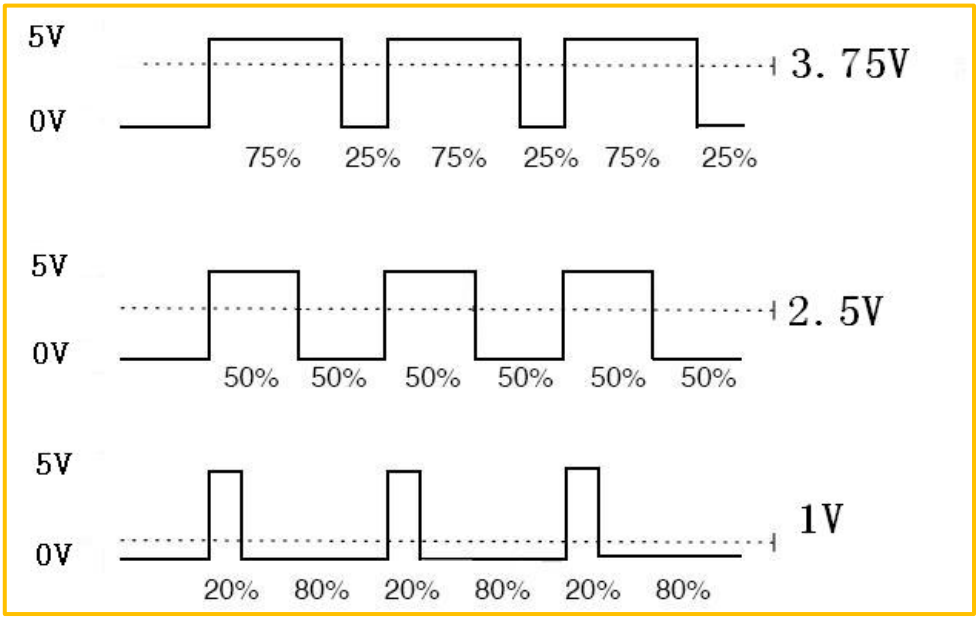
PWM Experiment

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.

potentiometer

Potentiometers are adjustable resistors that has following functions:

1. Used as voltage divider
2. Used as a rheostat
3. Used as current controller

Connection of three pins of potentiometer:

1. Potentiometer (or trimmer resistance, etc.) for conventional pins (just for example, a potentiometer with 3 pins), the resistance value at both ends is fixed, while the resistance value of the middle pin to any pin is variable;

2. It is equivalent to dividing the potentiometer into two resistors in series from the middle pin, and the total resistance value in series is fixed;

3. Therefore, if it is used as a variable shunt resistor, one end is connected to the input voltage, the middle end is connected to the output, and the rest is grounded;

4. If it is used as a variable resistor, one end is connected to the input voltage, and the middle end is connected to the output. The remaining end can be suspended or connected to the middle end.

The program principle

In the process of writing the program, we continuously read the collected value of the AD interface by analogRead and converted it into a voltage value, then converted it into a 256-level duty cycle, and then derived the average voltage value by using the analogWrite (pwm_pin, val) function. Voltage changes can be judged by the brightness of the LED.

