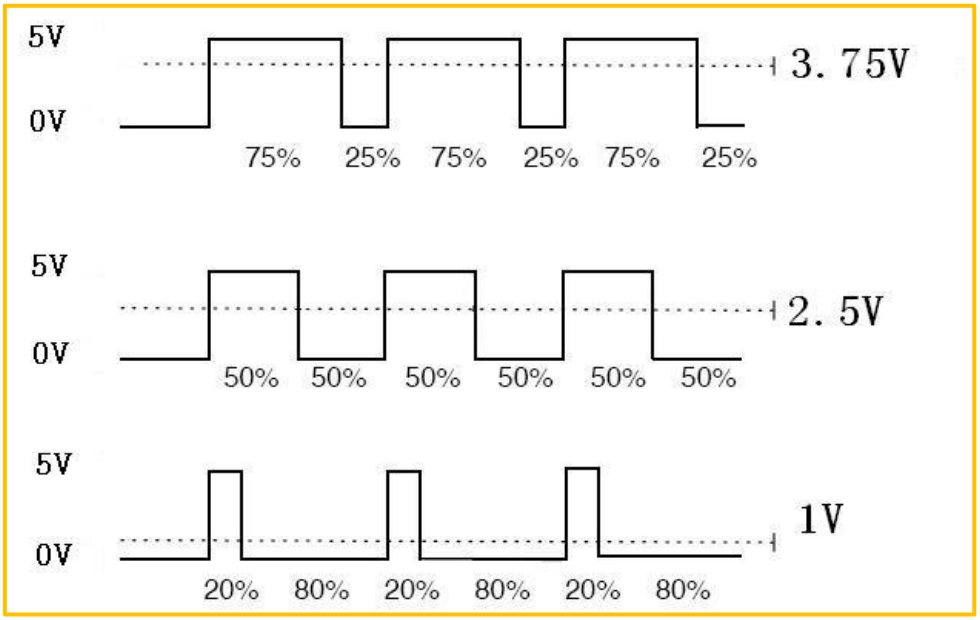
PWM Experiment

Introduction to PWM

Pulse Width Modulation is the full name of PWM. It is an analog signal level for digital encoding. Due to computers cannot output analog voltage but only 0 or 5v digital voltage value, we can apply the method of modulating duty cycle of square wave to encodes a specific analog signal level through high resolution counter. PWM signal is still digital, because in any given moment, amplitude of DC power supply supply is either full 5v (ON) or 0v (OFF). Voltage or current source is added to the analogue load through a (ON) or (OFF) repeat pulse sequence. That is when the DC power supply was added to the load, which is broken when power supply was disconnected. As long as there is enough bandwidth, any analog value can code via PWM. The output voltage value is calculated through on-off time. Output voltage = (turn-on time/pulse time) * maximum voltage value



1、Amplitude of pulse width variation (minimum/maximum)

2、Pulse period (The reciprocal of the number of pulse frequency in 1 second)

3、The voltage level (for example: 0 v to 5 v)

Arduino controller has six PWM interfaces, namely, digital interface 3, 5, 6, 9, 10, 11 (marked with silk-screen on the board ~), the PWM of Arduino don't have a lot of complicated operation, the output frequency is fixed at 50 Hz, application interface is analogWrite (pin, value).The first parameter is the pin, the second parameter value(0 ~ 256)

corresponds to the average output voltage (0 ~ 5 V). Say the second parameter is 128, then the duty cycle of PWM output is 50%, the average voltage is 2.5 V.

