AD analog-to-digital conversion experiment

Introduction to AD

We often hear A / D or D / A conversion in professional vocabulary, so what are A / D and D / A? AD conversion is analog-to-digital conversion, which can also be rectification. As the name suggests, it is to convert analog signals into digital signals.

The analog quantity can be electrical signals such as voltage and current, or non-electrical signals such as pressure, temperature, humidity, displacement and sound. But before the A / D conversion, the input signal input to the A / D converter must be converted into a voltage signal by various sensors through various physical quantities. After A / D conversion, the output digital signal is 10 bits.

DA conversion is digital-to-analog conversion, which is to convert a discrete digital quantity into an analog quantity with a changing connection. Corresponding to digital-to-analog conversion is analog-to-digital conversion, which is the inverse process of digital-to-analog conversion. Next, we will introduce digital-to-analog conversion from the aspects of converter classification, technical indicators, methods of analog-to-digital conversion, and parameters of analog-to-digital converters.

Arduino Mega 2560 has 16 analog interfaces, numbered from A0 to A15. In addition to the analog interface function, 16 interfaces can also be used as digital interfaces, numbered 54-69. After a brief understanding, let us start the following experiment. Potentiometer is a well-known typical analog value output element, it will be used to complete this experiment.

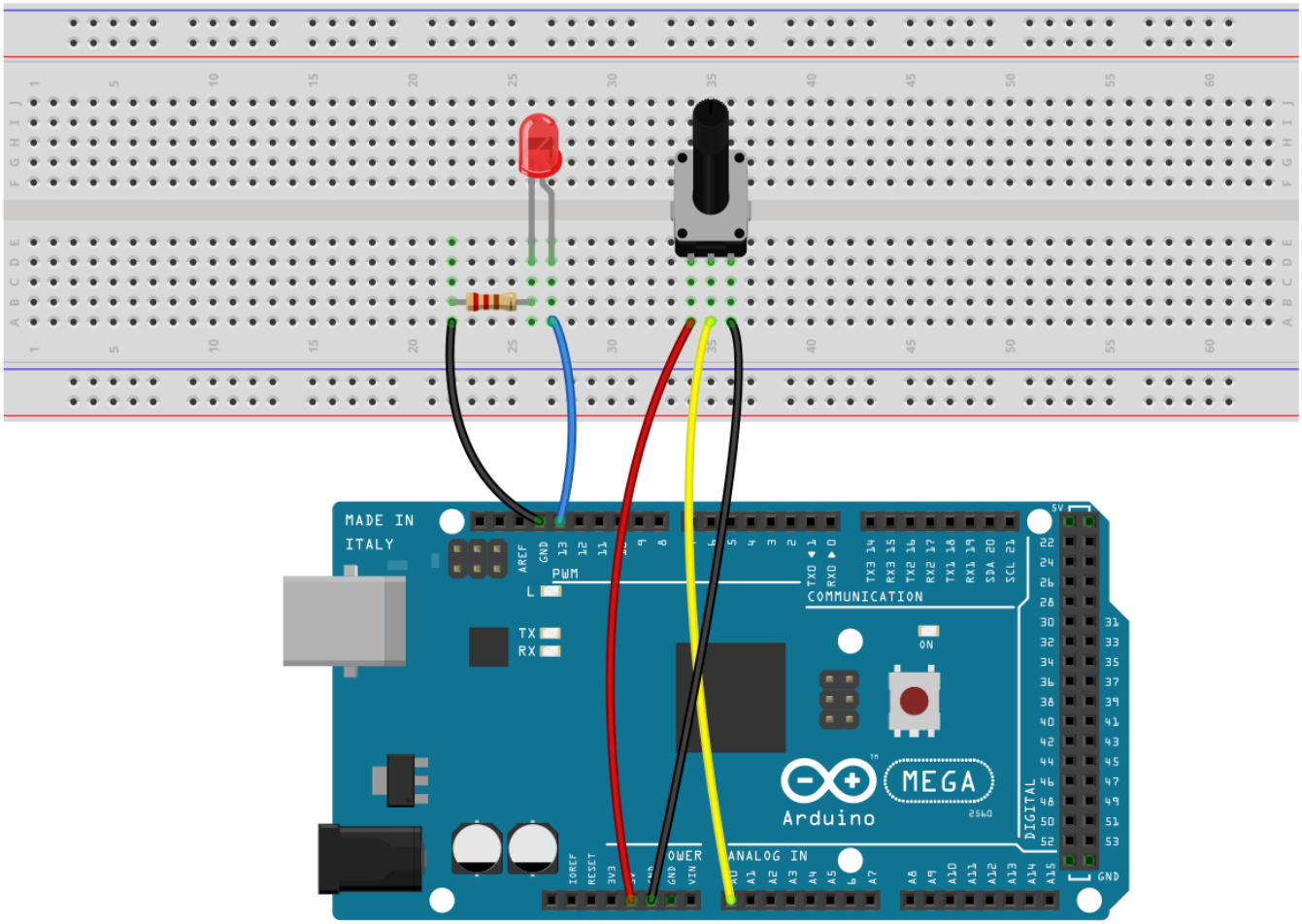
Component List

- Keywish Arduino Mega 2560 Mainboard
- Breadboard
- USB cable
- 10k Potentiometer * 1
- Several jumper wires

Experiment Purpose

In this experiment, we'll convert resistance value of potentiometer into analog value and read it out, then the value will be displayed on the screen, this is also a very application example for us to grasp in order to complete our experiments required in the future.

Wiring of Circuit



Experiment Principle

Through the function analogRead ();, statements can read out the value of analog interface. ATMEAG2560 takes A/D sampling by 10 bit, so the analog value range is 0 ~ $2^{10}-1$（0-1023）, the number is just the value of the AD, it needs to be converted into the actual voltage value, so we will use the following formula to calculate:

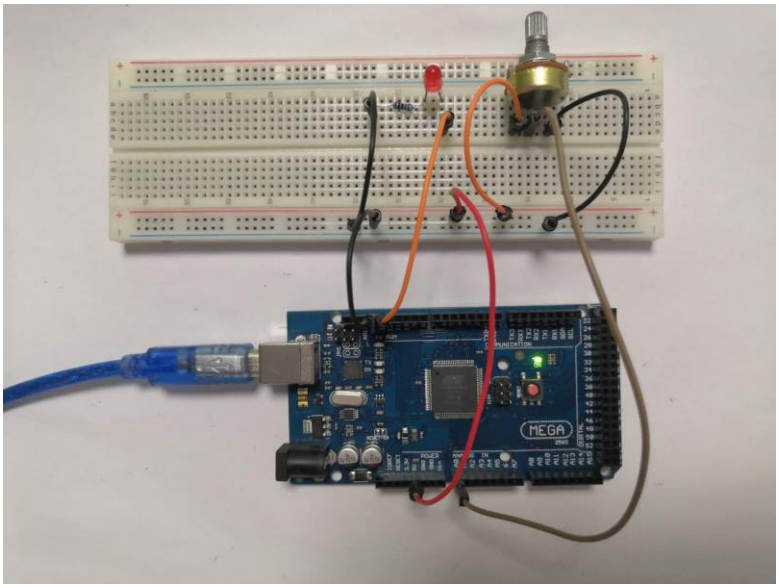
$$V_R = \frac{Value}{2^{10}-1} \times V_{DD}$$

