicroPython is a downscaled version of Python
- It is typically used for Microcontrollers and constrained systems (like Pico)
- https://micropython.org
- https://docs.micropython.org/en/latest/index.html

Thonny



- Thonny is a simple and user-friendly Python Editor
- Cross-platform: Windows, macOS and Linux
- Its free
- https://thonny.org

Built-in Temperature Sensor

Hans-Petter Halvorsen

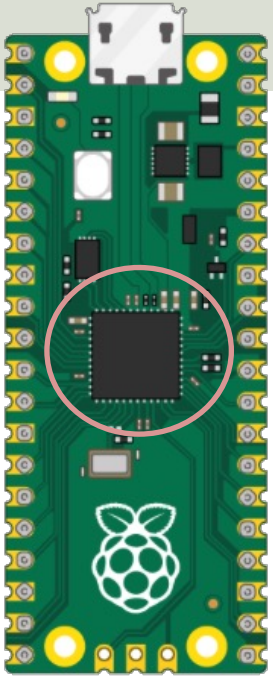
Table of Contents

Built-in Temperature Sensor

- The raspberry Pi Pico has a built-in Temperature Sensor
- The Temperature Sensor is inside the RP2040 microcontroller chip which is located on the Raspberry Pi Pico
- It can be used for simple applications and for test and demo purposes

https://www.raspberrypi.com/documentation/microcontrollers/rp2040.html#internal-temperature-sensor

RP2040 Datasheet



4.9.5. Temperature Sensor

The temperature sensor measures the Vbe voltage of a biased bipolar diode, connected to the fifth ADC channel (AINSEL=4). Typically, Vbe = 0.706V at 27 degrees C, with a slope of -1.721mV per degree. Therefore the temperature can be approximated as follows:

T = 27 - (ADC_voltage - 0.706)/0.001721

As the Vbe and the Vbe slope can vary over the temperature range, and from device to device, some user calibration may be required if accurate measurements are required.

The temperature sensor's bias source must be enabled before use, via CS.TS_EN. This increases current consumption on ADC_AVDD by approximately 40µA.

NOTE

The on board temperature sensor is very sensitive to errors in the reference voltage. If the ADC returns a value of 891 this would correspond to a temperature of 20.1°C. However if the reference voltage is 1% lower than 3.3V then the same reading of 891 would correspond to 24.3°C. You would see a change in temperature of over 4°C for a small 1% change in reference voltage. Therefore if you want to improve the accuracy of the internal temperature sensor it is worth considering adding an external reference voltage.

Datasheet (Chapter 4.9.5. Temperature Sensor)

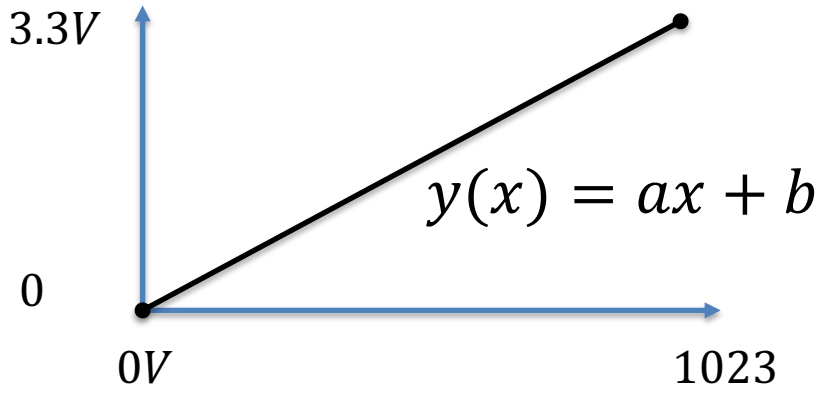
https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf

ADC Value to Voltage Value

Analog Pins: The built-in Analog-to-Digital Converter (ADC) on Pico is 16bit, producing values from 0 to 65535.

The `read_u16()` function gives a value between 0 and 65535. It must be converted to a Voltage Signal 0 - 3.3v

ADC = 0 -> 0v
ADC = 65535 -> 3.3v



This gives the following conversion formula:

$$y(x) = \frac{3.3}{65535}x$$

Main Code Structure

1. Initialization

The internal temperature sensor is connected to an internal ADC pin #4

```
adcpin = 4
sensor = machine.ADC(adcpin)
```

2. Read from ADC:

```
adc_value = sensor.read_u16()
```

3. Convert raw ADC Value (0-65535) to Voltage Value (0-3.3v):

```
volt = (3.3/65535)*adc_value
```

4. Convert from Voltage Value to Temperature in degrees Celsius:

```
temperature = 27 - (volt - 0.706)/0.001721
```

Built-in Temperature Sensor

```
import machine
import time

adcpin = 4
sensor = machine.ADC(adcpin)

def ReadTemperature():
    adc_value = sensor.read_u16()
    volt = (3.3/65535)*adc_value
    temperature = 27 - (volt - 0.706)/0.001721
    return round(temperature, 1)

while True:
    temperature = ReadTemperature()
    print(temperature)
    time.sleep(5)
```

Thonny - C:\Users\hansha\OneDrive\Documents\Industrial IT and Automation\IoT\Raspberry Pi Pico\Code Examples\temperature_sensor_builtin.py @ 17 ...

File Edit View Run Tools Help

temperature_sensor_builtin.py

```python
import machine
import time

adcpin = 4
sensor = machine.ADC(adcpin)

def ReadTemperature():
    adc_value = sensor.read_u16()
    volt = (3.3/65535)*adc_value
    temperature = 27 - (volt - 0.706)/0.001721
    return round(temperature, 1)


while True:
    temperature = ReadTemperature()
    print(temperature)
    time.sleep(5)
```

Shell

```
>>> %Run -c $EDITOR_CONTENT
  27.0
  27.5
  27.0
  26.6
  26.1
  24.2
  24.2
```

MicroPython (Raspberry Pi Pico) • COM7

PicoSensor.py

```
from machine import ADC

class Temperature:
    def __init__(self):
        adcpin = 4
        self.sensor = ADC(adcpin)

    def ReadTemperature(self):
        adc_value = self.sensor.read_u16()
        volt = (3.3/65535)*adc_value
        temperature = 27 - (volt - 0.706)/0.001721
        return round(temperature, 1)
```

Main Program:

```
from PicoSensor import Temperature
import time

sensor = Temperature()

while True:
    temperature = sensor.ReadTemperature()
    print(temperature)
    time.sleep(5)
```

Raspberry Pi Pico Resources

- Raspberry Pi Pico: https://www.raspberrypi.com/products/raspberry-pi-pico/
- Raspberry Pi Foundation: https://projects.raspberrypi.org/en/projects?hardware[]=pico
- Getting Started with Pico: https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico
- Getting started with your Raspberry Pi Pico W: https://projects.raspberrypi.org/en/projects/get-started-pico-w
- MicroPython: https://docs.micropython.org/en/latest/index.html

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog