PWM

1. Learning purpose
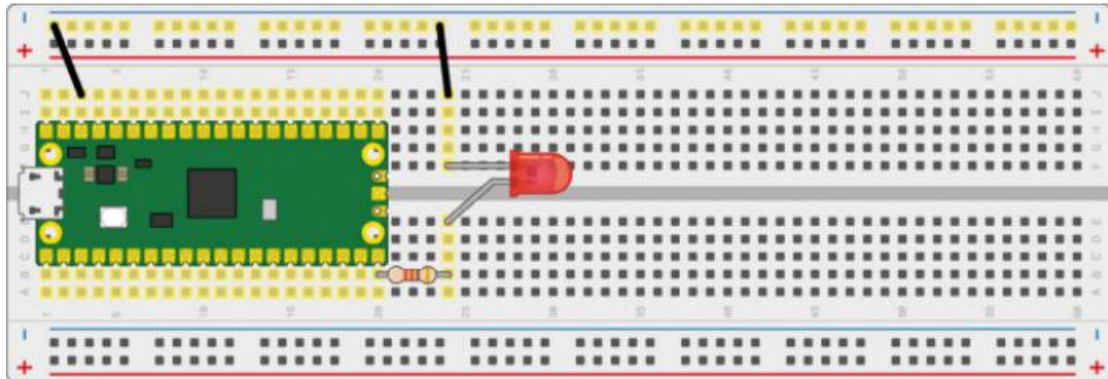
1.1 Learn how to use buzzer and Raspberry Pi Pico board.

1.2 Learn how to control LED by PWM.

2. Hardware construction

List:

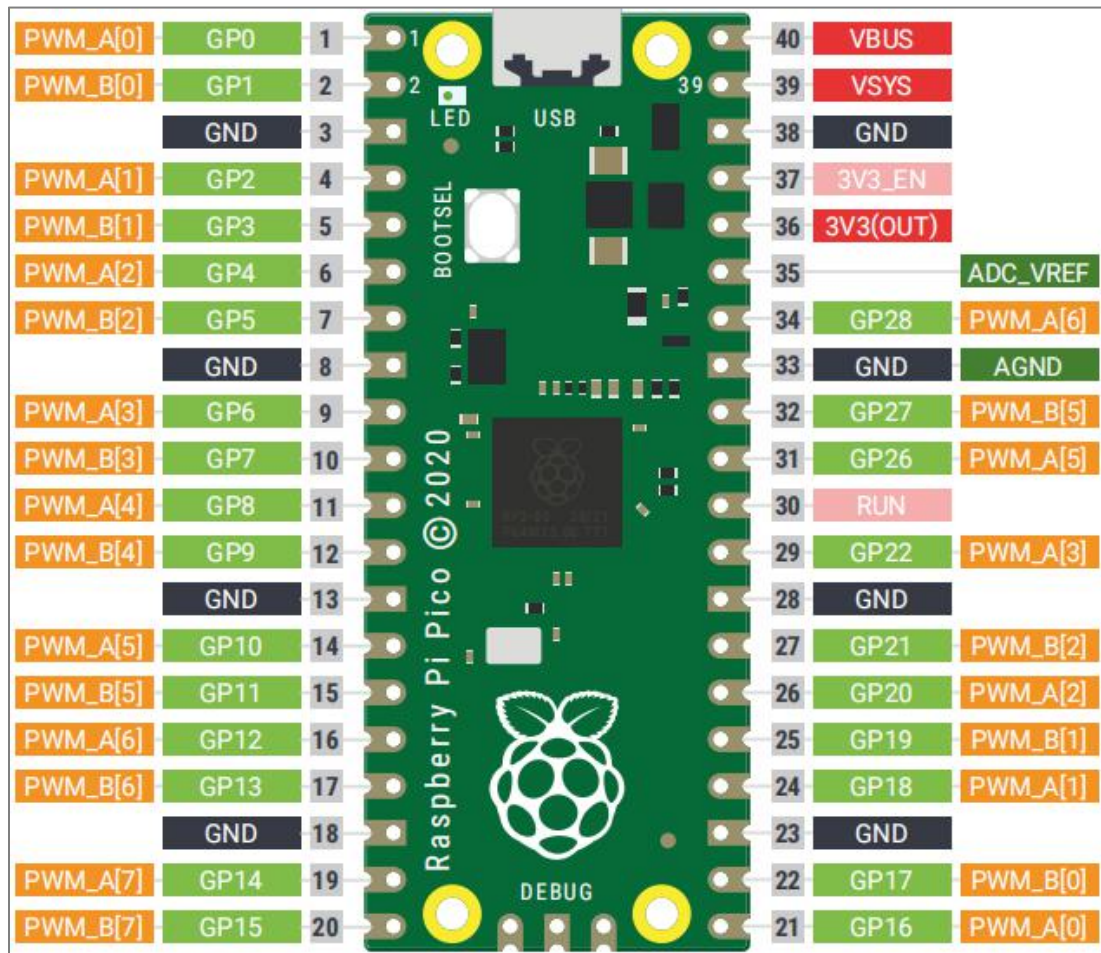LED light brightness*1

220Ωresistor *1

3. About code

Every GPIO pin on Pico can be pulse width modulated.

```c
#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 PWM related functions.

gpio_set_function(15, GPIO_FUNC_PWM)
Select IO15 function as PWM.

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)
Return the PWM slice of GPIO15.

pwm_set_wrap(slice_num, 255)
Set the highest value that the counter will reach to 255 before returning to 0 (TOP).

pwm_set_chan_level(slice_num, PWM_CHAN_B, 0)
Set the comparison value of the PWM counter 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.

Modify the PWM duty cycle value through the for loop.

4. Experimental phenomenon
After the program is downloaded, we can see that the LED lights gradually turn on from off.