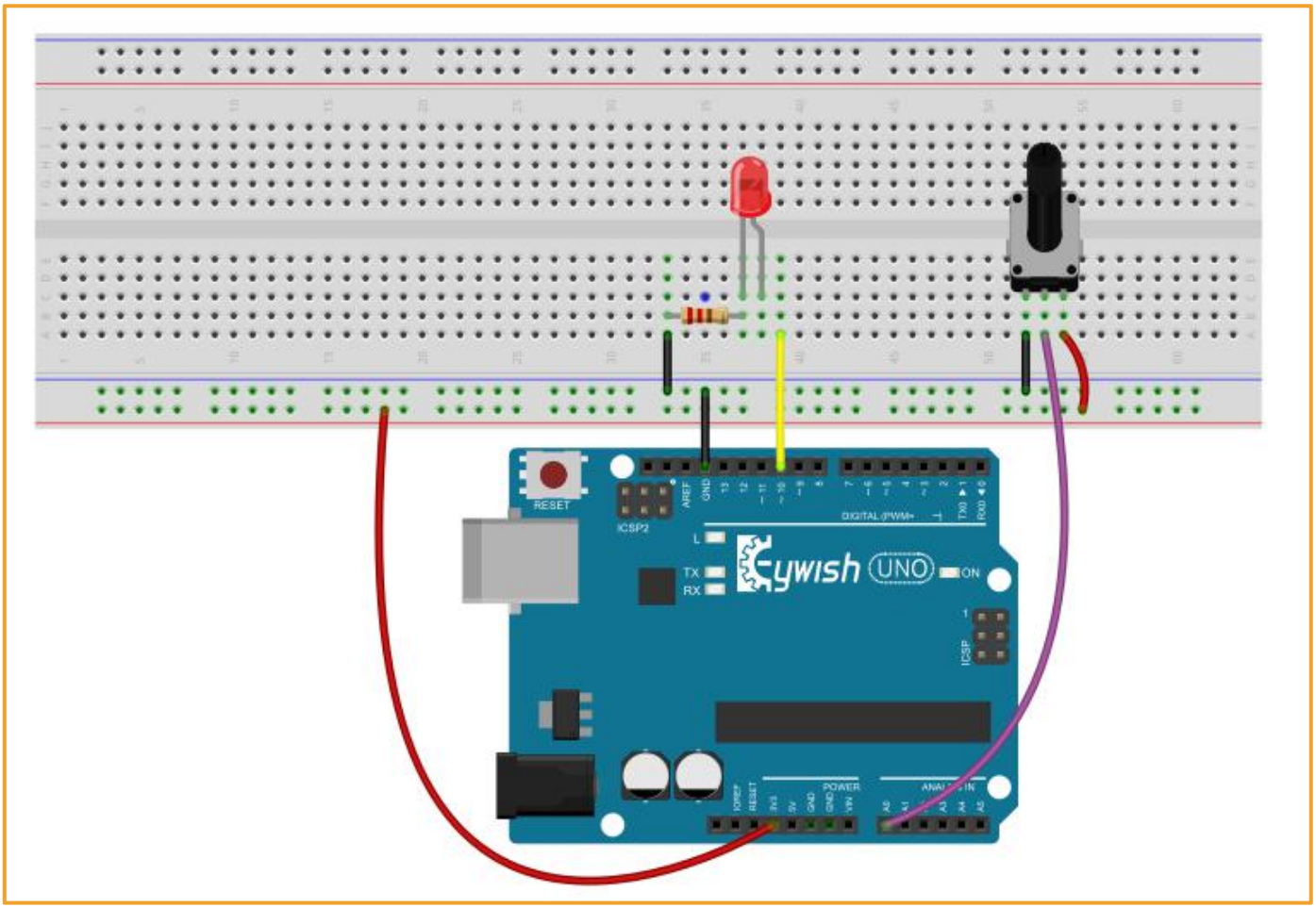
Experiment Purpose

We have done Button-controlling lights experiment before which is digital signal controlling digital interface, and we also realized the potentiometer experiment. Now we are to complete a experiment of light controlled by potentiometer.

Component List

- Keywish Arduino Uno Mainboard
- Breadboard
- USB cable
- 10k potentiometer * 1
- Red LED * 1
- 220Ω Resistor * 1
- Several jumper wires

Wiring of Circuit



Program Principle

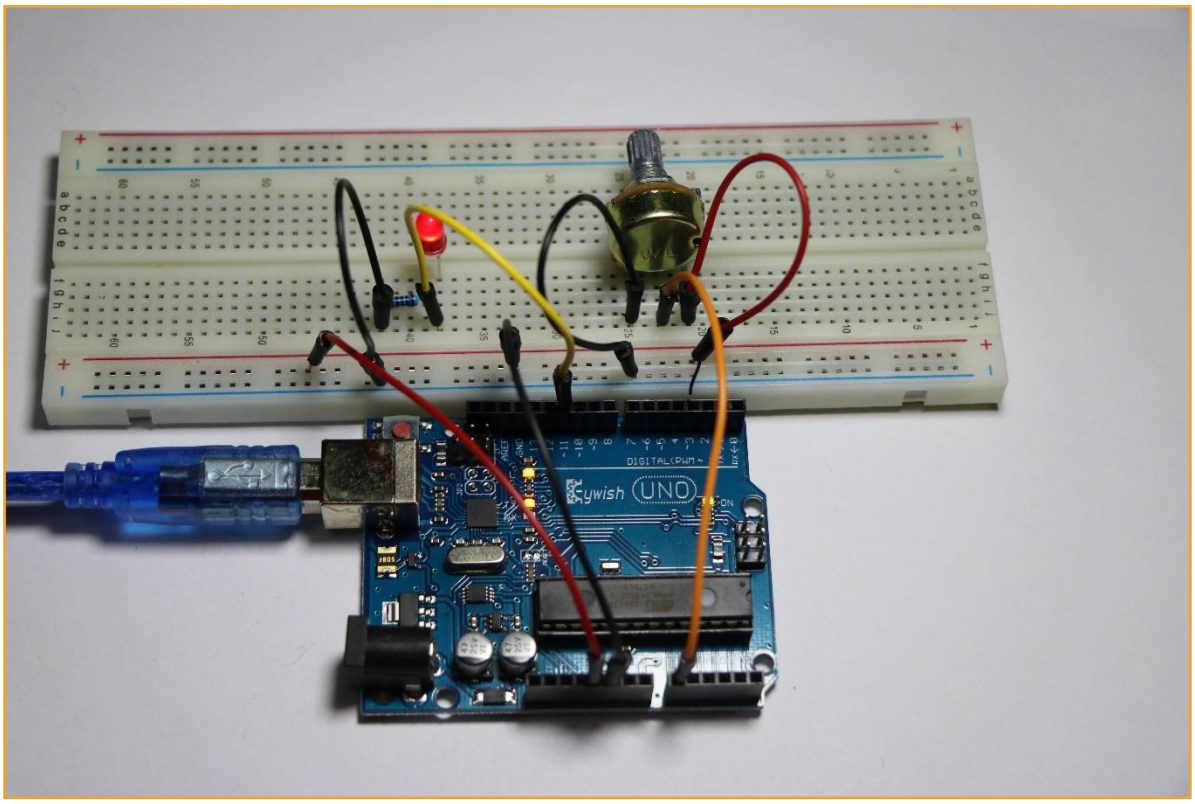
In the process of writing programs, we constantly read the collected value of AD interface through analogRead (pin) and convert it into voltage values, convert it into duty cycle by 256 scale after that, and analogWrite (pwm_pin, val) function to export the average voltage value. The change of the voltage can be perceived through the brightness of the LED.

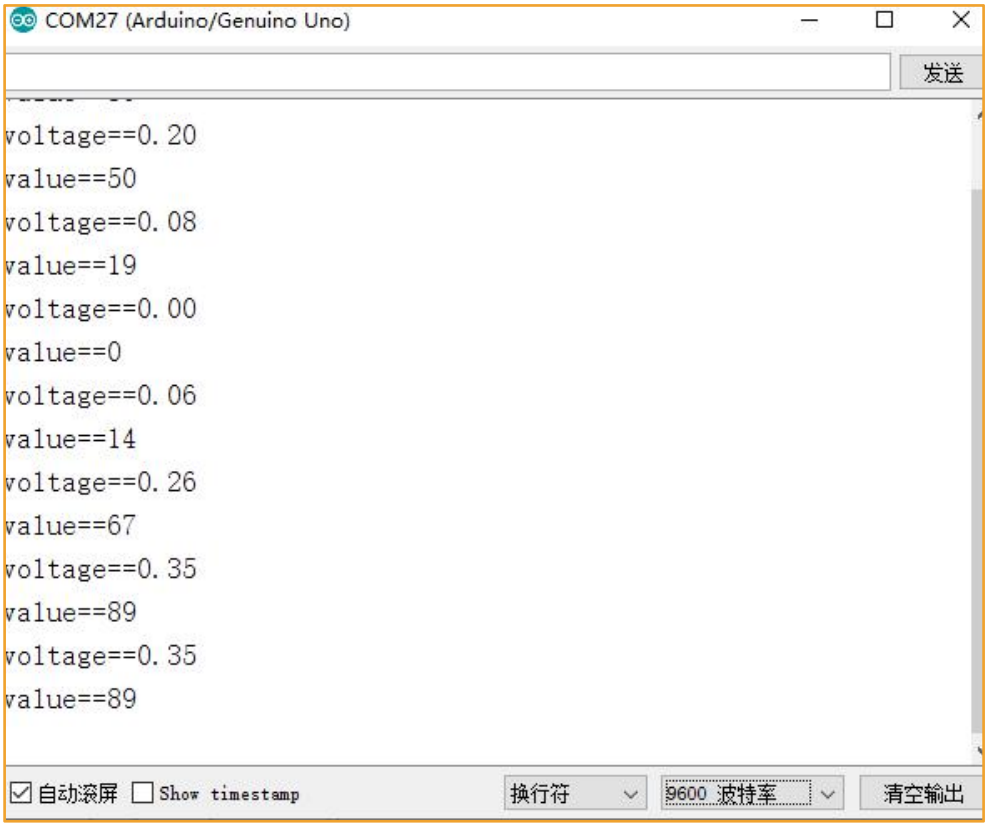
Code

```
int  ADPIN  = A0  ;
int  PWM LEDPIN = 10 ;
int  value = 0 ;
float voltage = 0.0 ;
void setup()
{
    pinMode(ADPIN,INPUT);   // define ADPIN input PWM LEDPIN output
    pinMode(PWM LEDPIN,OUTPUT);
    Serial.begin(115200);    //Serial Baud rate is 115200
}
void loop()
{
    value =  analogRead(ADPIN);       //read analog pin raw data
    voltage = ( ( float )value )/1023 ;
    value = (int)voltage * 256 ;      //covert to voltage to PWM duty cycle
    analogWrite(PWM LEDPIN,value);
    delay(1000);
}
```

Once the program is download, rotating the potentiometer knob, then we can not only numerical on the screen ,but also changes of the brightness of LED

Experiment Result





Mblock programming program

MBlock prepared PWM lighting adjustment program as shown in the figure below: