

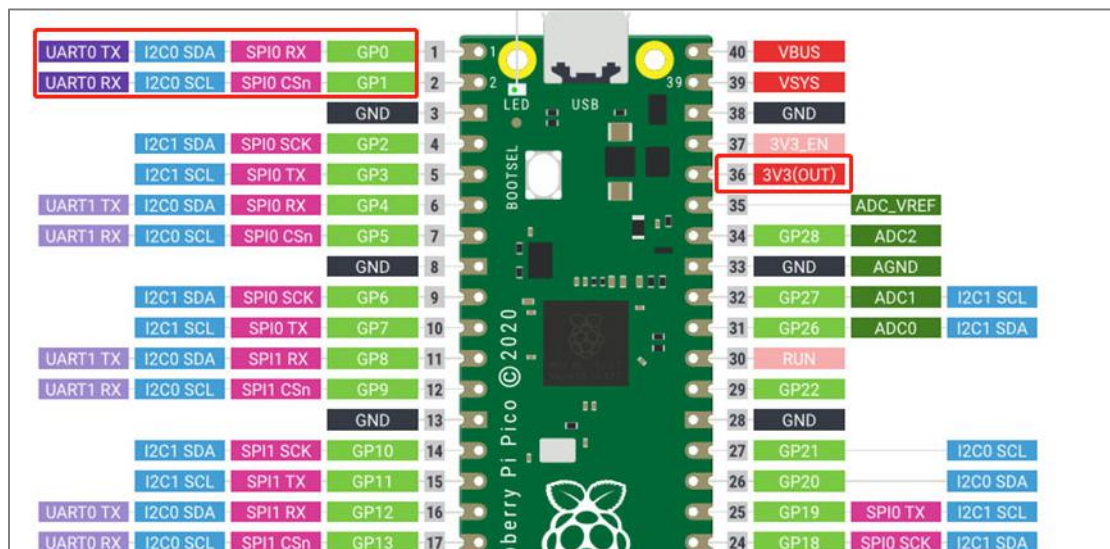
On board temperature sensor

1. Learning purpose

1. Learn how to use ADC pins on the Raspberry Pi Pico board.
2. Learn how to read the temperature of the on board temperature sensor.

2. Hardware construction

This course does not require additional hardware equipment to directly use the temperature sensor on the Raspberry Pi Pico board.



3. About code

Thonny programming

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/gpio.h"
#include "hardware/adc.h"

int main() {
    stdio_init_all();
    printf("ADC Example, measuring GPIO26\n");

    adc_init();

    adc_set_temp_sensor_enabled(true);

    adc_select_input(4);
```

```

while (1) {
// 12-bit conversion, assume max value == ADC_VREF == 3.3 V
const float conversion_factor = 3.3f / (1 << 12);
uint16_t result = adc_read();
float reading = adc_read() * conversion_factor;
float temperature = 27 - (reading - 0.706) / 0.001721;
printf("reslut: 0x%x, reading: %f V, temperature: %f \n", result, reading, temperature);

sleep_ms(500);
}
}

```

`#include "hardware/adc.h"`

Commonly used adc hardware library files.

`stdio_init_all()`

Initialization

`adc_init()`

Initialize ADC.

`adc_set_temp_sensor_enabled(true)`

Set whether to supply power to the on board temperature sensor.

`adc_select_input(4)`

ADC input is pin 4, which is on-board temperature sensor pin

`const float conversion_factor = 3.3f / (1 << 12)`

The resolution is 12, the highest IO voltage is 3.3V, and the ratio between them is the voltage corresponding to the unit value, so as to calculate the voltage read by the IO.

`adc_read()`

Perform ADC conversion, wait for the result, and then return the result.

`float reading = adc_read() * conversion_factor`

Calculate the converted voltage value.

`float temperature = 27 - (reading - 0.706) / 0.001721`

Converted to temperature value.

`printf("reslut: 0x%x, reading: %f V, temperature: %f \n", result, reading, temperature)`

Print the value of related variables, which can be viewed through serial port 0.

4. Experimental phenomenon

After the program is downloaded, if there is an external USB to TTL module, we can view the data through the serial debugging assistant, as shown in the following way.

```
reslut: 0x372, reading: 0.709790 V, temperature: 24.797762

[2021-03-02 10:46:19.810]# RECV ASCII>
reslut: 0x371, reading: 0.709790 V, temperature: 24.797762

[2021-03-02 10:46:20.313]# RECV ASCII>
reslut: 0x371, reading: 0.709790 V, temperature: 24.797762

[2021-03-02 10:46:20.828]# RECV ASCII>
reslut: 0x371, reading: 0.709790 V, temperature: 24.797762

[2021-03-02 10:46:21.339]# RECV ASCII>
reslut: 0x371, reading: 0.709790 V, temperature: 24.797762

[2021-03-02 10:46:21.825]# RECV ASCII>
reslut: 0x371, reading: 0.709790 V, temperature: 24.797762
```