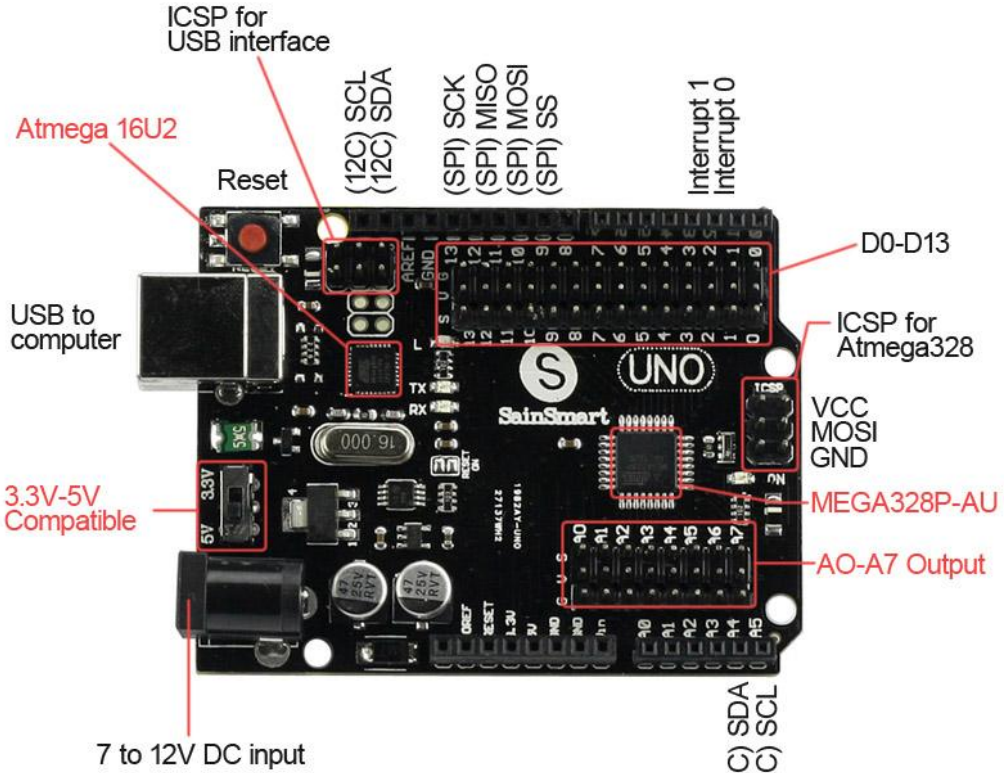
SainSmart UNO R3 Practice Kit



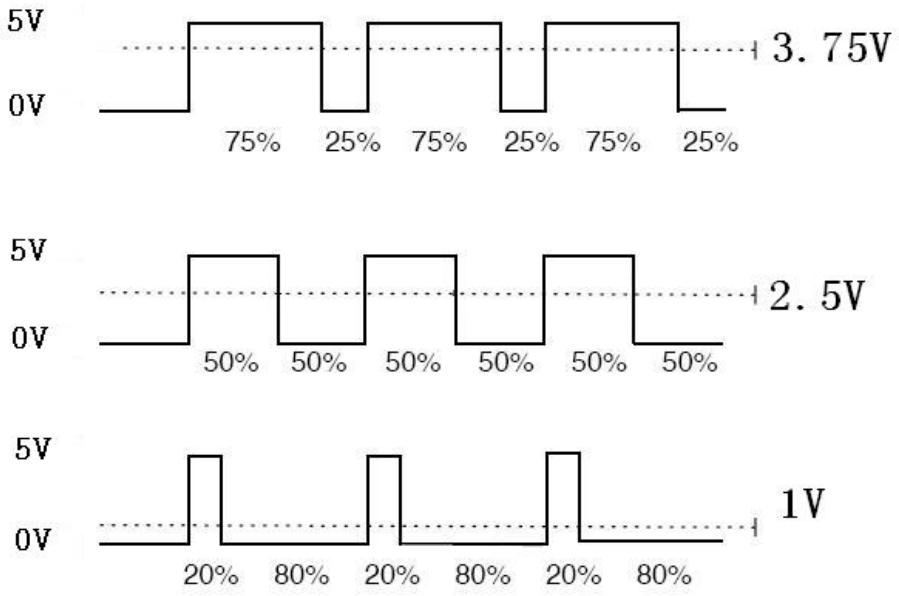
www.sainsmart.com

www.sainsmart.com/vanilla

SainSmart forun

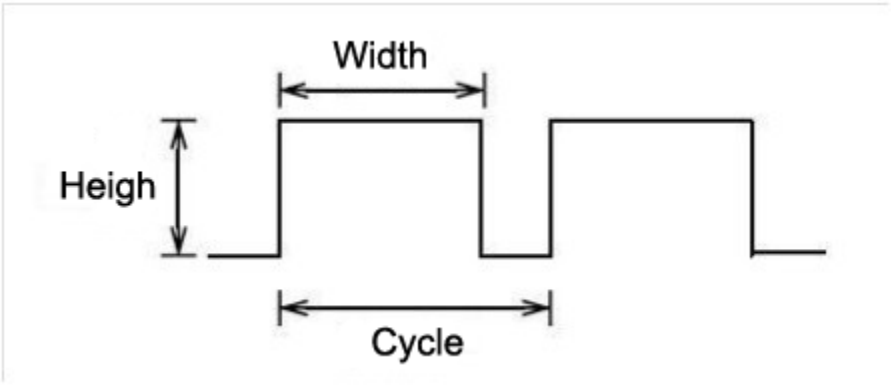
Chapter 1 PWM control light brightness

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.



In the graphic below, the green lines represent a regular time period. This duration or period is the inverse of the PWM frequency. In other words, with Arduino's PWM frequency at about 500Hz, the green lines would measure 2 milliseconds each. A call to analogWrite() is on a scale of 0 - 255, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time) for example.

PWM is used in many places, dimmer lighting, motor speed, sound production and so on.



Here are some of the PWM three basic parameters:

1, the pulse width variation (min / max)

2, the pulse period (the reciprocal of the number of frequencies within 1 second pulse)
3, the voltage height (e.g.: 0V-5V)

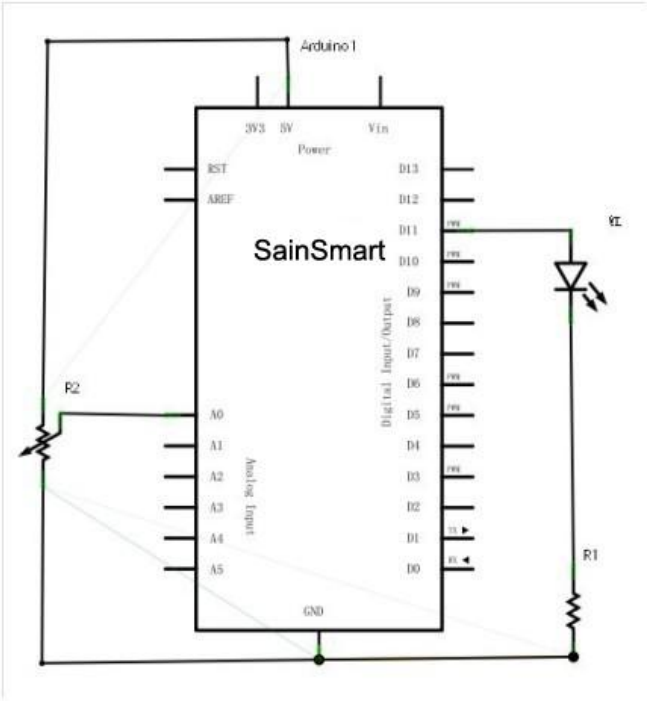
The 6 PWM interface of SainSmart UNO R3 controller is a digital interface 3,5,6,9,10,11, we have to complete a potentiometer control lamp experiment.

