

3.7 Battery power display

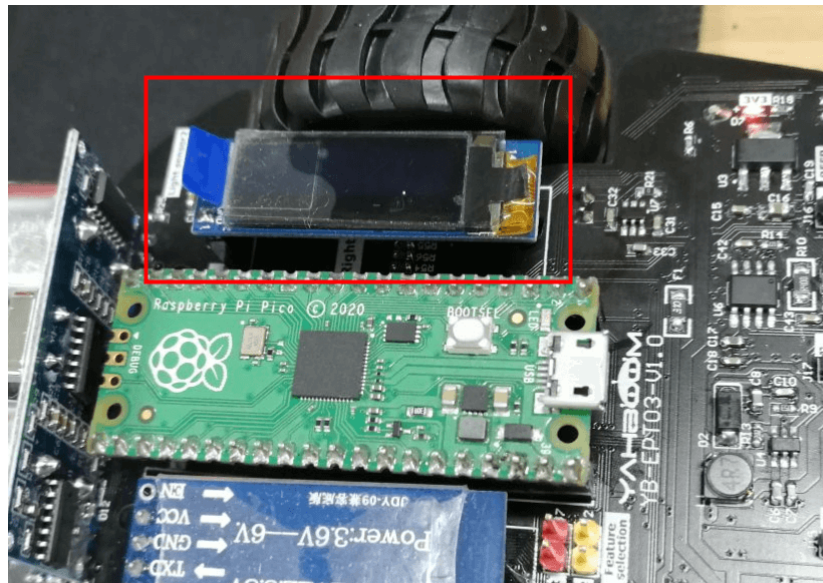
1. Learning Objectives

In this course, we will learn how to detect the current battery level of the car and display it through OLED.

2. About Hardware

We need to use detection circuit and OLED on Pico robot.

Note: OLED display must be plugged in before running the routine in this section, otherwise the program will report an error.



The basic principle of the power detection hardware is to adjust the battery voltage to an appropriate value through a resistor and input it to the main control.

Raspberry Pi Pico detects the voltage through the ADC to determine the power.

3. About code

Code path: Code -> 1.basic course -> 7.Battery power display.py

```
from machine import Pin, I2C, ADC
from pico_car import SSD1306_I2C
import time

#initialization OLED
i2c=I2C(1, scl=Pin(15),sda=Pin(14), freq=100000)
oled = SSD1306_I2C (128, 32, i2c)
#initialization ADC
Quantity_of_electricity = machine.ADC(28)

while True:
    #Display power on OLED
    #Under 20000, there is no power at all
    oled.text ('Battery:', 0, 0)
    oled.text(str(Quantity_of_electricity.read_u16()), 65, 0)
    oled.show()
    oled.fill (0)
```



```
time.sleep(0.1)
```

```
from pico_car import SSD1306_I2C
```

Use SSD1306_I2C of pico_car.

```
import time
```

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

```
from machine import Pin, I2C, ADC
```

The machine library contains all the instructions that MicroPython needs to communicate with Pico and other MicroPython-compatible devices, extending the language of physical computing, using the Pin, I2C and ADC libraries here.

```
i2c=I2C(1, scl=Pin(15),sda=Pin(14), freq=100000)
```

Set the IIC 1 pin to SCL 15, SDA 14, and the frequency to 100000.

```
oled = SSD1306_I2C (128, 32, i2c)
```

Initialize the size of the OLED to 128*32, and pass in the IIC parameters set earlier.

```
Quantity_of_electricity = machine.ADC(28)
```

Initialize ADC port 28.

```
oled.text(str(Quantity_of_electricity.read_u16()), 65, 0)
```

Set the OLED to display the value of the battery power at the position of 65,0. We read the value of the ADC through Quantity_of_electricity.read_u16(), convert the value into a string through the str() function, and then display it by the OLED.

```
oled.show ()
```

Display the set OLED content.

```
oled.fill (0)
```

Clear the settings and prepare for the next display.

4. Experimental Phenomenon

After the code is downloaded, we can see that the OLED displays 'Battery: battery level', which will change with the voltage.

After testing, when the value is lower than 20000, which means battery at low battery state, and if it is less than 25000, which means battery at medium battery state. .