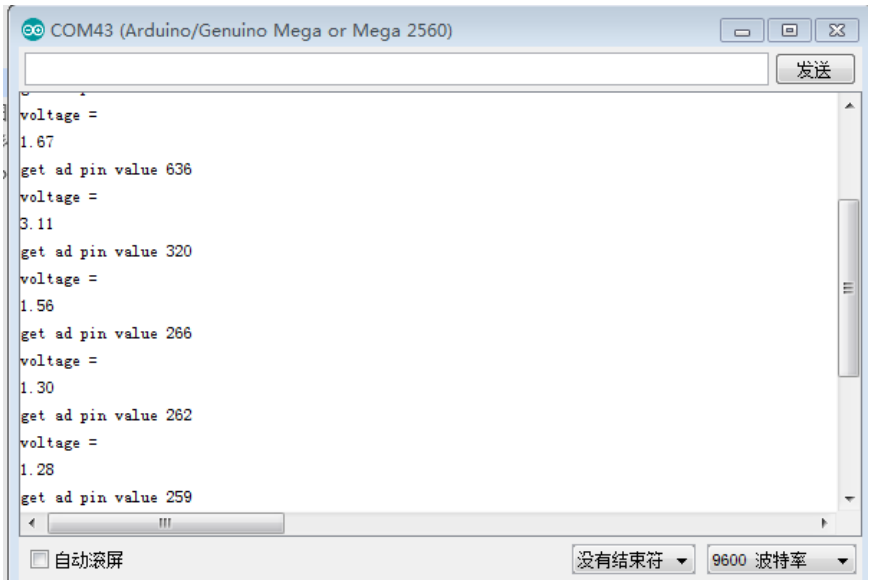
- V_R : real voltage
- Value : Sampled AD value
- V_{DD} : AD reference voltage value

Code

```
int  ADPIN  = A0 ;
int  LEDPIN = 13 ;
int  value = 0 ;
float voltage = 0.0 ;
void setup()
{
    pinMode(ADPIN,INPUT);   // define ADPIN input LEDPIN output
    pinMode(LEDPIN,OUTPUT);
    Serial.begin(115200);    //Serial Baud rate is 115200
}
void loop()
{
    digitalWrite(LEDPIN,HIGH);      // light on led
    value =  analogRead(ADPIN);     // read analog pin raw data
    voltage = ( ( float )value )/1023 ;
    voltage = voltage * 5 ;         // convert analog raw data to real
                                    voltage = //(analog/1023)*5
    Serial.print("get ad pin value ");   //printf Analog pin value
    Serial.print(value);
    Serial.println("\nvoltage =  ");
    Serial.println(voltage);
    delay(1000);
    digitalWrite(LEDPIN,LOW);             //turn off led
}
```

Experiment Result





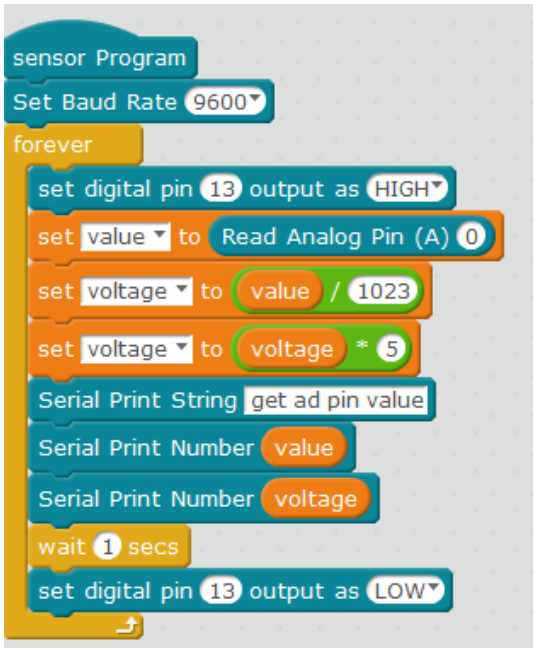
This experiment is done here. Now when you rotate the potentiometer knob , you will see numerical changes on the screen. This method of reading the analog value will always accompany us, it is also our common function, because most of sensors output analog value, we read the analog value and do corresponding algorithm processing, then it can be applied to the function that we need to implement.