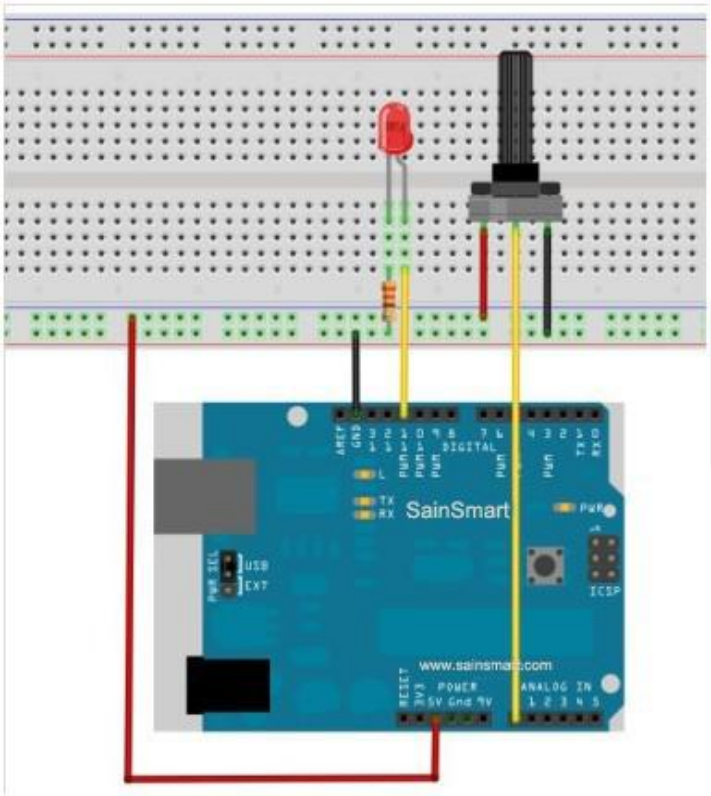
Experiment component

- Potentiometer :1;
- Red M5 LED : 1;
- 220Ω Resistance : 1;
- Bread board : 1;
- Jumper wires.

Potentiometer is the analog value input we received analog port lights connected to the PWM interface, which can produce different PWM signal lamp brightness change.

Connect your circuit as the below diagram





Example code:

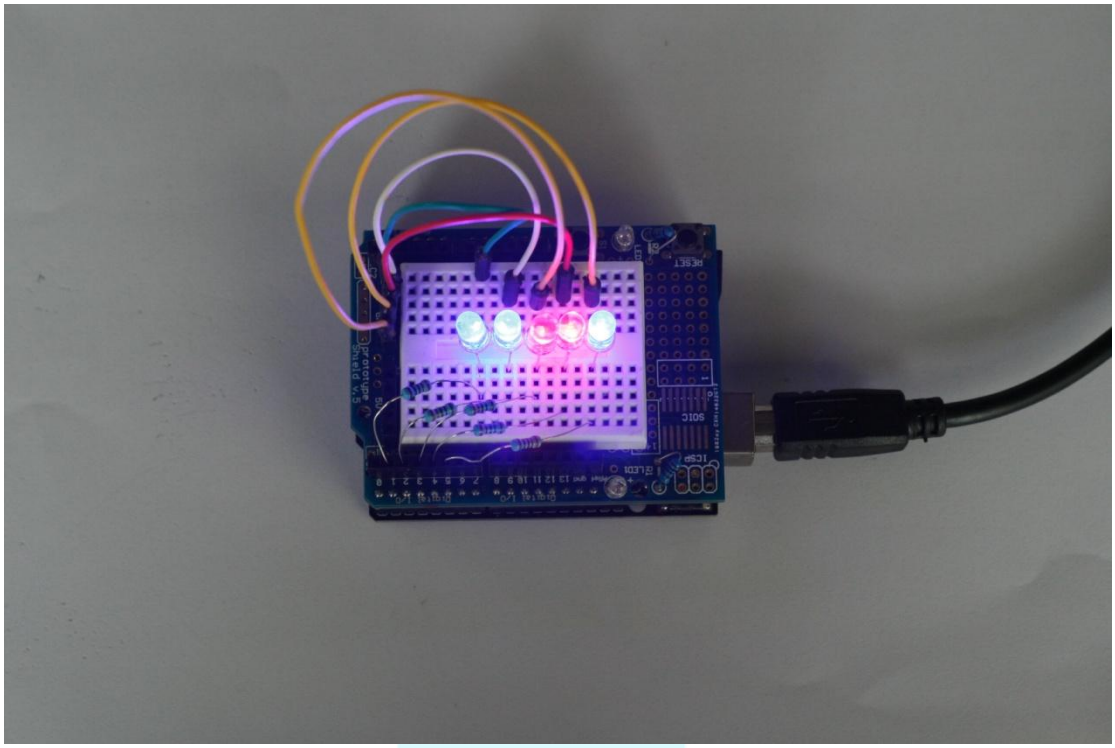
```
int potpin=0;//define analog interface 0;
int ledpin=11;//define digital interface 11(PWM output)
int val=0;// temporary values of the variables from the sensor
void setup()
{
pinMode(ledpin,OUTPUT);// define digital interface 11 output
Serial.begin(9600);// set baud rate is 9600
// Note: The analog interface is automatically set to enter
}
void loop()
{
val=analogRead(potpin);// Read the analog value of the sensor and assigned to the val
Serial.println(val);// Show val variable
analogWrite(ledpin,val/4);// turn on the LED and set the brightness (PWM output maximum 255)
delay(10);//delay 0.01s
}
```

Chapter 2 Advertising lights

Experiment component

- Red M5 LED : 6;
- 220Ω Resistance : 6;
- SainSmart Prototype shield : 1;
- Jumper wires.

Connect your circuit as the below diagram



Example code:

```
int BASE = 1 ;    //the I/O of 1st LED
int NUM = 6;      //amount of LED

void setup()
{
    for (int i = BASE; i < BASE + NUM; i ++)
    {
      pinMode(i, OUTPUT);     //set digital I/O output
    }
}
void loop()
{
    for (int i = BASE; i < BASE + NUM; i ++)
    {
      digitalWrite(i, LOW);       //set digital I/O low, that gradually turn off the lights
      delay(200);               //delay
    }
    for (int i = BASE; i < BASE + NUM; i ++)
    {
```

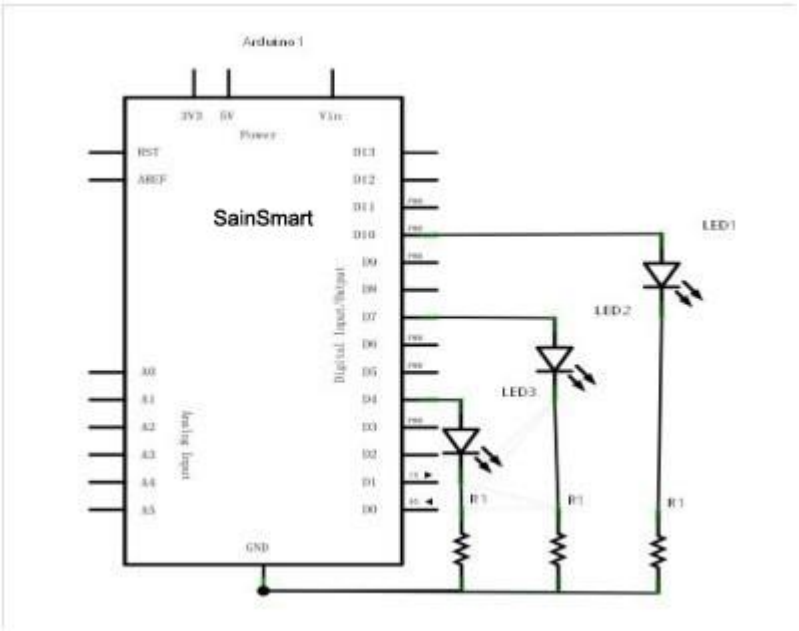
```
        digitalWrite(i, HIGH);       // set digital I/O low high, that gradually turn on the lights
        delay(200);                  //delay
    }
}
```

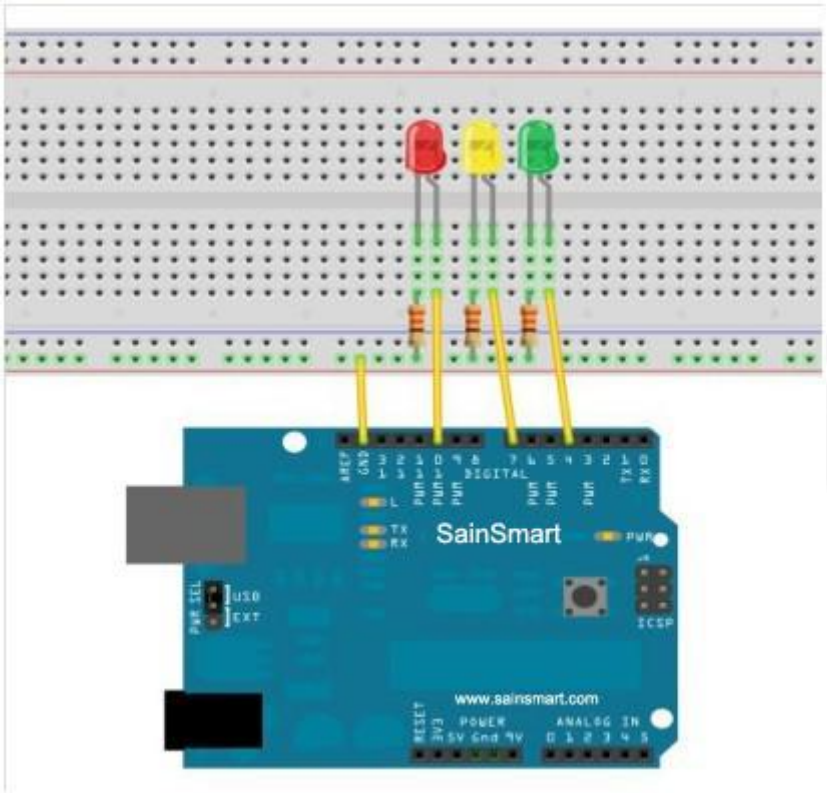
Chapter 3 Traffic lights

Experiment component

- Red M5 LED : 1;
- Yellow M5 LED : 1;
- Green M5 LED : 1
- 220Ω Resistance : 3;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram





Example code:

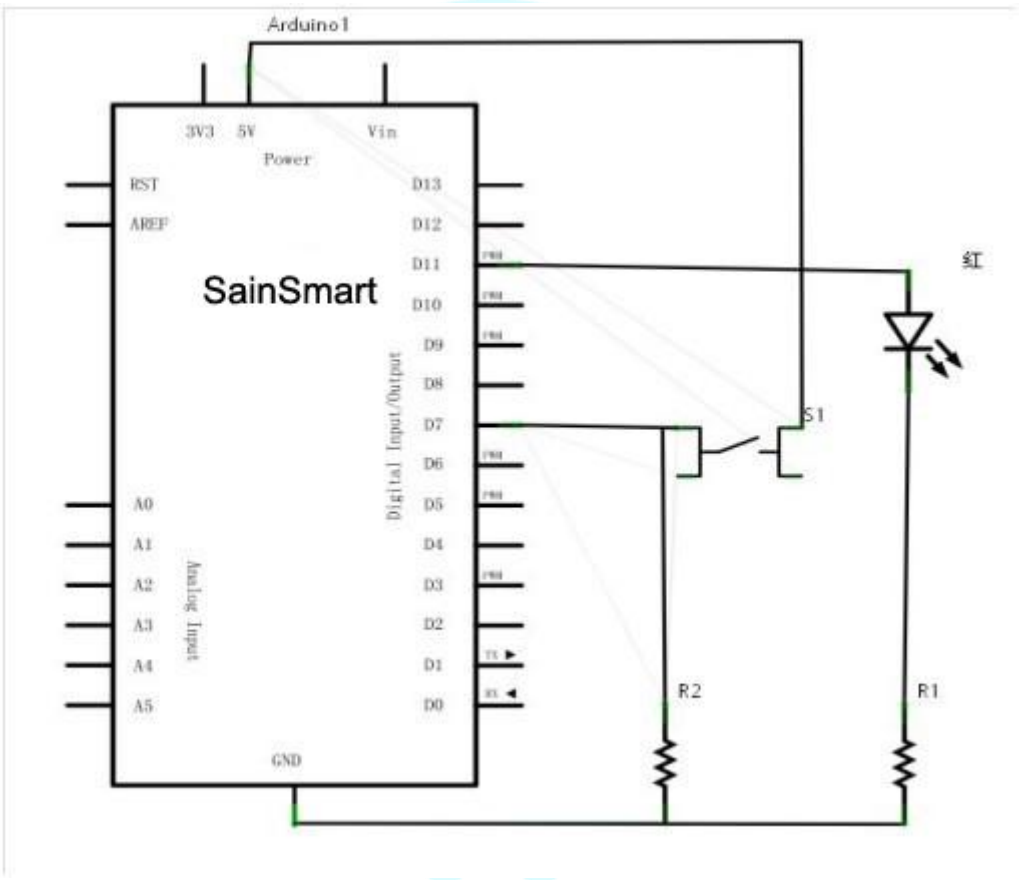
```
int redled =10; //define digital pin10
int yellowled =7; //define digital pin7
int greenled =4; //define digital pin4
void setup()
{
pinMode(redled, OUTPUT);//define red LED output
pinMode(yellowled, OUTPUT); //define yellow LED output
pinMode(greenled, OUTPUT); //define green LED output
}
void loop()
{
digitalWrite(redled, HIGH);//lights up red LED
delay(1000);//delay 1s
digitalWrite(redled, LOW); //light off red LED
digitalWrite(yellowled, HIGH);// lights up yellow LED
delay(200);//delay 02s
digitalWrite(yellowled, LOW);// light off yellow LED
digitalWrite(greenled, HIGH);// lights up red LED
delay(1000);//delay 1s
digitalWrite(greenled, LOW);// light off green LED
}
```

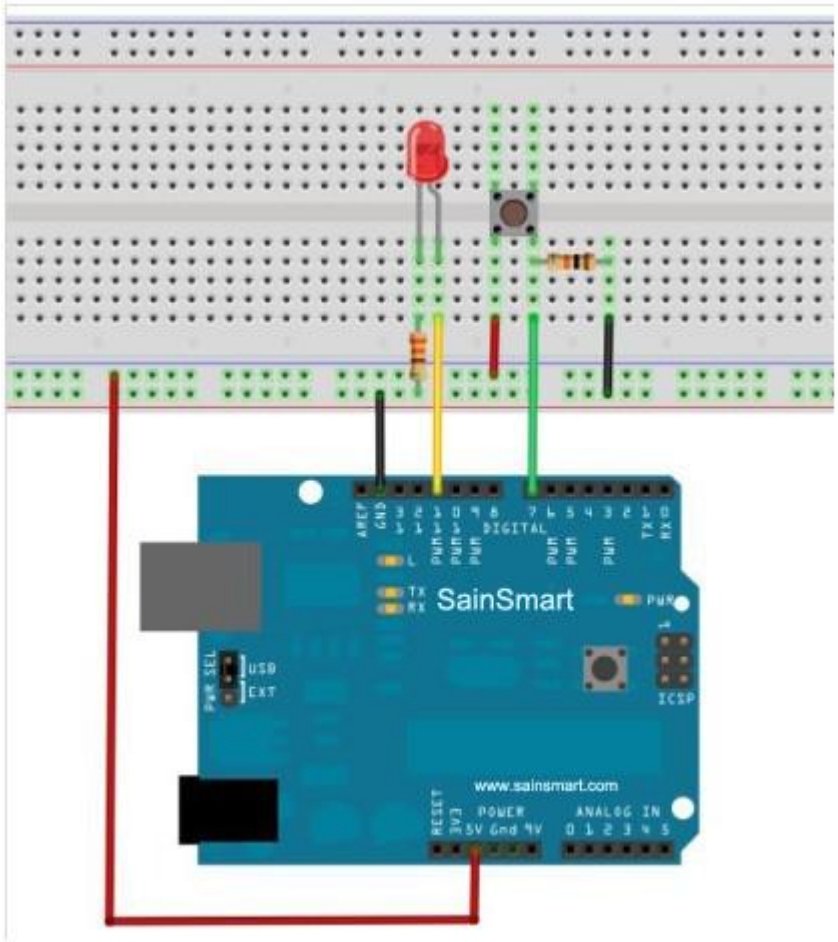
Chapter 4 Pushbutton control lights

Experiment component

- Red M5 LED : 1;
- Pushbutton : 1
- 220Ω Resistance : 1;
- 10kΩ Resistance : 1;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram





Example code:

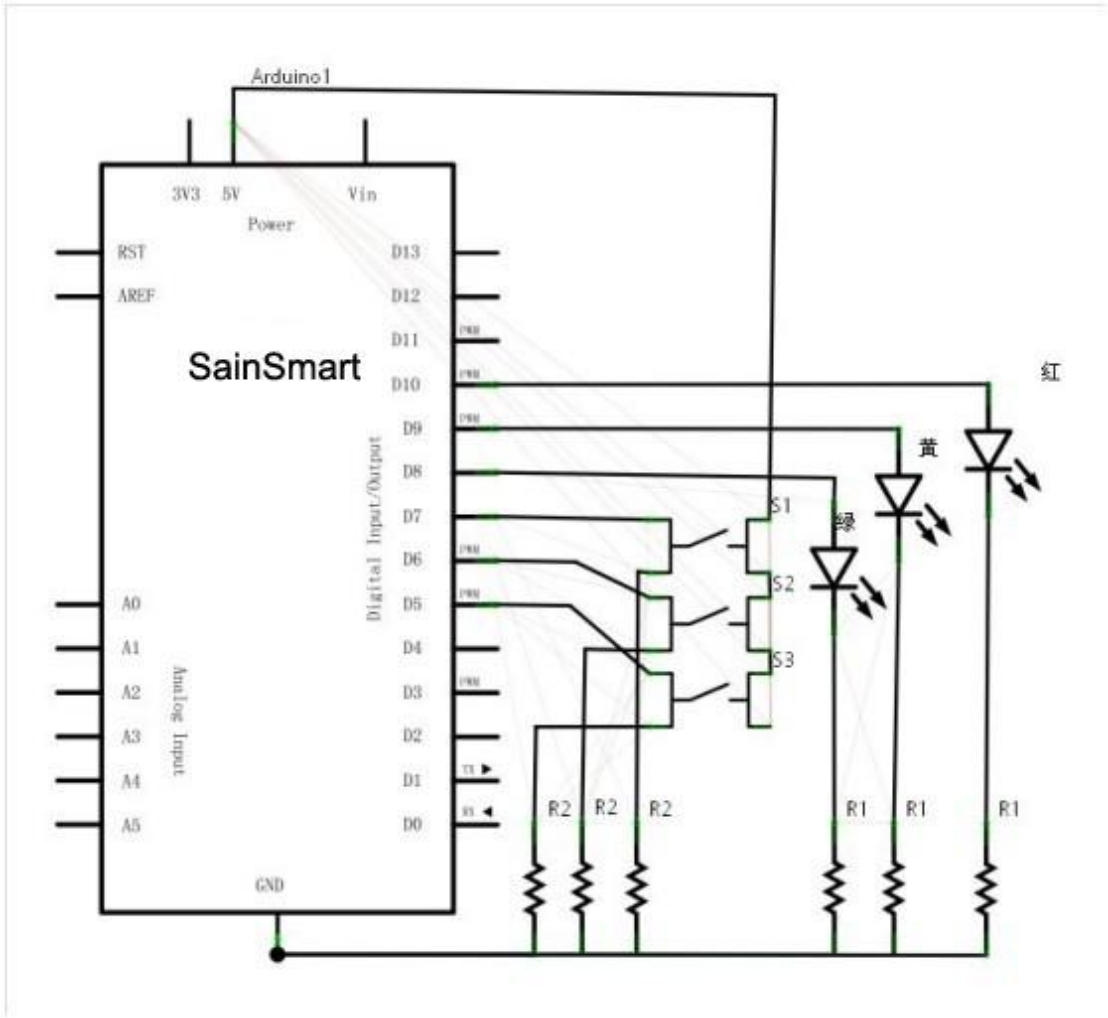
```
int ledpin=11;//define pin11
int inpin=7;//define pin7
int val;//define variable val
void setup()
{
pinMode(ledpin,OUTPUT);//define LED pin output
pinMode(inpin,INPUT);//define pushbutton pin input
}
void loop()
{
val=digitalRead(inpin);//    read the level value of pin7 and assigned it to val;
if(val==LOW)//if the pushbutton was push down, light up LED;
{ digitalWrite(ledpin,LOW);}
else
{ digitalWrite(ledpin,HIGH);}
}}
```