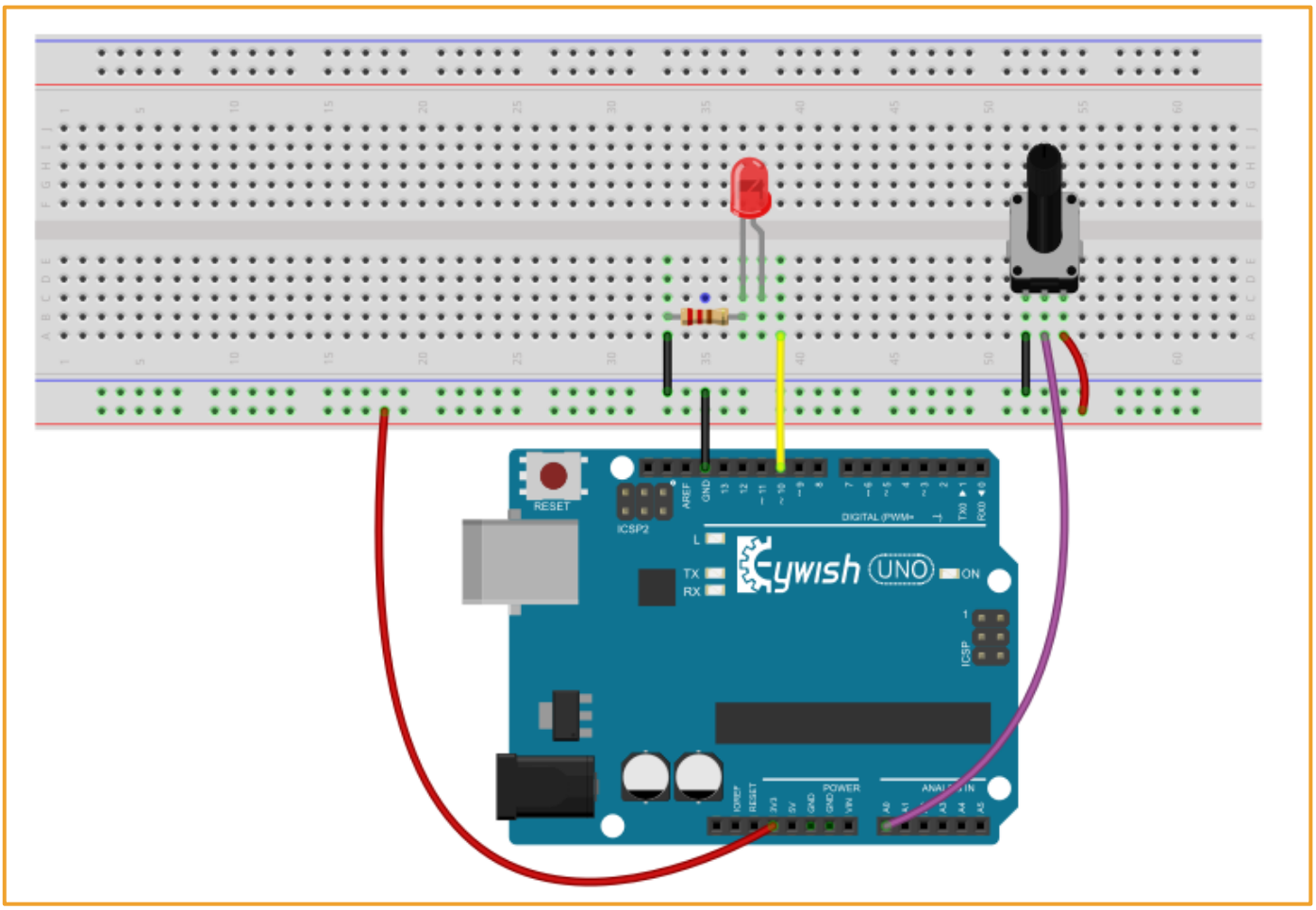
Experiment Purpose

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

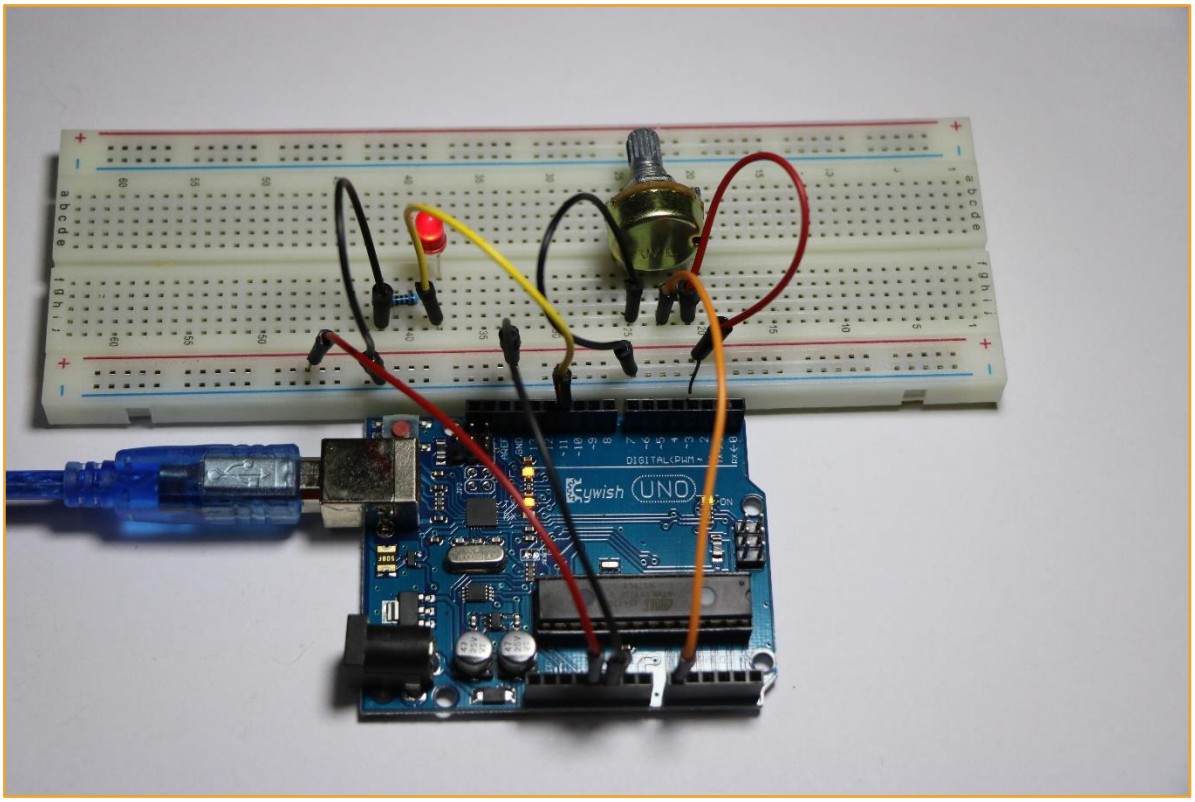


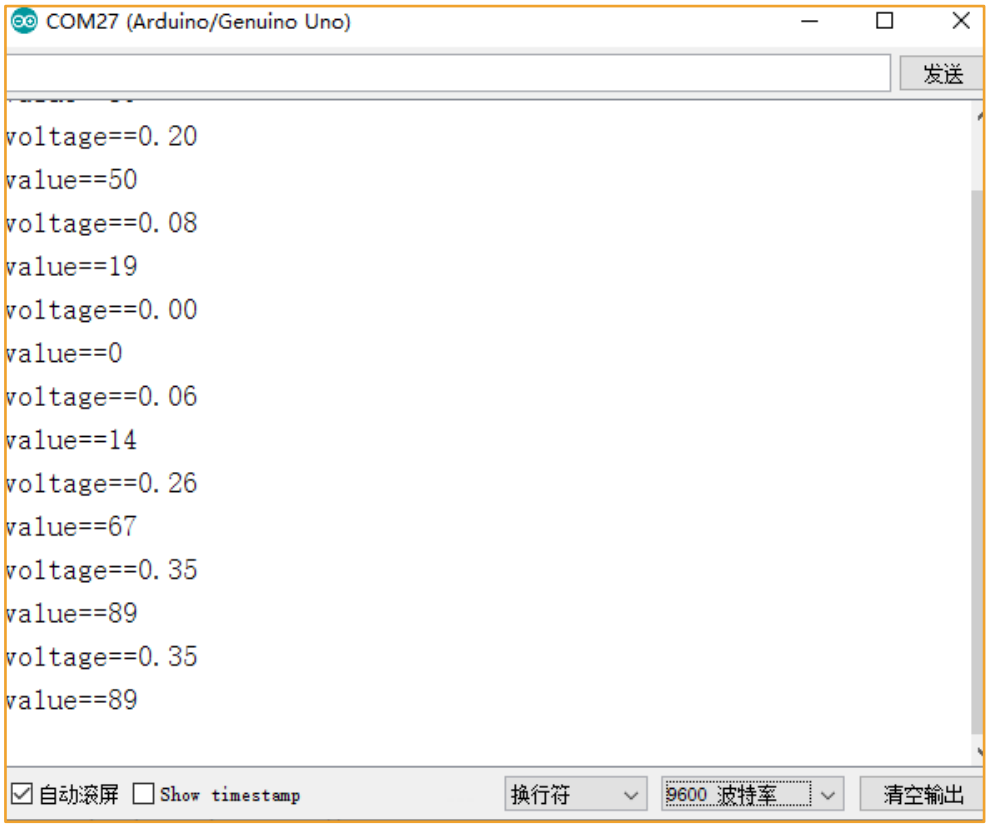
Code

```
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

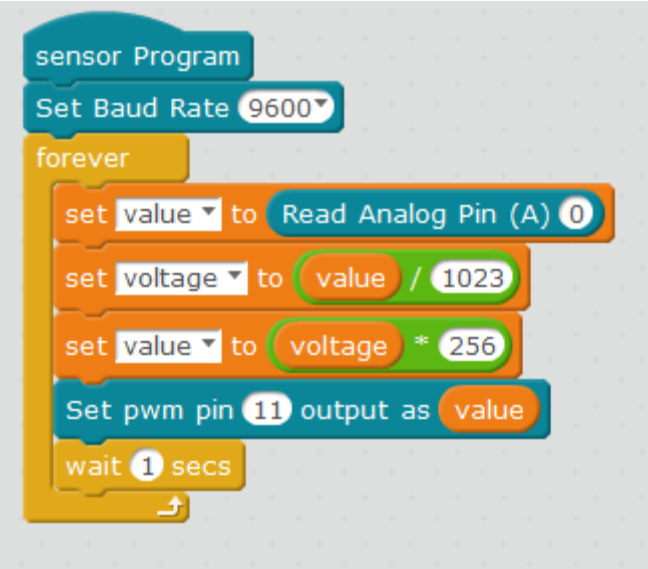




Through this experiment, we can observe the change of the