Mblock programming program

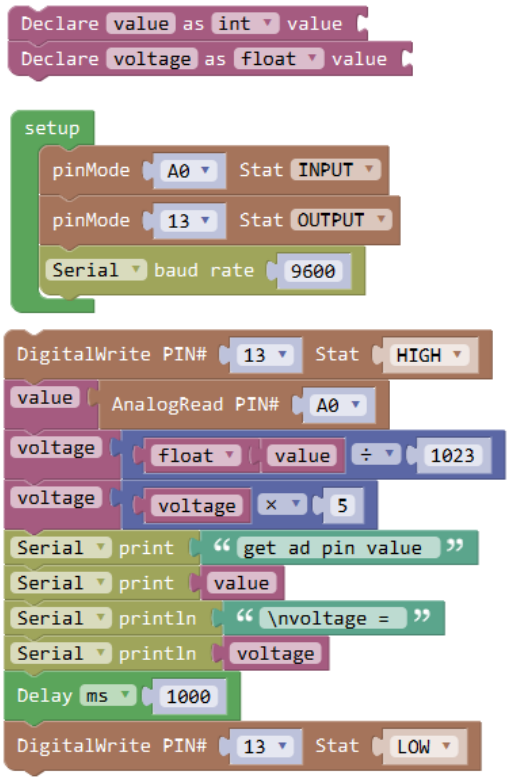
MBlock prepared AD digital-to-analog conversion program as shown in the figure below:

- 

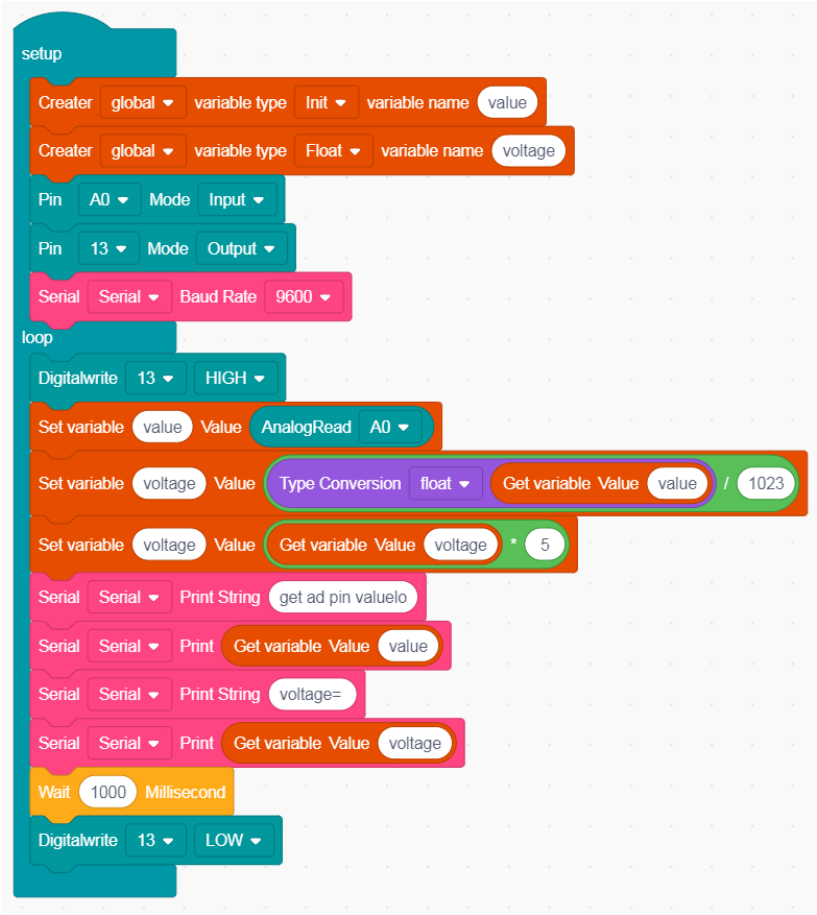
-- Set the digital pin output to low or high level



Mixly graphical programming program



MagicBlock graphical programming program


setup
Creater global variable type Init variable name value
Creater global variable type Float variable name voltage
Pin A0 Mode Input
Pin 13 Mode Output
Serial Serial Baud Rate 9600
loop
Digitalwrite 13 HIGH
Set variable value Value AnalogRead A0
Set variable voltage Value Type Conversion float Get variable Value value / 1023
Set variable voltage Value Get variable Value voltage * 5
Serial Serial Print String get ad pin valuelo
Serial Serial Print Get variable Value value
Serial Serial Print String voltage=
Serial Serial Print Get variable Value voltage
Wait 1000 Millisecond
Digitalwrite 13 LOW