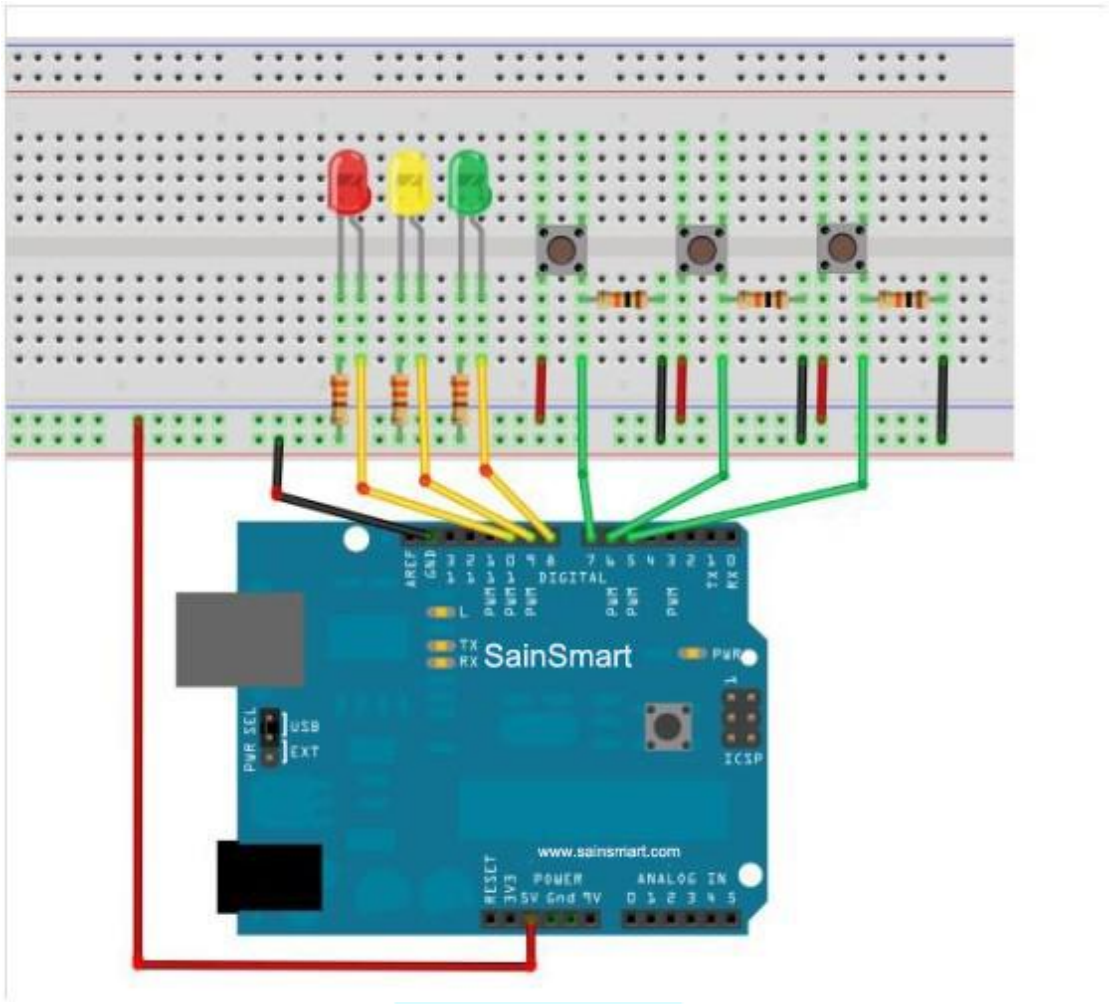
Chapter 5 Responder

Experiment component

- Red M5 LED : 1;
- Yellow M5 LED : 1;
- Green M5 LED : 1;
- Pushbutton : 3
- 220Ω Resistance : 3;
- 10kΩ Resistance : 3;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram





Example code:

```
int redled=10;
int yellowled=9;
int greenled=8;
int redpin=7;
int yellowpin=6;
int greenpin=5;
int red;
int yellow;
int green;
void setup()
{
pinMode(redled,OUTPUT);
pinMode(yellowled,OUTPUT);
pinMode(greenled,OUTPUT);
pinMode(redpin,INPUT);
pinMode(yellowpin,INPUT);
pinMode(greenpin,INPUT);
}
```

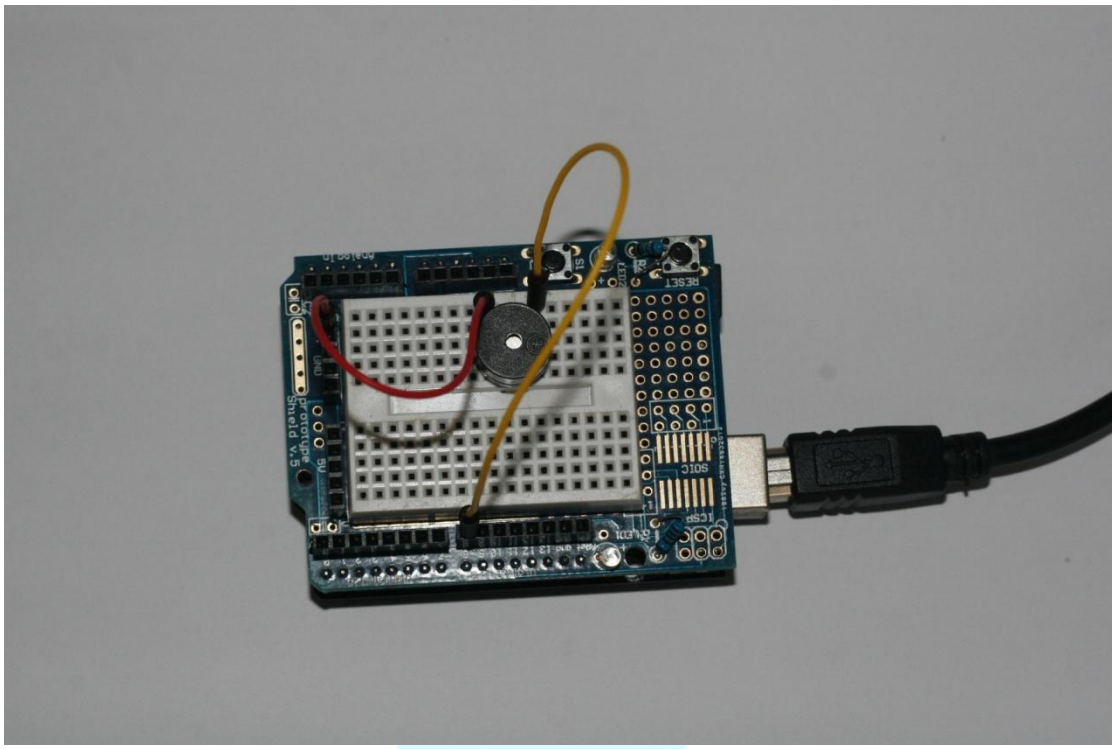
```
void loop()
{
red=digitalRead(redpin);
if(red==LOW)
{ digitalWrite(redled,LOW);}
else
{ digitalWrite(redled,HIGH);}
yellow=digitalRead(yellowpin);
if(yellow==LOW)
{ digitalWrite(yellowled,LOW);}
else
{ digitalWrite(yellowled,HIGH);}
green=digitalRead(greenpin);
if(green==LOW)
{ digitalWrite(greenled,LOW);}
else
{ digitalWrite(greenled,HIGH);}
}
```

Chapter 6 Buzzer

Experiment component

- Buzzer : 1;
- Pushbutton : 1
- SainSmart Prototype Shield : 1;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram



Example code:

```
int buzzer=8;//set buzzer I/O pin number;
void setup()
{
pinMode(buzzer,OUTPUT);//set buzzer I/O pin mode output;
}
void loop()
{
unsigned char i,j;//define variable;
while(1)
{
for(i=0;i<80;i++)// Output a frequency sound
{
digitalWrite(buzzer,HIGH);// vocalize
delay(1);//delay 1ms
digitalWrite(buzzer,LOW);//mute
delay(1);//delay 1ms
}
for(i=0;i<100;i++)// Output another frequency sound
```

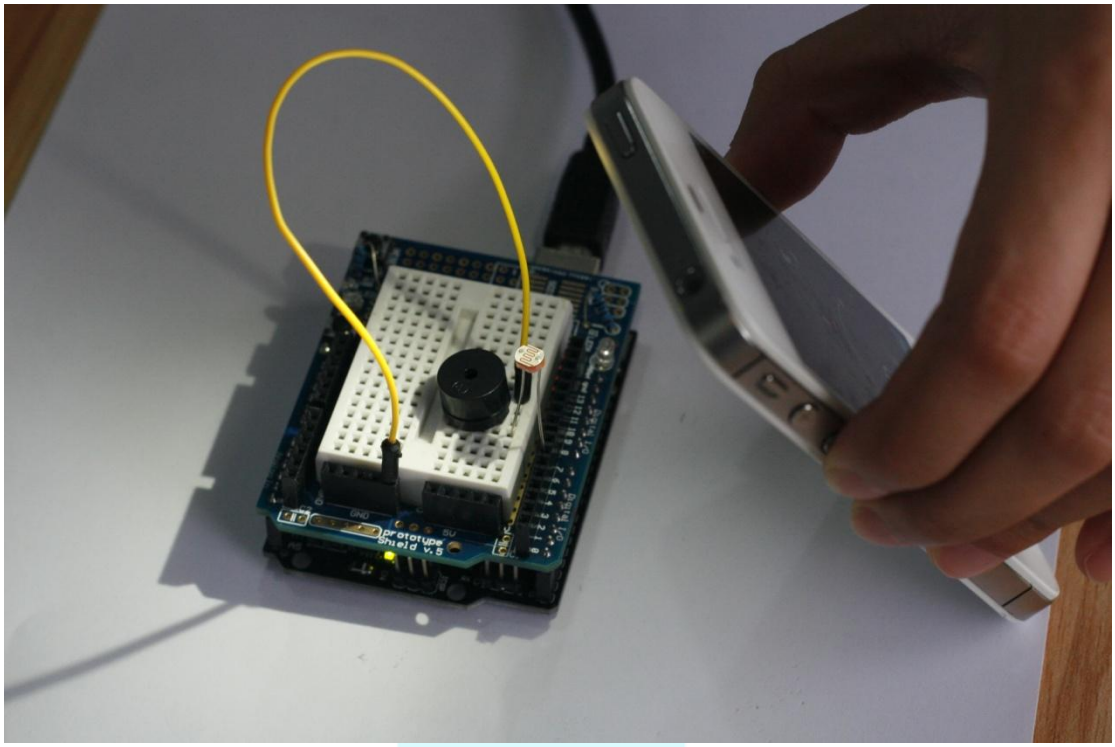
```
{
digitalWrite(buzzer,HIGH);//vocalize
delay(2);//delay 2ms
digitalWrite(buzzer,LOW);//mute
delay(2);//delay 2ms
}
}
}
```

Chapter 7 Light control sound

Experiment component

- Buzzer : 1;
- Photoresistor : 1
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram



Example code:

```
void setup()
{
pinMode(6,OUTPUT);
}
void loop()
{
while(1)
{
char i,j;
while(1)
{
for(i=0;i<80;i++) // Output a frequency sound
{
digitalWrite(6,HIGH);
delay(1);
digitalWrite(6,LOW);
delay(1);
}
```

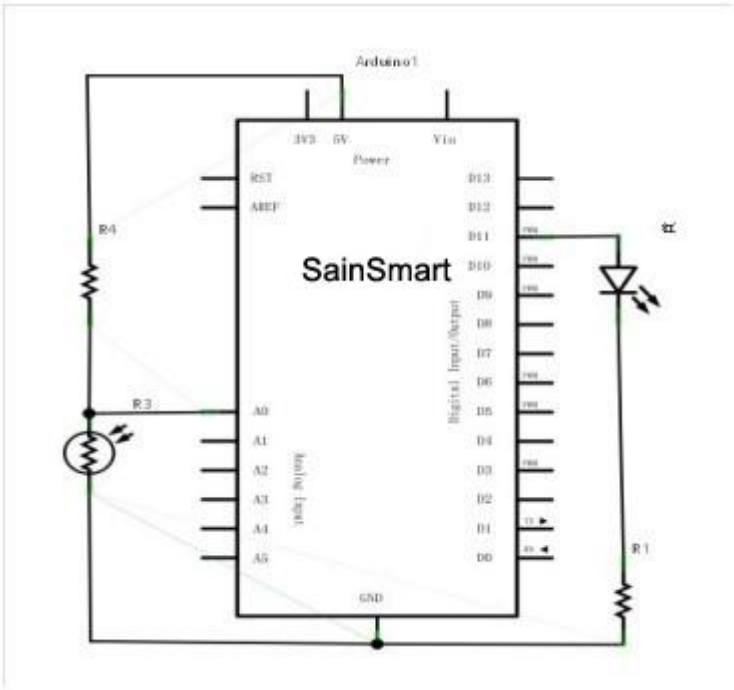
```
for(i=0;i<100;i++) // Output another frequency sound
{
digitalWrite(6,HIGH);
delay(2);
digitalWrite(6,LOW);
delay(2);
}
}
}
}
```

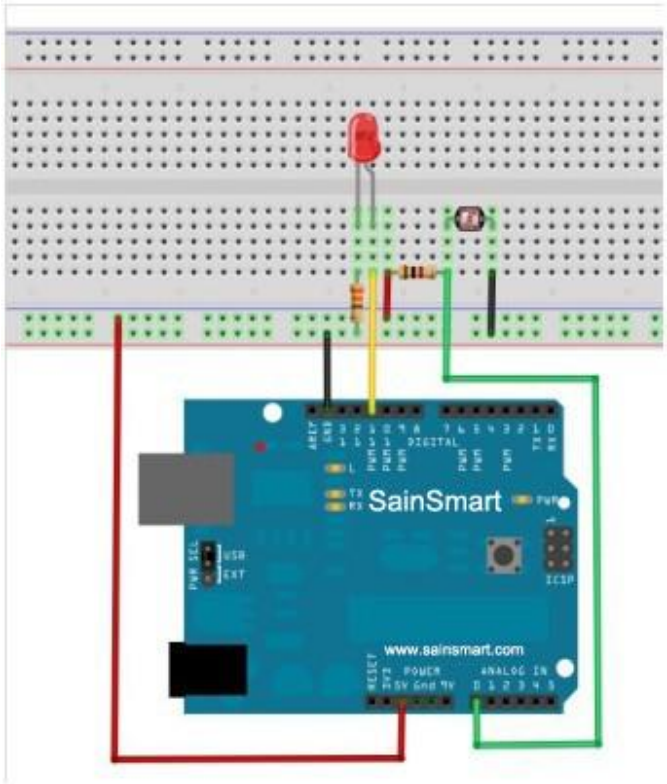
Chapter 8 Photosensitive lights

Experiment component

- Red M5 LED : 1;
- Photoresistor : 1;
- 220Ω Resistance : 1;
- 10kΩ Resistance : 1;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram





Example code:

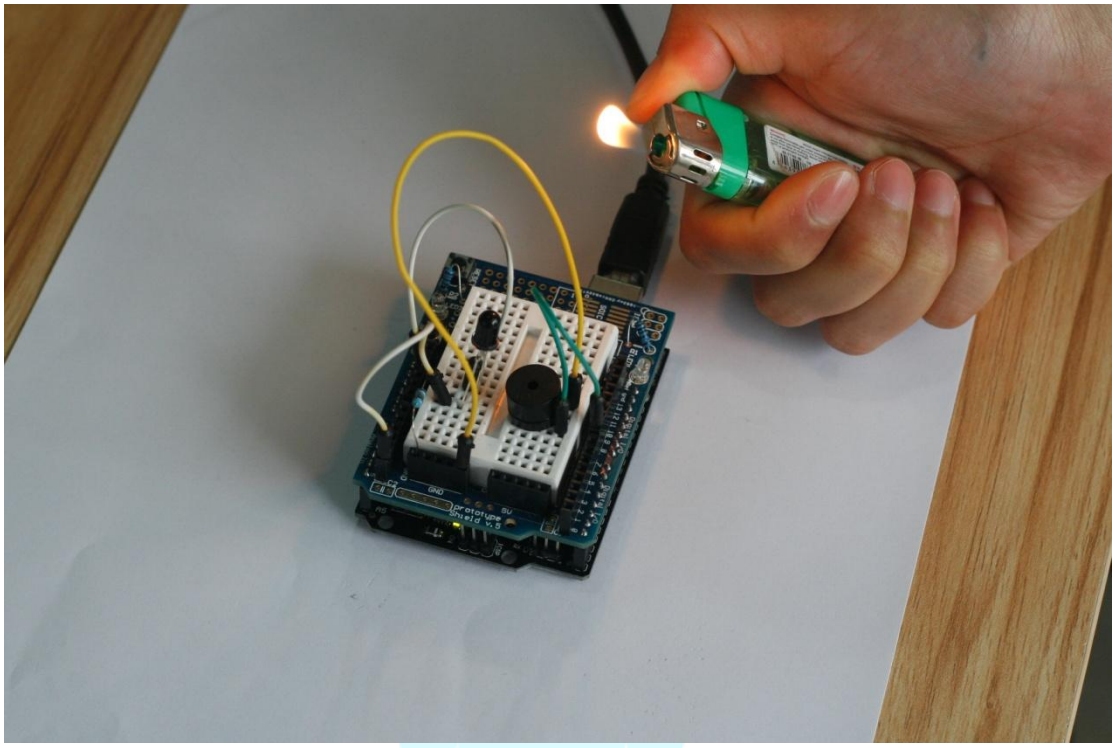
```
int potpin=0;//define analog pin0 connected with photoresistor;
int ledpin=11;//define digital pin11 output PWM adjust LED brightness;
int val=0;//define variable;
void setup()
{
pinMode(ledpin,OUTPUT);//define digital pin11 output;
Serial.begin(9600);// set baud rate is 9600
}
void loop()
{
val=analogRead(potpin);// read the sensor analog value and assigned it to val;
Serial.println(val);//show the value of val;
analogWrite(ledpin,val);// light the LED and set the brightness (PWM output maximum 255)
delay(10);//delay 0.01s
}
```

Chapter 9 Flame sensor

Experiment component

- Flame sensor : 1;
- Buzzer : 1;
- 10Ω Resistance : 1;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram



Example code:

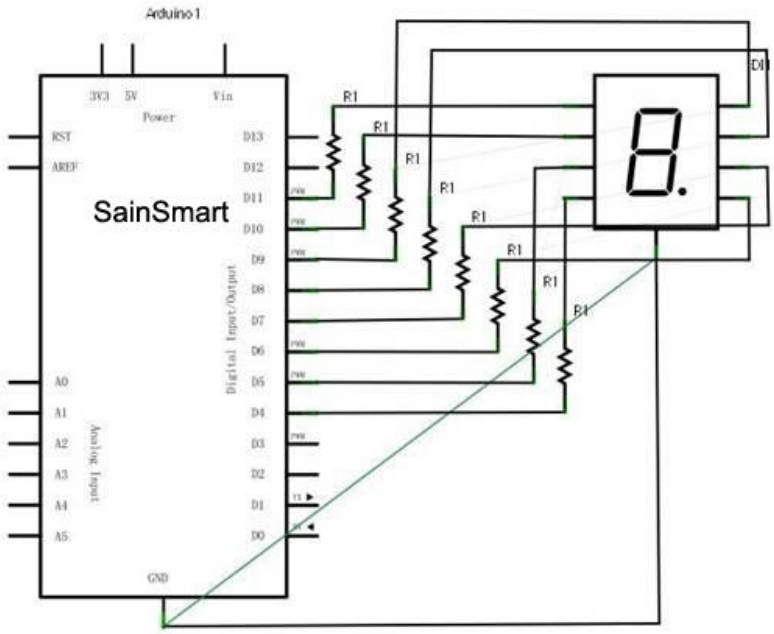
```
int flame=A5;//define flame sensor
  int Beep=8;// define buzzer interface digital interface 7
  int val=0;// define numeric variables
val void setup()
{ pinMode(Beep,OUTPUT);// defined LED output
  pinMode(flame,INPUT);// define buzzer input
  Serial.begin(9600);// set the baud rate to 9600 }
void loop() { val=analogRead(flame);// read flame sensor analog value
Serial.println(val);// output analog values, and print them out
  if(val>=600)// buzzer sounds when the analog value is greater than 600
  { digitalWrite(Beep,HIGH); } else { digitalWrite(Beep,LOW); } }
```

