

AD Conversion Experiment

Introduction

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 or rectification. In fact, it converts 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 A/D conversion, the input signal input to the A/D converter must be converted into voltage signals by various sensors. After A/D conversion, the output digital signal can have **8-bit, 10-bit, 12-bit, 14-bit, 16-bit** and so on.

DA conversion is digital-to-analog conversion, which converts discrete digital quantities into continuous analog quantities. The analog to digital conversion is the inverse process of the digital to analog conversion. Next, we will introduce the digital-to-analog conversion from these aspects of the category of converter, technical indicators, methods of analog-to-digital conversion and parameters of the analog-to-digital converter.

The Raspberry Pi expansion board has 8 analog interfaces(**A0-A7**). Next we start the following experiment. The potentiometer is a typical analog value output component and we will use it.

Component List

- ◆ Raspberry Pi main board
- ◆ Raspberry Pi expansion board
- ◆ Breadboard
- ◆ USB Data Cable
- ◆ 10k Potentiometer * 1
- ◆ Several jumper wires

Experimental Purpose

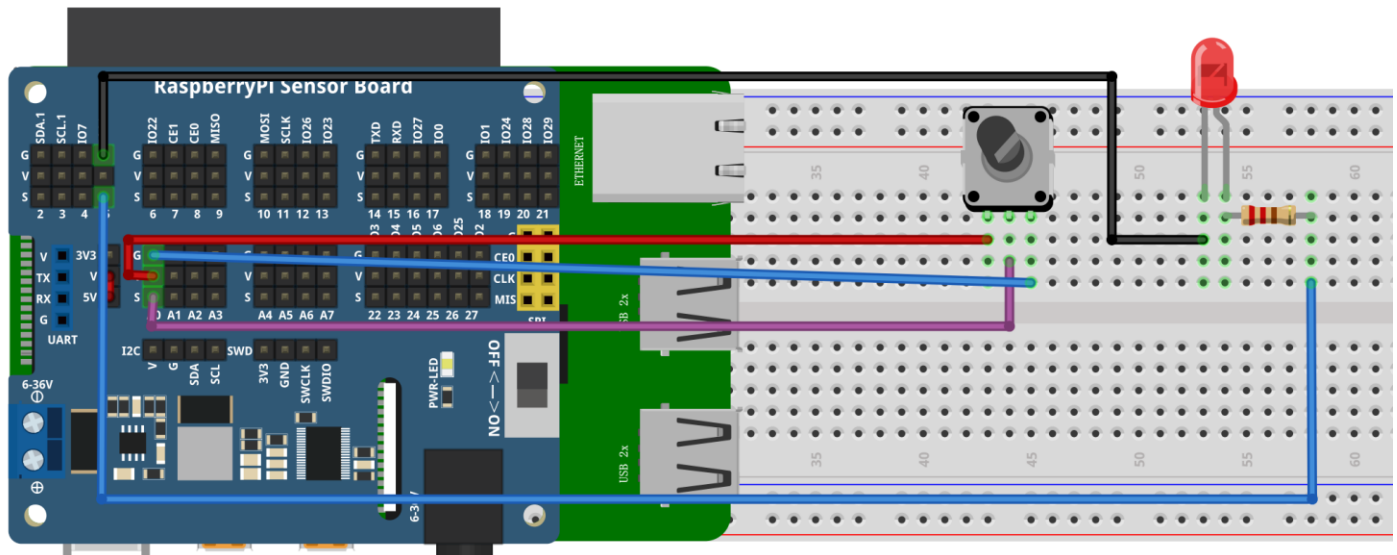
In this experiment, we convert the resistance value of the potentiometer into the analog value and then display it on the screen. This is also a very useful example that we need to learn.

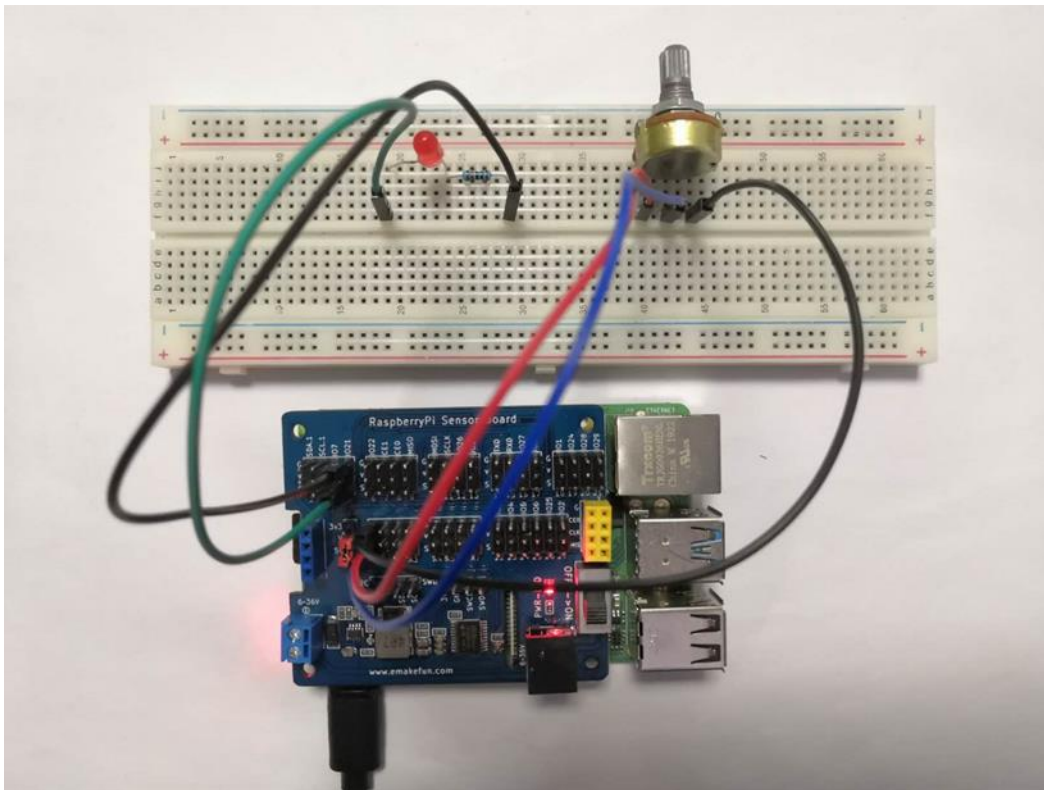
Experimental Principle

By reading the analog value of the sliding resistance meter and converting the analog value into the brightness value of the LED.

Wiring

Raspberry Pi	LED
GND	GND
IO21(wiringPi)/5(BCM)	IN
Raspberry Pi	Potentiometer
VCC	VCC
GND	GND
A0	OAT





C++ program

```
#include <stdio.h> // Import base library
#include <wiringPi.h> // Import the Raspberry Pi WiringPi encoding IO control library
#include <wiringPiI2C.h> // Import the Raspberry Pi WiringPi coded I2C control library

int LEDPIN = 21 ; // led light connection 2
int value = 0 ;
float voltage = 0.0 ;
int cyc = 5000;
void LED_pwm(int temp)
{
    digitalWrite(LEDPIN, HIGH);
    delayMicroseconds(temp);
    digitalWrite(LEDPIN, LOW);
    delayMicroseconds(cyc-temp);
}
int main()
{
    wiringPiSetup();
    wiringPiI2CSetup(0x04);
    pinMode(LEDPIN, OUTPUT);
    while(1)
```

```
{
    value = wiringPiI2CReadReg16(0x04, 0x10);
    printf("%d\n",value);
    LED_pwm(value);
    delay(1000); // 1 second delay
}
}
```

Python program

```
import time
import smbus as smbus
import RPi.GPIO as GPIO
import time

LEDPIN = 5
cyc = 0.5
ADC=smbus.SMBus(1)#Declare to use I2C 1
GPIO.setmode(GPIO.BCM)
GPIO.setup(LEDPIN, GPIO.OUT)

def led_pwm(val):
    GPIO.output(LEDPIN, True)
    time.sleep(val)
    GPIO.output(LEDPIN, False)
    time.sleep(cyc-val)

while True:
    ADC.write_byte(0x04, 0x10)#Write a byte to the slave
    val = ADC.read_word_data(0x04, 0x10)
    val = val / 10000
    print(val)#Raspberry Pi reads the data returned by the expansion board and prints it
    out
    led_pwm(val)
    time.sleep(1)#Delay 1 second
```

Java program

```
import com.pi4j.wiringpi.I2C;
import com.pi4j.wiringpi.Gpio;
```

```
import com.pi4j.wiringpi.GpioInterrupt;
import com.pi4j.wiringpi.GpioInterruptListener;
import com.pi4j.wiringpi.GpioInterruptEvent;
import com.pi4j.wiringpi.GpioUtil;
import com.pi4j.wiringpi.SoftPwm;

public class AD {
    static int LEDPIN = 21;
    static int value,value_y,value_sw;
    static {
        // setup wiring pi
        if (Gpio.wiringPiSetup() == -1) {
            System.out.println(" ==>> GPIO SETUP FAILED");
        }

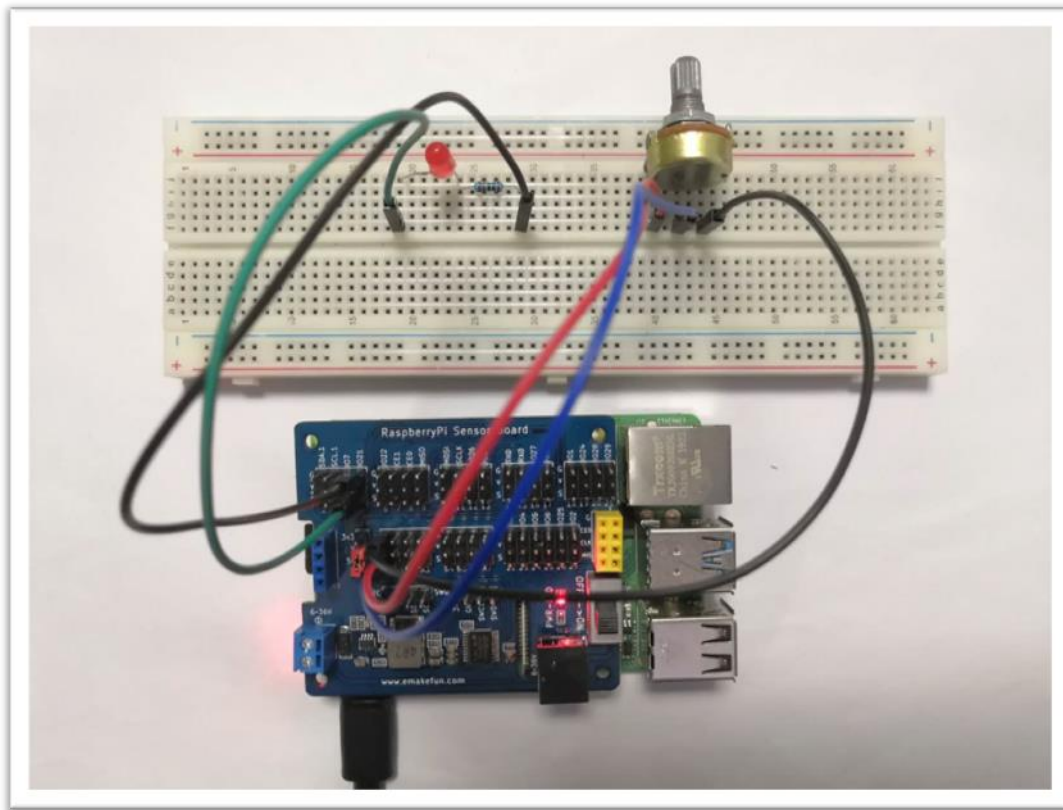
        Gpio.pinMode(LEDPIN, Gpio.OUTPUT);
    }

    static void led_pwm(int val){
        // create soft-pwm pins (min=0 ; max=100)
        SoftPwm.softPwmCreate(LEDPIN, 0, 100);
        SoftPwm.softPwmWrite(LEDPIN, val);
    }

    public static void main(String args[]) throws InterruptedException{

        int fd = I2C.wiringPiI2CSetup(0x04);
        for (;;) {
            value = I2C.wiringPiI2CReadReg16(fd, 0x10) * 100 / 4096;
            AD.led_pwm(value);
        }
    }
}
```

Experimental Effect



We have finished this experiment. Now, while you turn the potentiometer knob and you can see the change in the brightness of the LED. This method of reading analog value will always be used by people, due to most sensors only output the analog value, after you read the analog value, you must make corresponding algorithm processing for it and then you can apply it in experiments.