

AD analog-to-digital conversion experiment

Introduction to AD conversion

We hear A/D or D/A conversion in the professional vocabulary a lot, so what are the A/D and D/A? A/D (Analog to Digital) means the device to convert analog signal into digital signal, then DA converts digital signal into analog signal.

The analog can be voltage, current and other electrical signals, or pressure, temperature, humidity, displacement, sound and other non-electrical signals. However, before A/D conversion, input signals into A/D converter must be converted into voltage signals by various sensors. After A/D conversion, the output digital signal can have 8 bits, 10 bits, 12 bits, 14 bits, 16 bits and so on.

DA conversion is a digital-to-analog conversion, which converts discrete digital quantities into connected analog quantities. Analog-to-digital conversion corresponds to analog-to-digital conversion, which is the inverse process of analog-to-digital conversion. Next, we will introduce digital-to-analog conversion from the aspects of converter classification, technical indicators, adc methods and adc parameters

Arduino has six analog interfaces numbered from 0 to 5, the six interface can also be interface function reuse. In addition to the analog interface function, the six interface can be used as digital interfaces and numbered from digital 14-19. After the simple understanding, let us begin our experiment below. Potentiometer is a kind of well-known typical analog value output element, it will be used to complete this experiment.

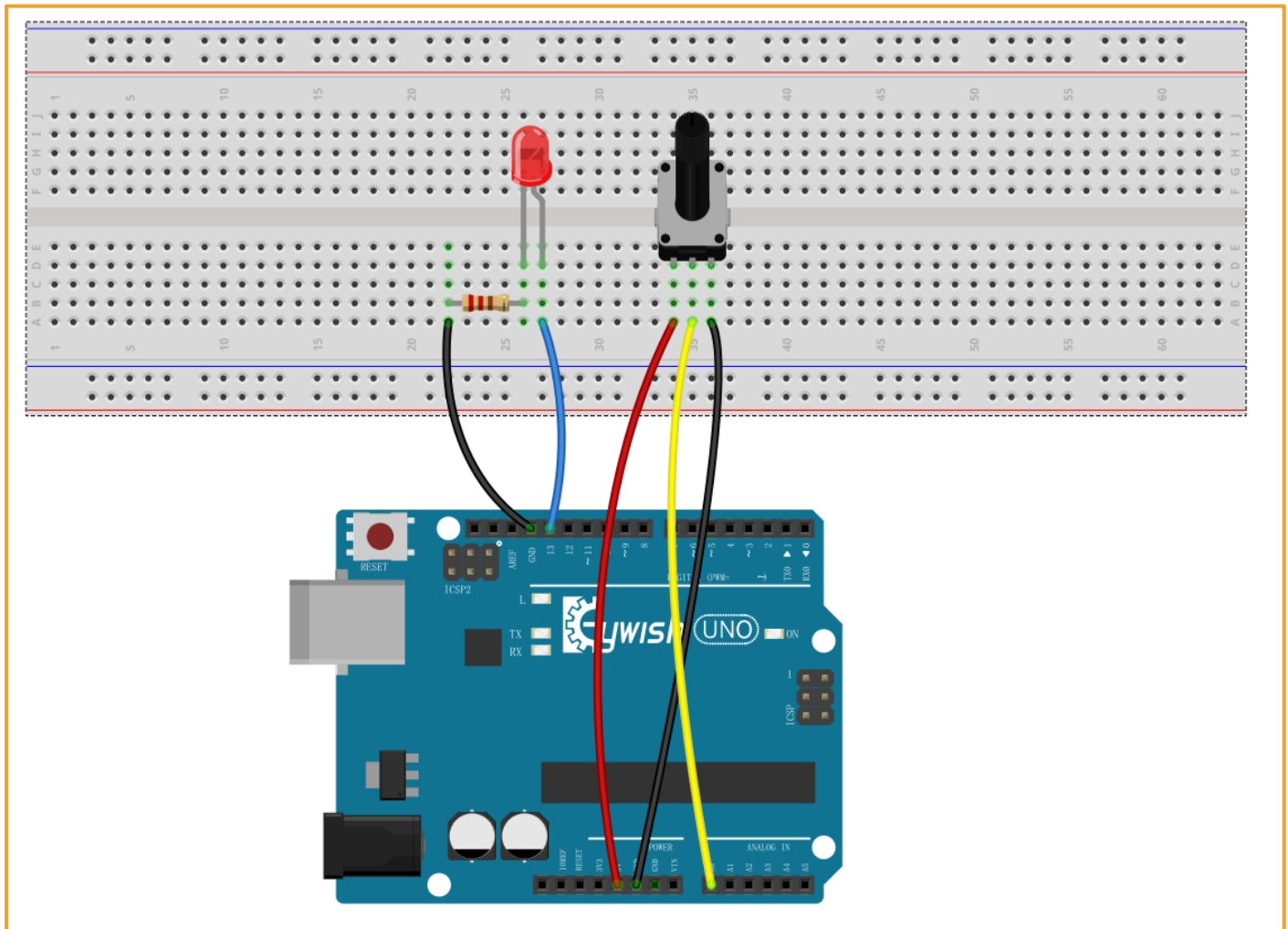
Component List

- ◆ Keywish Arduino Uno 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. Arduino 328 takes A/D sampling by 10 bit, so the analog value range is $0 \sim 2^{10} - 1$ (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

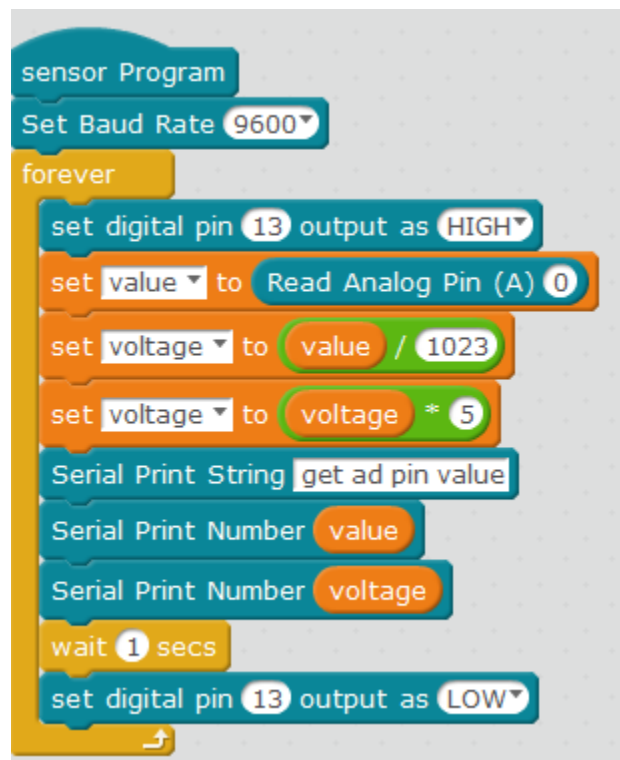
Code

```
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
}
```


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.

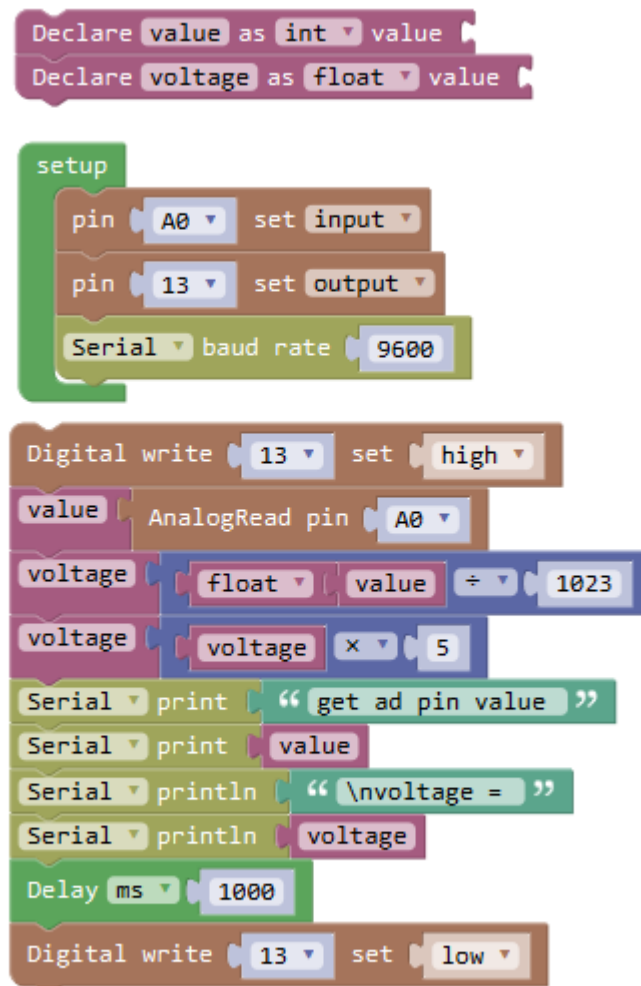
Mblock graphical programming program

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



Mixly graphical programming program

Mixly writes the AD digital-to-analog conversion program as shown in the figure below:



MagicBlock graphical programming program

MagicBlock writes the digital management program as shown in the figure below:

