

5. PWM pins

I. Learning objectives

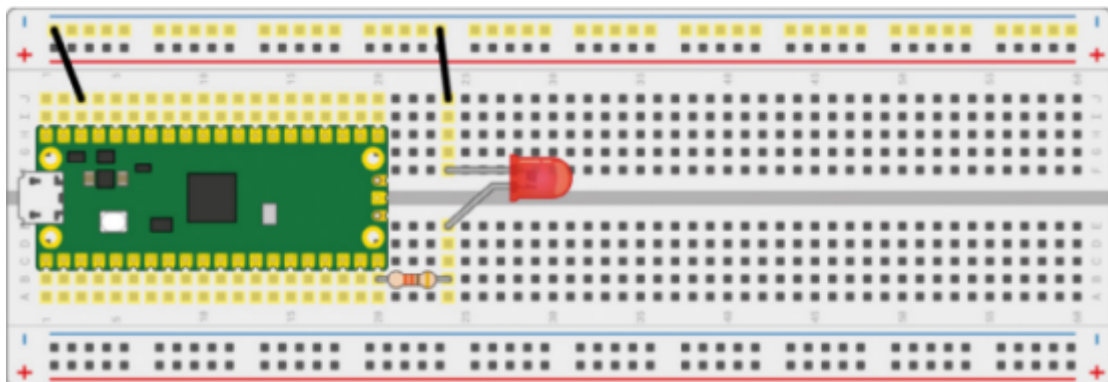
1. Learn to control LED lights with the Raspberry Pico 2 motherboard.
2. Understand the PWM output of the Raspberry Pico 2.

II. Hardware construction

This course.

LED light*1

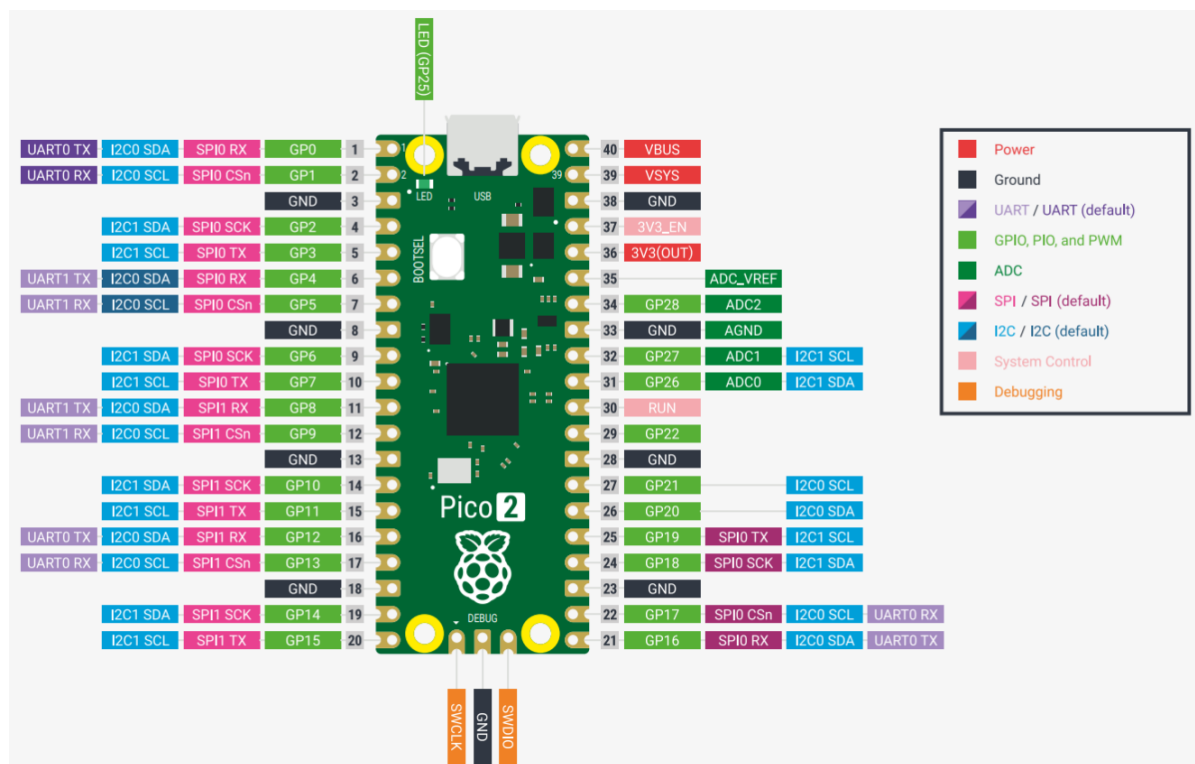
220Ω resistor*1



III. Program analysis

Each GPIO pin on the PICO 2 is capable of pulse width modulation, but the microcontroller's pulse width modulation block consists of eight slices, each with two outputs.

You will know if you accidentally use the same PWM output twice, because every time you change the PWM value on a pin, it will also affect the conflicting pin. If this happens, look at the pinout diagram and your circuit in the figure and find a PWM output that is not yet used.



GPIO	PWM
0	PWM0 A
1	PWM0 B
2	PWM1 A
3	PWM1 B
4	PWM2 A
5	PWM2 B
...	...
12	PWM6 A
13	PWM6 B
14	PWM7 A
15	PWM7 B
16	PWM0 A
17	PWM0 B
18	PWM1 A
19	PWM1 B
...	...
28	PWM6 A
29	PWM6 B

```

#include "pico/stdlib.h"
#include "hardware/pwm.h"

int main() {

    gpio_set_function(15, GPIO_FUNC_PWM);
    uint slice_num = pwm_gpio_to_slice_num(15);
    pwm_set_wrap(slice_num, 255);
    pwm_set_chan_level(slice_num, PWM_CHAN_B, 0);
    // Set the PWM running
    pwm_set_enabled(slice_num, true);

    while(1)
    {
        for(int i = 0; i <= 255; i++)
        {
            pwm_set_chan_level(slice_num, PWM_CHAN_B, i);
            sleep_ms(50);
        }
        for(int i = 255; i >= 0; i--)
        {
            pwm_set_chan_level(slice_num, PWM_CHAN_B, i);
            sleep_ms(50);
        }
    }
}

```

#include "hardware/pwm.h"

This library contains the relevant functions used by PWM.

gpio_set_function(15, GPIO_FUNC_PWM);

Select IO15 function as PWM. The first parameter is the GPIO number, and the second parameter is which function to use from the table.

Reference:

```

enum gpio_function { GPIO_FUNC_XIP = 0, GPIO_FUNC_SPI = 1, GPIO_FUNC_UART = 2,
GPIO_FUNC_I2C = 3, GPIO_FUNC_PWM = 4,

GPIO_FUNC_SIO = 5, GPIO_FUNC_PIO0 = 6, GPIO_FUNC_PIO1 = 7, GPIO_FUNC_GPCK = 8,
GPIO_FUNC_USB = 9, GPIO_FUNC_NULL =
0xf }

```

pwm_gpio_to_slice_num(15);

Returns the PWM slice for GPIO15.

pwm_set_wrap(slice_num, 255);

Sets the highest value the counter will reach before wrapping back to 0 to 255. Also known as TOP.

`pwm_set_chan_level(slice_num, PWM_CHAN_B, 0);`

Set the PWM counter comparison value for channel B. The first parameter is the PWM slice number, the second parameter is the channel number, and the third parameter is the comparison value.

`pwm_set_enabled(slice_num, true);`

Enable/disable PWM. The first parameter is the PWM slice number, the second parameter true is enabled, and false is disabled.

Through the While loop, modify the PWM duty cycle value through the for loop.

4. Program creation and burning

To create a project, please refer to the previous tutorial on creating a project, which will not be repeated here.

After writing the program, you need to add the standard library to the build according to the program you wrote before compiling. Open the CMakeLists.txt in the project and check target_link_libraries(pwm pico_stdlib hardware_pwm). You can see that the pico_stdlib and hardware_pwm libraries are used in the previous program. If the function used adds other libraries, you need to add the corresponding libraries. If you are not sure which library to add, you can check the official pico-examples case downloaded earlier, which contains examples of various functions and corresponding CMakeLists.txt files for reference.

After the modification is completed, save and exit, and enter the build path of the project.

Enter the following command to compile

```
cmake .. -G "NMake Makefiles"
nmake
```

After the compilation is completed, files in formats such as .bin .hex .elf .uf2 can be generated in the build directory.

Drag u2f in the above file into the disk recognized by Pico 2 to burn it. (Note: When burning for the first time, it is an empty code. Pico 2 can directly recognize the disk when connected to USB. When there is an executable program in it, you need to press and hold the BOOTSEL button and then connect USB)

5. Experimental phenomenon

After the program is downloaded, we can see the process of the LED light slowly turning on from off.