brightness of the light as well as the value of the voltage and value of the serial monitor.

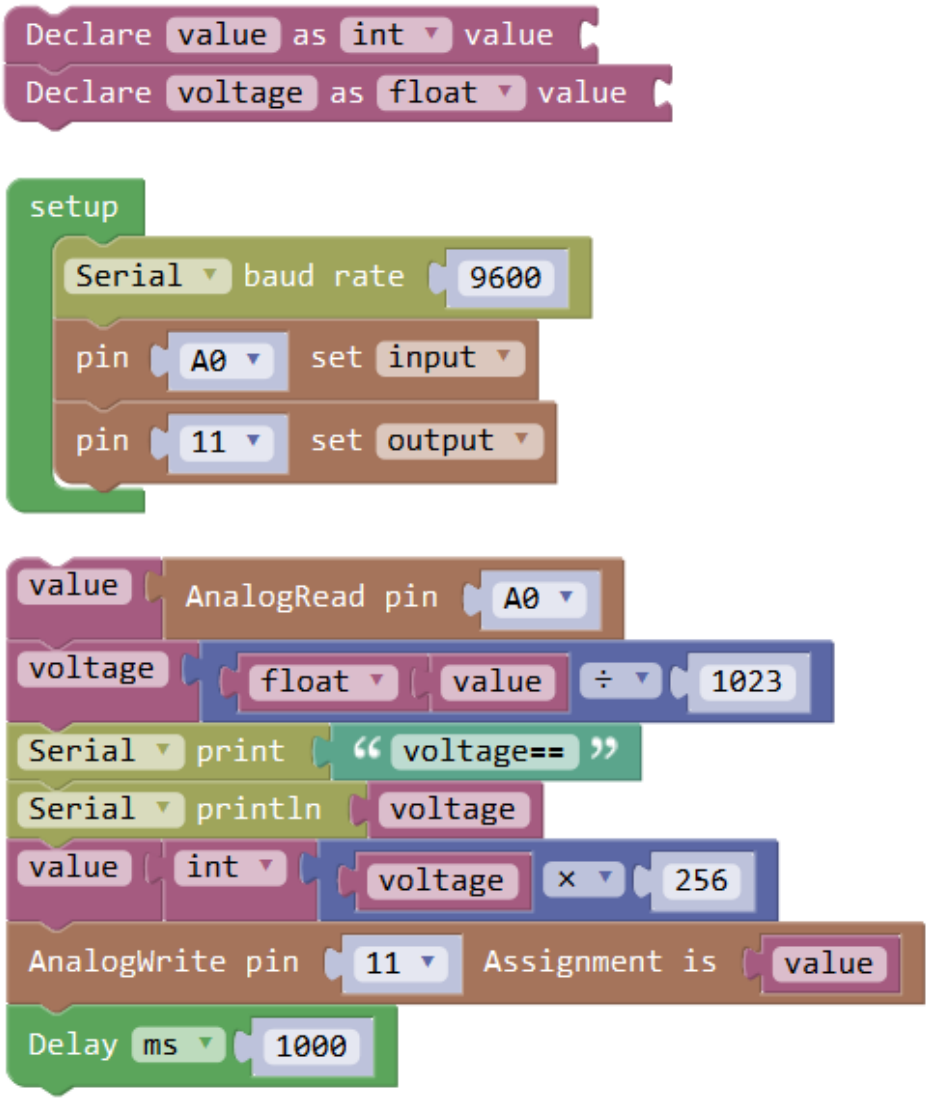
Mblock graphical programming program

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



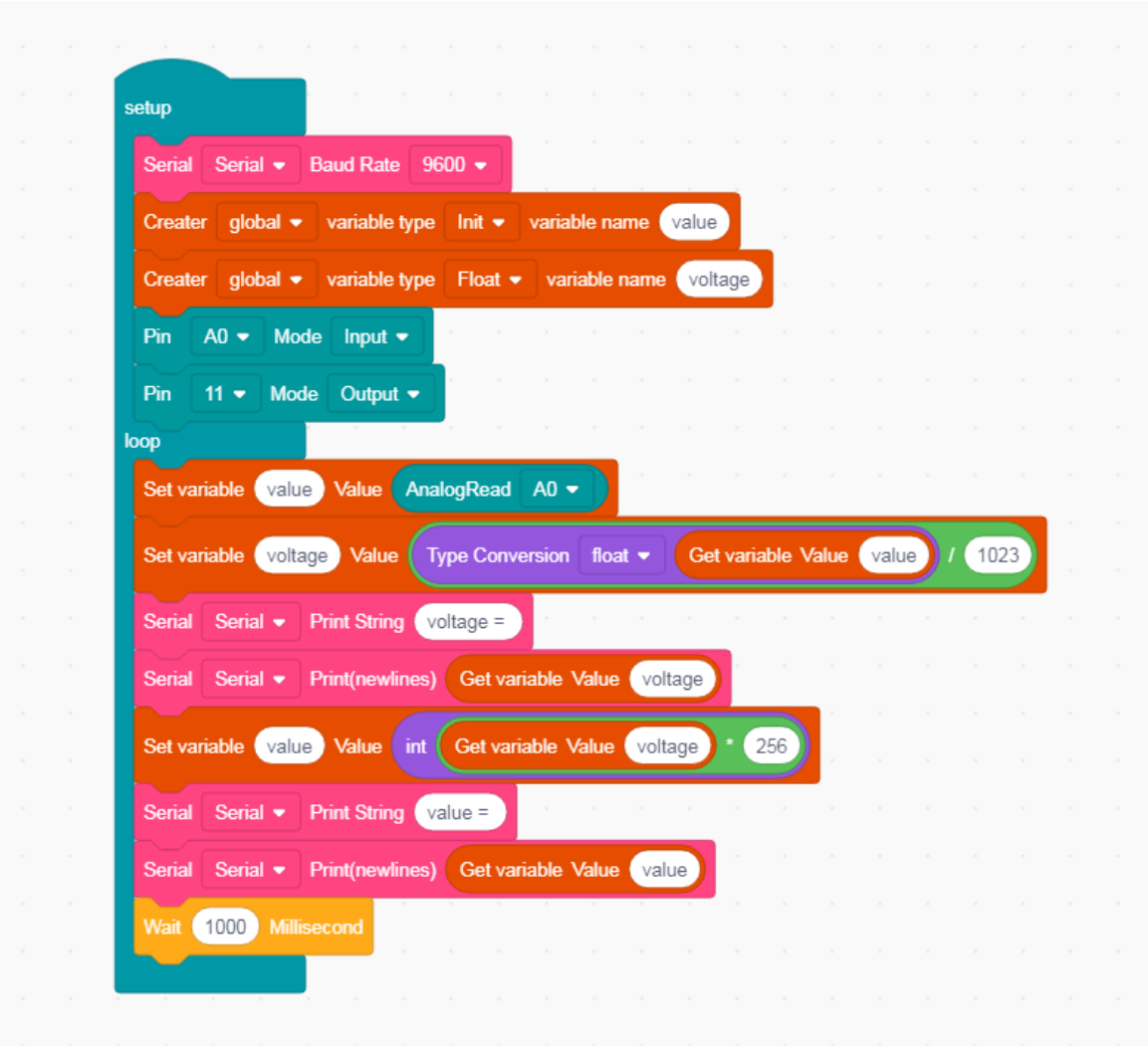
Mixly graphical programming program

Mixly programmed the program of PWM adjustment lamp brightness, as shown in the figure below:



MagicBlock graphical programming program

MagicBlock programmed PWM to adjust the brightness of LED light, as shown in the figure below:


setup
Serial Serial Baud Rate 9600
Creater global variable type Init variable name value
Creater global variable type Float variable name voltage
Pin A0 Mode Input
Pin 11 Mode Output
loop
Set variable value Value AnalogRead A0
Set variable voltage Value Type Conversion float Get variable Value value / 1023
Serial Serial Print String voltage =
Serial Serial Print(newlines) Get variable Value voltage
Set variable value Value int Get variable Value voltage * 256
Serial Serial Print String value =
Serial Serial Print(newlines) Get variable Value value
Wait 1000 Millisecond