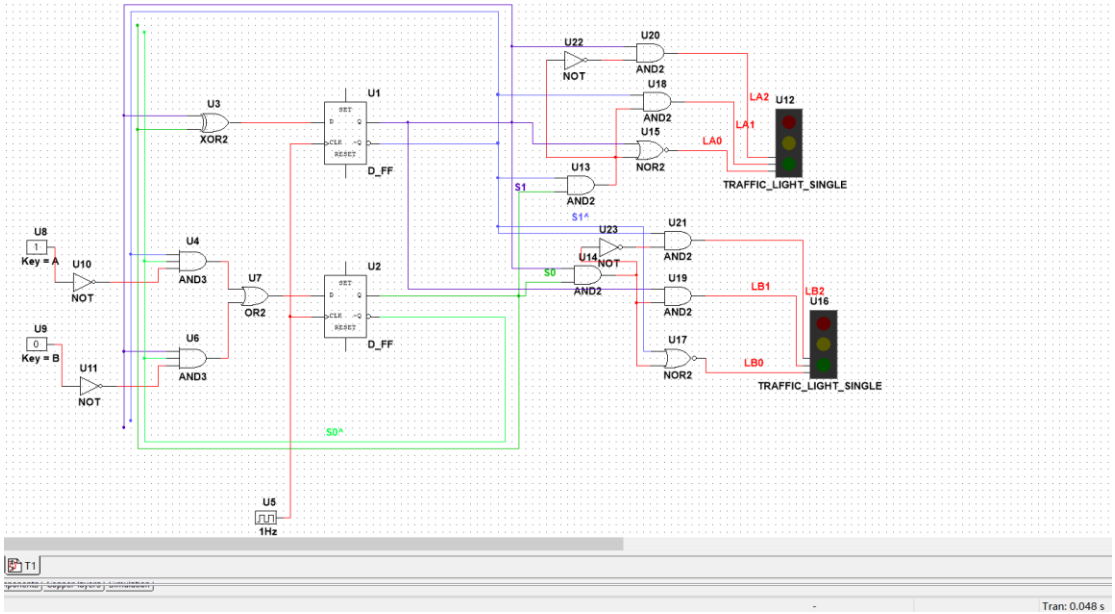
1.FSM I

Question 1



Let the frequency of the digital_clock be 1Hz.

So the clock signal will change in every 1 second.

And it will follow as a loop.

Change the key to control A and B.

Question 2

Use pulse_voltage to change the signal in 1ns when the time is 27s and 30s in one loop.

The loop is 30s.

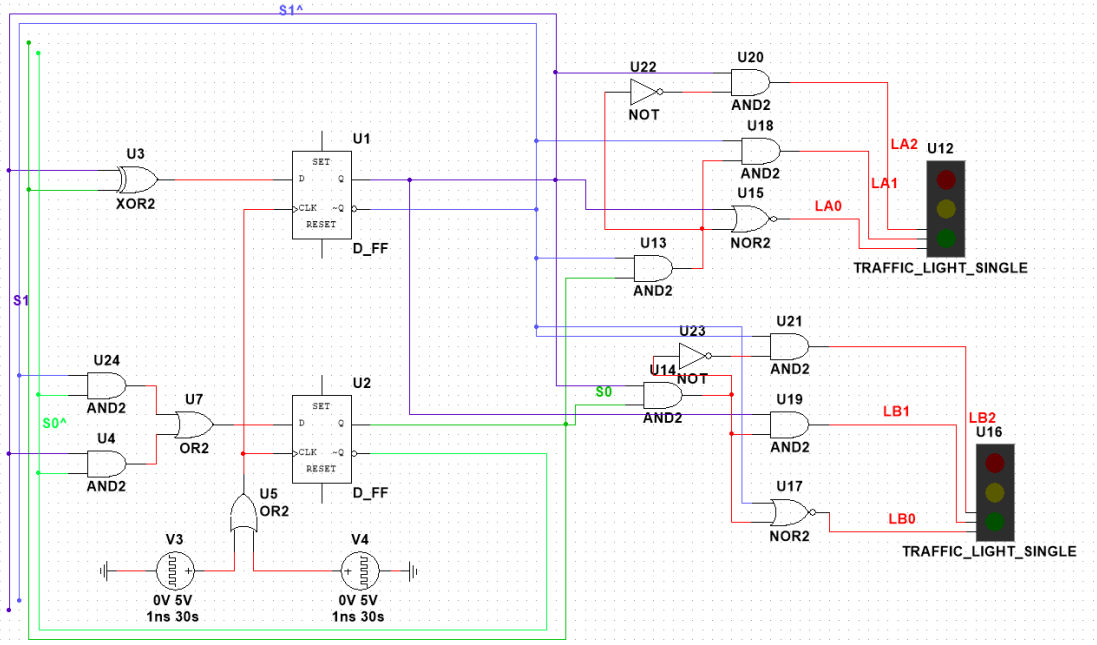
And the whole cycle of the two lights is 60s for one loop.

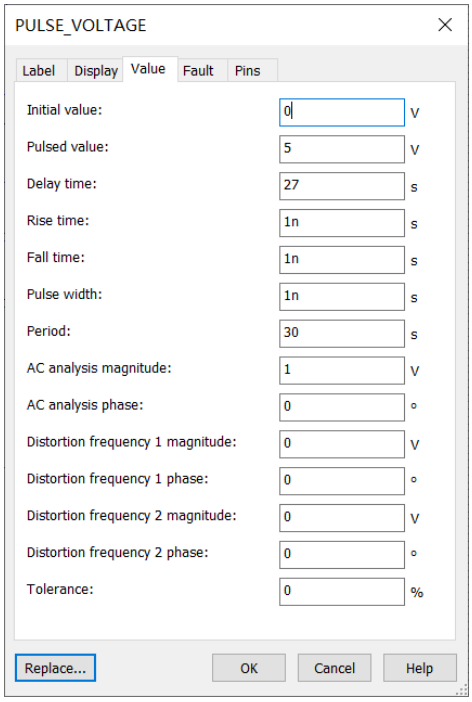
Let the traffic light above be L1, the traffic light below be L2.

In the first cycle,

The L1 start with green light on, turn to yellow at 27s, turn to red at 30s, turn to green at 60.s

The L2 start with red light on, turn to green at 30s, turn to yellow at 57s, turn to red at 60s.





The initial value is 0V, and Pulsed value is 5V.The pulse width be 1ns.

One PULSE_VOLTAGE delays 27s, the other on delays 30s.

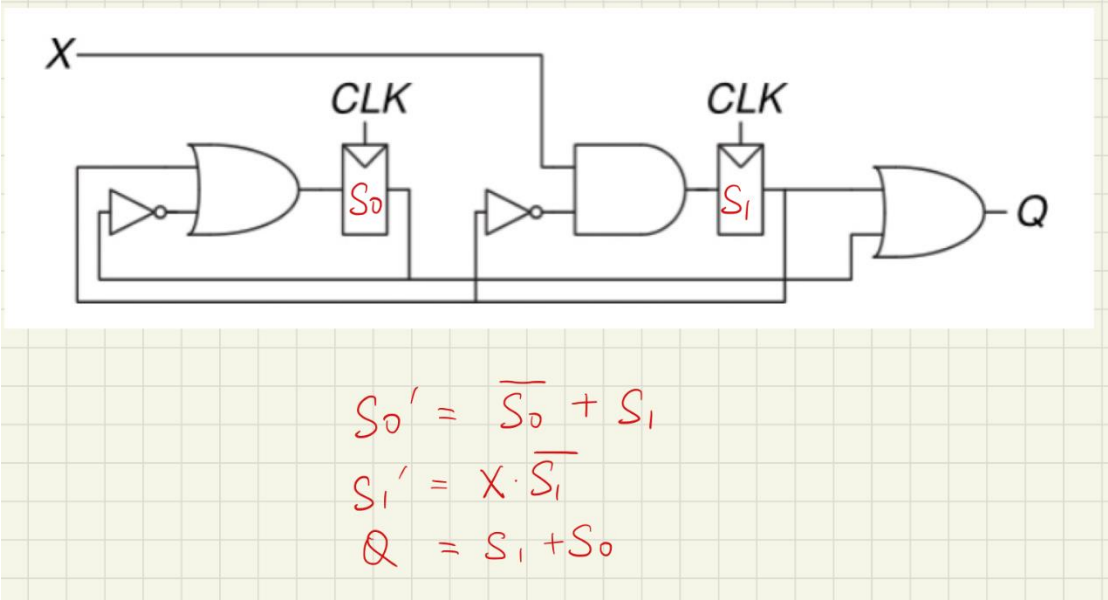
Let two PULSE_VOLTAGE use OR gate to connect together.

2.FSM II

The FSM is as followed.

Let the left D_FF be S0, the right D_FF be S1.

The S0,S1,Q's transform are as followed.



$$S_0' = \overline{S_0} + S_1$$

$$S_1' = X \cdot \overline{S_1}$$

$$Q = S_1 + S_0$$

The state transition is above

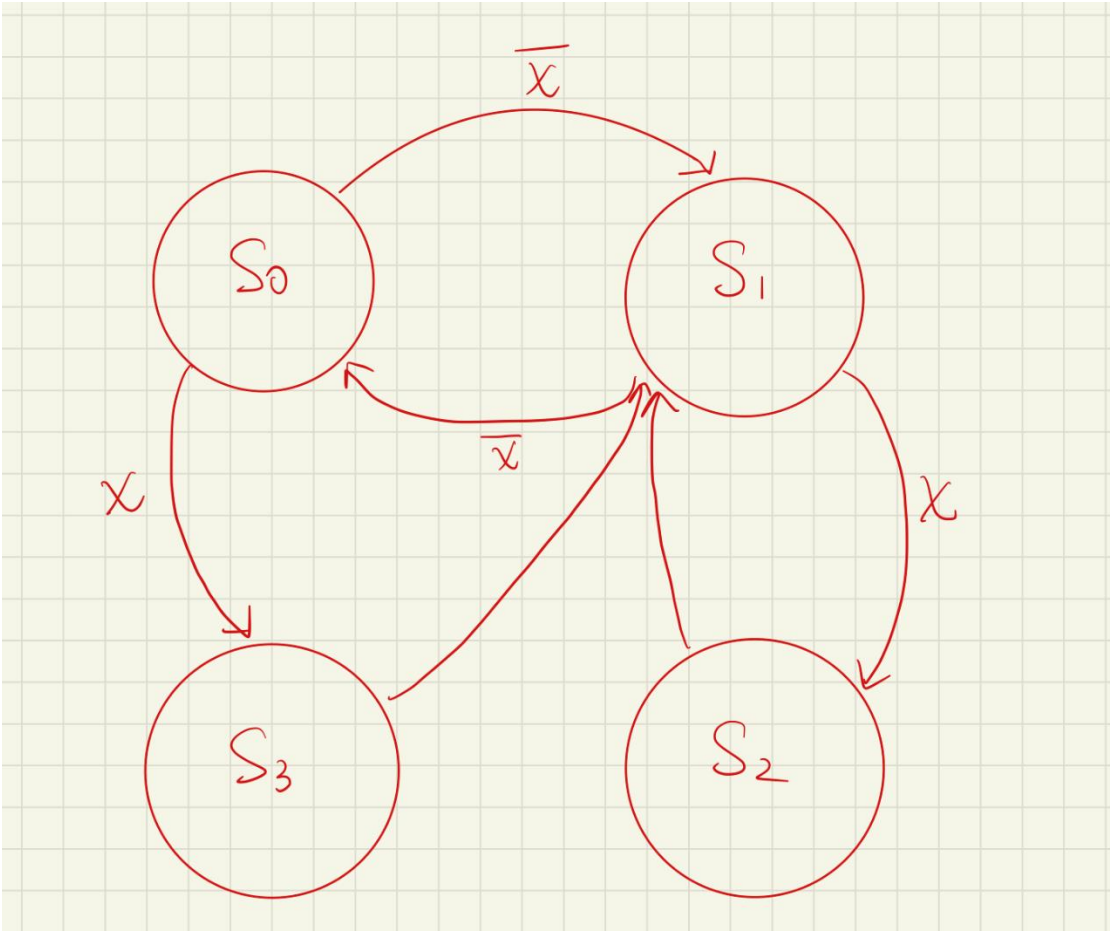
The state table and the output table(the output table fits the current state)

Current State		Input	Next State		Output
S_1	S_0	X	S_1'	S_0'	Q
0	0	0	0	1	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	0	1	1

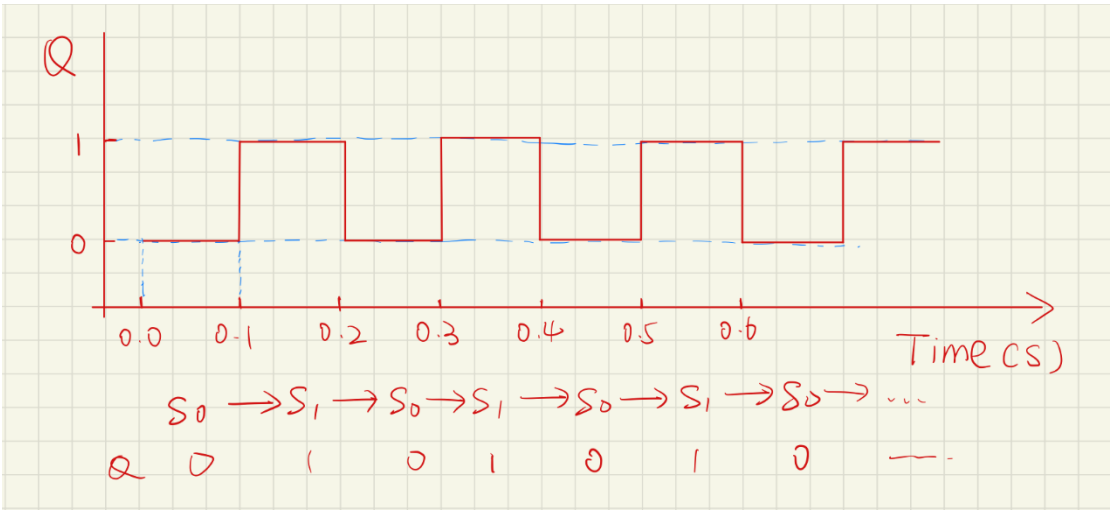
The output encoding

State	Encoding $S_{1:0}$
S_0	00
S_1	01
S_2	10
S_3	11

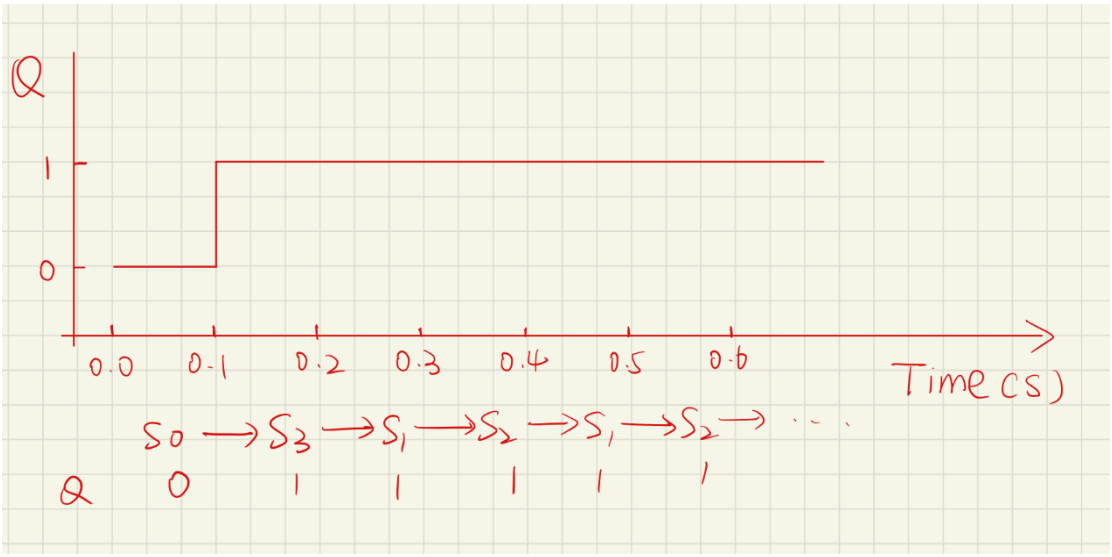
The tate transition diagram



The output waveform when X=0



The output waveform when X=1



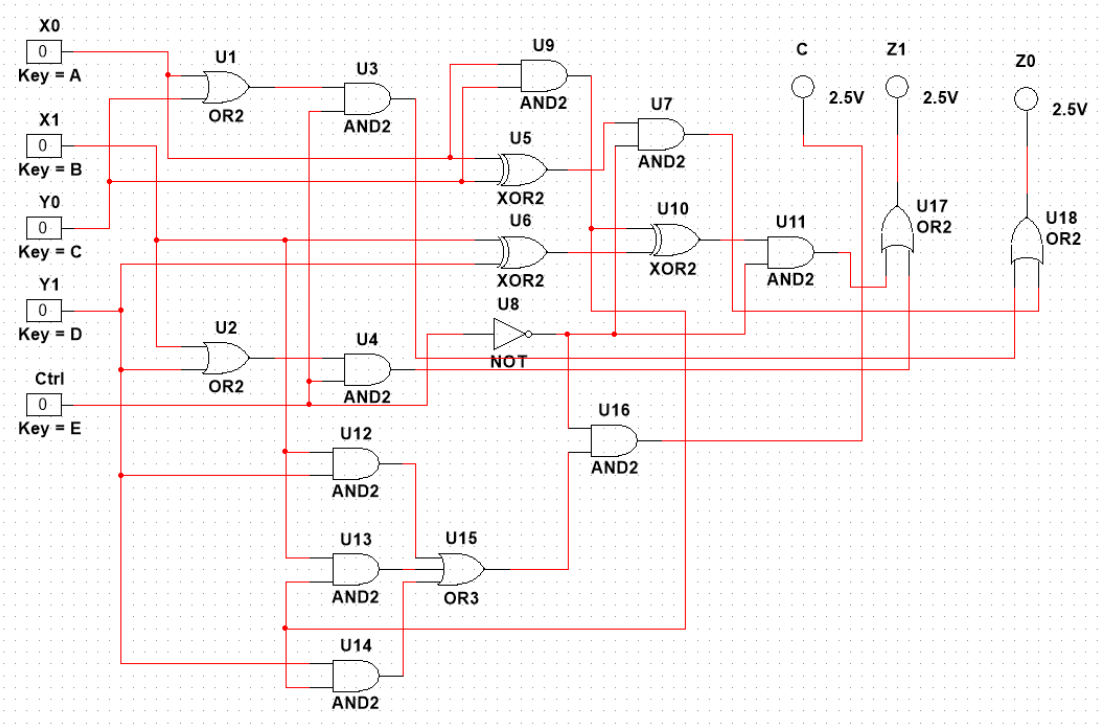
3.2 bits easy ALU

If the red(left) led is bright, then it means c=1,otherwise c=0

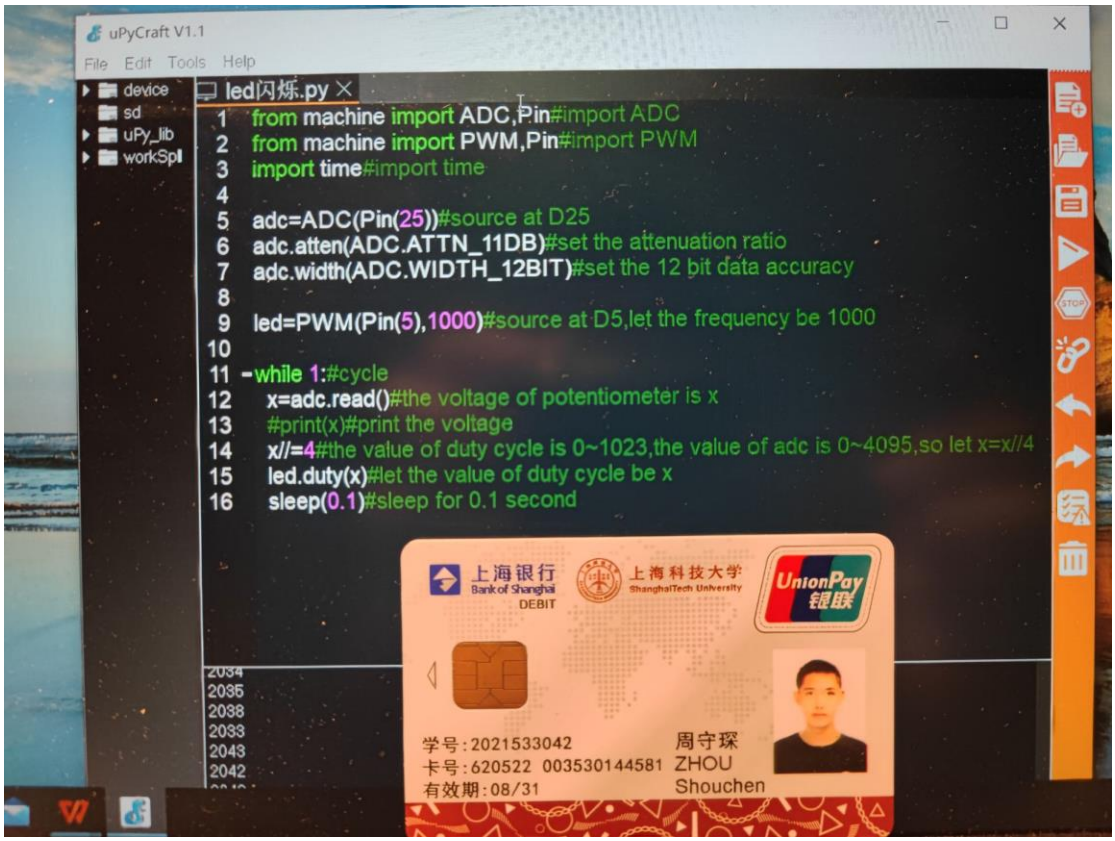
If the yellow(middle) led is bright, then it means z1=1,otherwise z1=0

If the green(right) led is bright, then it means z0=1,otherwise z0=0

The last digit of the ID number is 2,so use the or gate when ctrl=1.

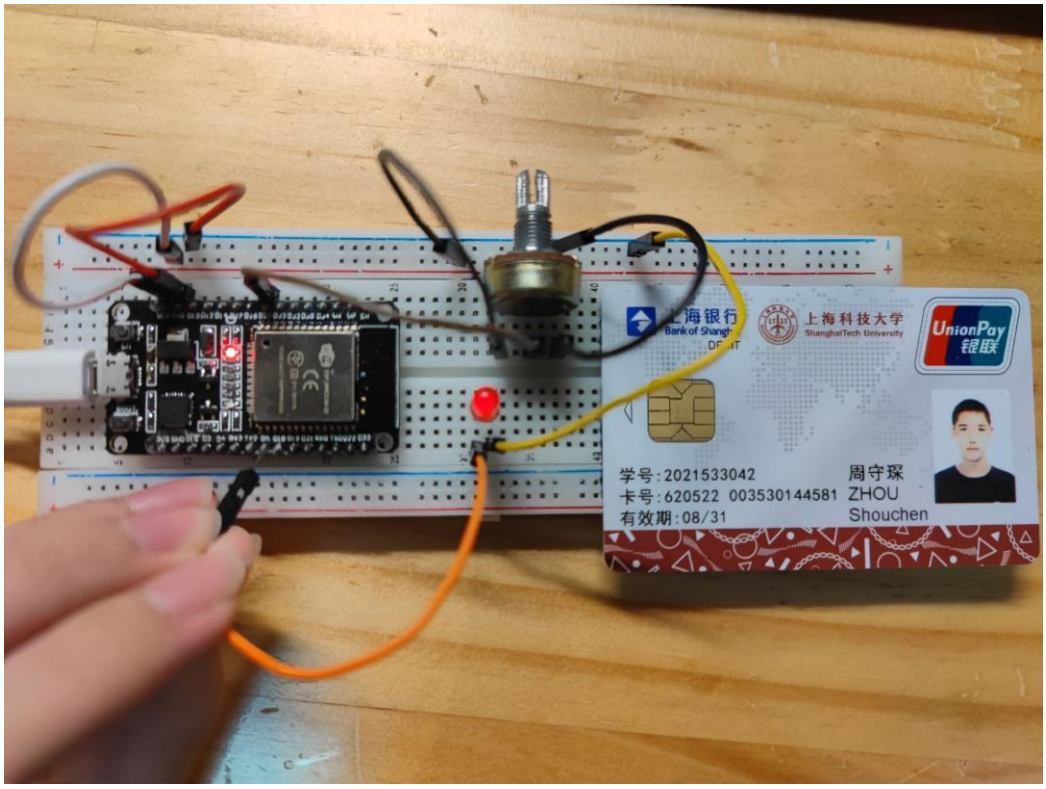


4.MCU development

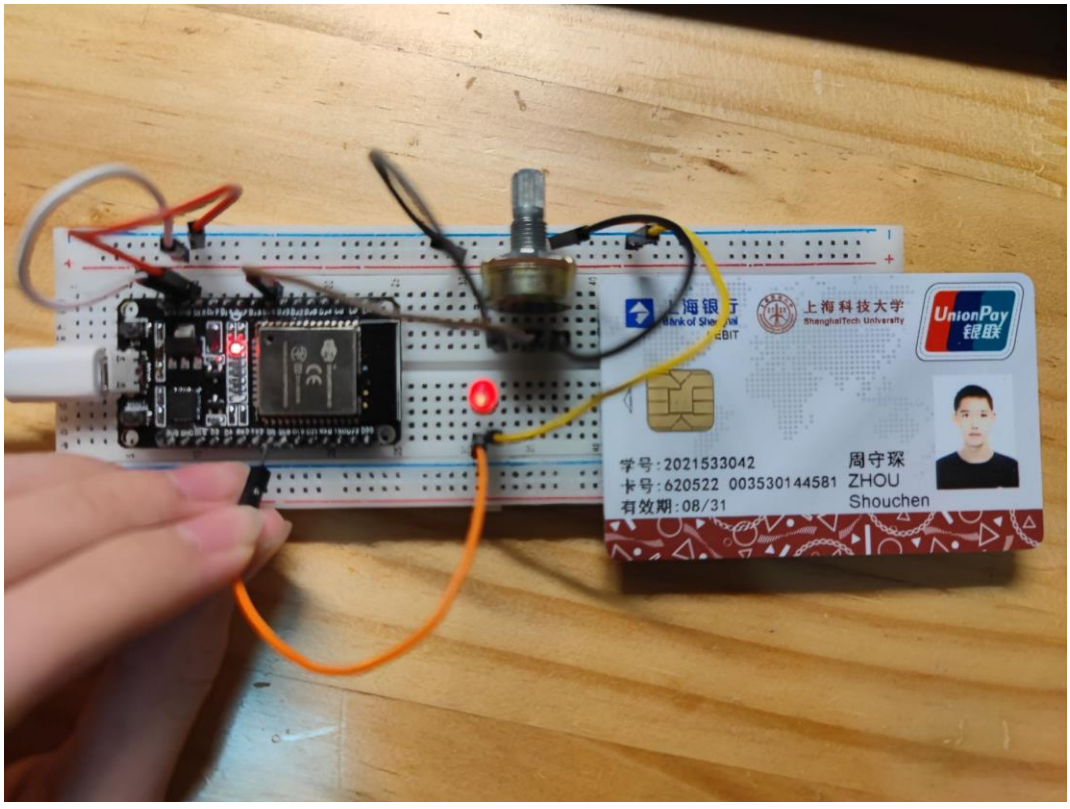



```
from machine import ADC,Pin#import ADC
from machine import PWM,Pin#import PWM
import time#import time

adc=ADC(Pin(25))#source at D25
adc.atten(ADC.ATTN_11DB)#set the attenuation ratio
adc.width(ADC.WIDTH_12BIT)#set the 12 bit data accuracy

led=PWM(Pin(5),1000)#source at D5,let the frequency be 1000

while 1:#cycle
    x=adc.read()#the voltage of potentiometer is x
    #print(x)#print the voltage
    x//=4#the value of duty cycle is 0~1023,the value of adc is 0~4095,so let x=x//4
    led.duty(x)#let the value of duty cycle be x
    sleep(0.1)#sleep for 0.1 second
```

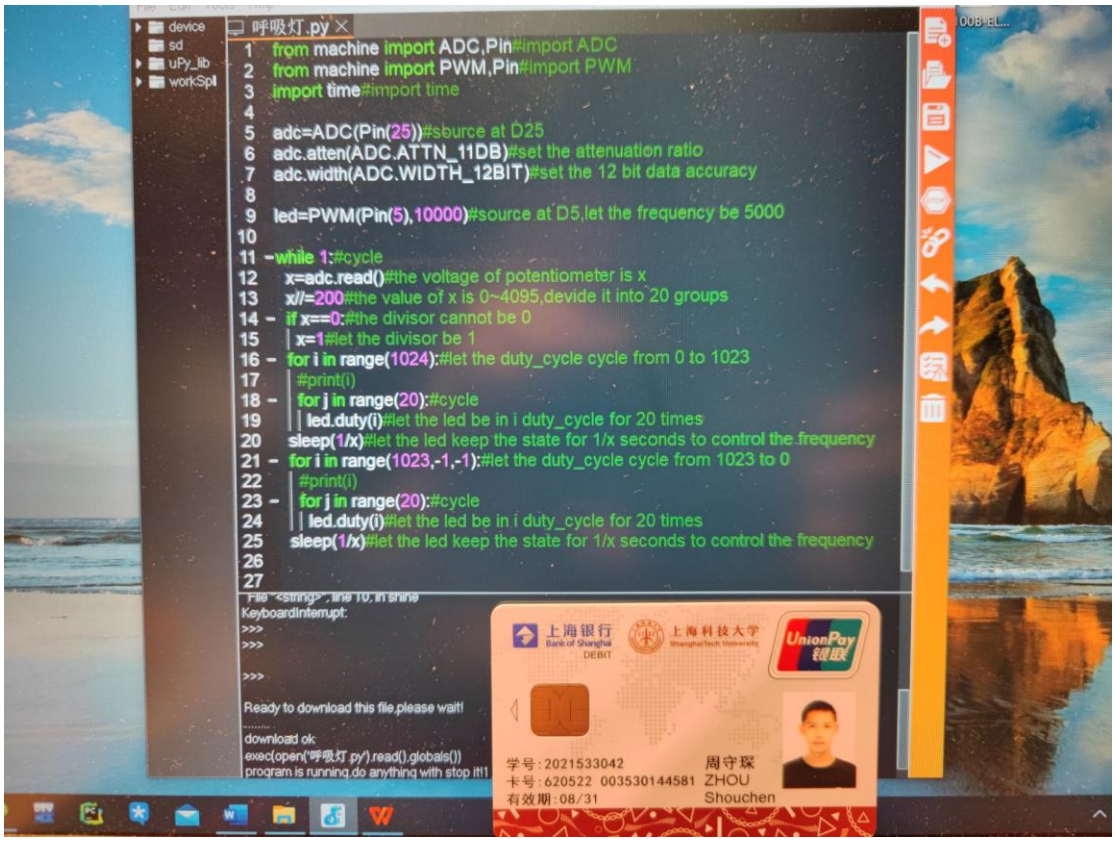
Adjust the potentiometer, let it get low voltage.
So brightness the of the led is low.



Adjust the potentiometer, let it get high voltage.
So brightness the of the led is high.



Bonus question



```
from machine import ADC,Pin#import ADC
from machine import PWM,Pin#import PWM
import time#import time

adc=ADC(Pin(25))#source at D25
adc.atten(ADC.ATTN_11DB)#set the attenuation ratio
adc.width(ADC.WIDTH_12BIT)#set the 12 bit data accuracy

led=PWM(Pin(5),10000)#source at D5,let the frequency be 5000

while 1:#cycle
    x=adc.read()#the voltage of potentiometer is x
    x//=200#the value of x is 0~4095,devide it into 20 groups
    if x==0:#the divisor cannot be 0
        x=1#let the divisor be 1
    for i in range(1024):#let the duty_cycle cycle from 0 to 1023
        #print(i)
        for j in range(20):#cycle
            led.duty(i)#let the led be in i duty_cycle for 20 times
    sleep(1/x)#let the led keep the state for 1/x seconds to control the frequency
    for i in range(1023,-1,-1):#let the duty_cycle cycle from 1023 to 0
        #print(i)
        for j in range(20):#cycle
            led.duty(i)#let the led be in i duty_cycle for 20 times
    sleep(1/x)#let the led keep the state for 1/x seconds to control the frequency
```