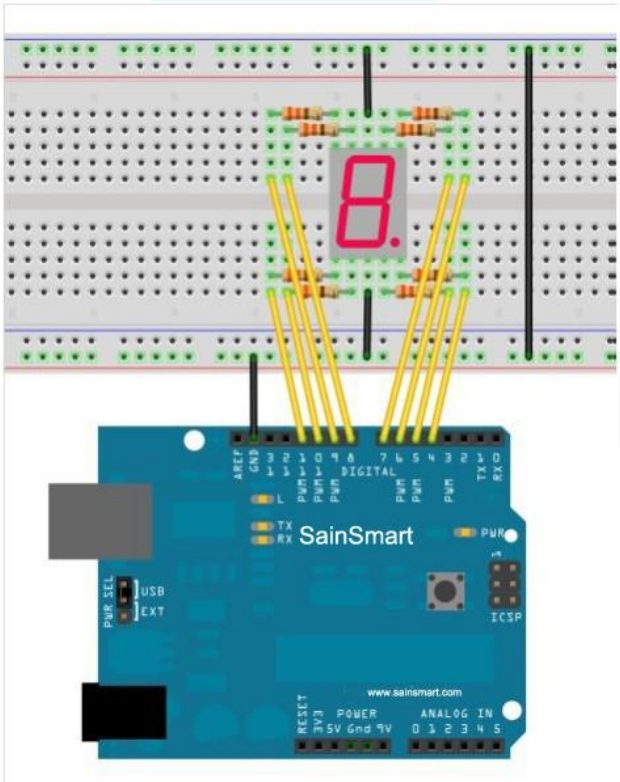
Chapter 10 Nixie Tube

Experiment component

- 8 Nixie Tube : 1;
- 220Ω Resistance : 1;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram





Example code:

```
// digital IO pin setting controls paragraphs
int a=7;// define digital pin7 connected to nixie tube a segment
int b=6;// define digital pin6 connected to nixie tube b segment
int c=5;// define digital pin5 connected to nixie tube c segment
int d=11;// define digital pin11 connected to nixie tube d segment
int e=10;// define digital pin10 connected to nixie tube e segment
int f=8;// define digital pin8 connected to nixie tube f segment
int g=9;// define digital pin9 connected to nixie tube g segment
int dp=4;// define digital pin4 connected to nixie tube dp segment
void digital_1(void) //display 1
{
unsigned char j;
digitalWrite(c,HIGH);// To the digital pin5 high, lit segment c
digitalWrite(b,HIGH);// lit segment b
for(j=7;j<=11;j++)// extinguish the remaining segment
digitalWrite(j,LOW);
digitalWrite(dp,LOW);// extinguish decimal point DP segment
}
void digital_2(void) //display 2
{
unsigned char j;
digitalWrite(b,HIGH);
digitalWrite(a,HIGH);
for(j=9;j<=11;j++)
digitalWrite(j,HIGH);
digitalWrite(dp,LOW);
digitalWrite(c,LOW);
digitalWrite(f,LOW);
}
void digital_3(void) //display 3
{
unsigned char j;
digitalWrite(g,HIGH);
digitalWrite(d,HIGH);
for(j=5;j<=7;j++)
digitalWrite(j,HIGH);
digitalWrite(dp,LOW);
digitalWrite(f,LOW);
digitalWrite(e,LOW);
}
void digital_4(void) //display 4
{
digitalWrite(c,HIGH);
```

```
digitalWrite(b,HIGH);
digitalWrite(f,HIGH);
digitalWrite(g,HIGH);
digitalWrite(dp,LOW);
digitalWrite(a,LOW);
digitalWrite(e,LOW);
digitalWrite(d,LOW);
}
void digital_5(void) //display 5
{
unsigned char j;
for(j=7;j<=9;j++)
digitalWrite(j,HIGH);
digitalWrite(c,HIGH);
digitalWrite(d,HIGH);
digitalWrite(dp,LOW);
digitalWrite(b,LOW);
digitalWrite(e,LOW);
}
void digital_6(void) //display 6
{
unsigned char j;
for(j=7;j<=11;j++)
digitalWrite(j,HIGH);
digitalWrite(c,HIGH);
digitalWrite(dp,LOW);
digitalWrite(b,LOW);
}
void digital_7(void) //display 7
{
unsigned char j;
for(j=5;j<=7;j++)
digitalWrite(j,HIGH);
digitalWrite(dp,LOW);
for(j=8;j<=11;j++)
digitalWrite(j,LOW);
}
void digital_8(void) //display 8
{
unsigned char j;
for(j=5;j<=11;j++)
digitalWrite(j,HIGH);
digitalWrite(dp,LOW);
}
```

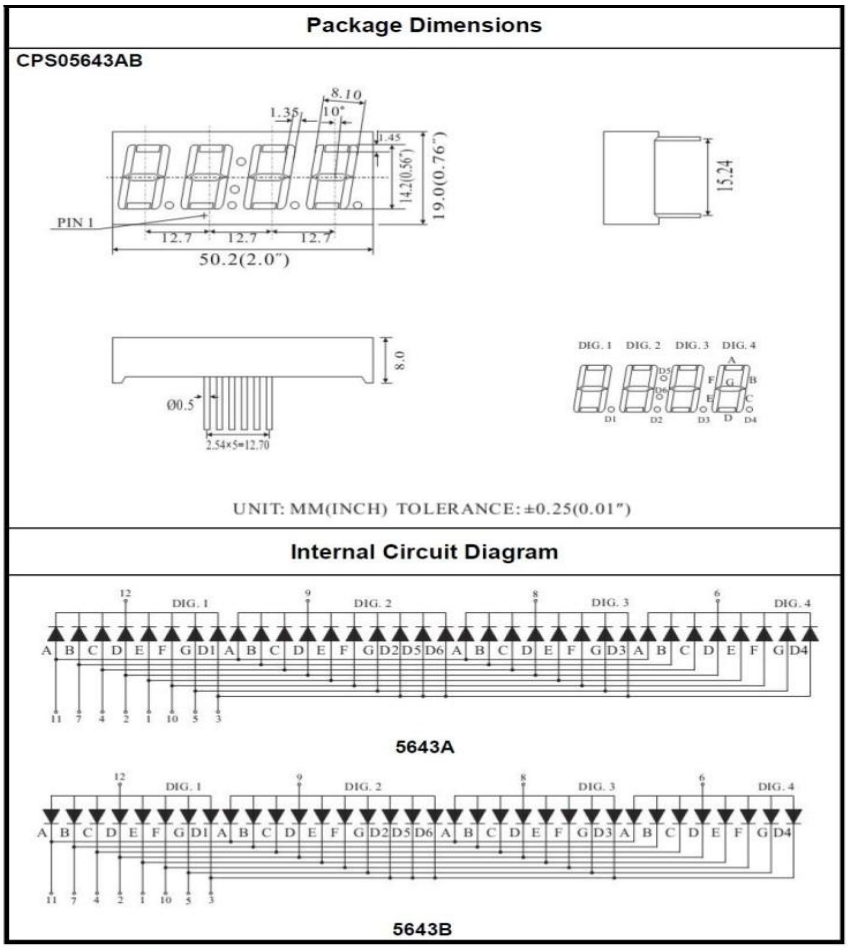
```
void setup()
{
int i;//
for(i=4;i<=11;i++)
pinMode(i,OUTPUT);//set pin4-11 output
}
void loop()
{
while(1)
{
digital_1();//display 1
delay(2000);//delay 2s
digital_2();//display 2
delay(1000); //delay 1s
digital_3();//display 3
delay(1000); //delay 1s
digital_4();//display 4
delay(1000); //delay1s
digital_5();//display 5
delay(1000); //delay 1s
digital_6();//display 6
delay(1000); //delay 1s
digital_7();//display 7
delay(1000); //delay 1s
digital_8();//display 8
delay(1000); //delay 1s
}
}
```

Chapter 11 4-bit Nixie Tube

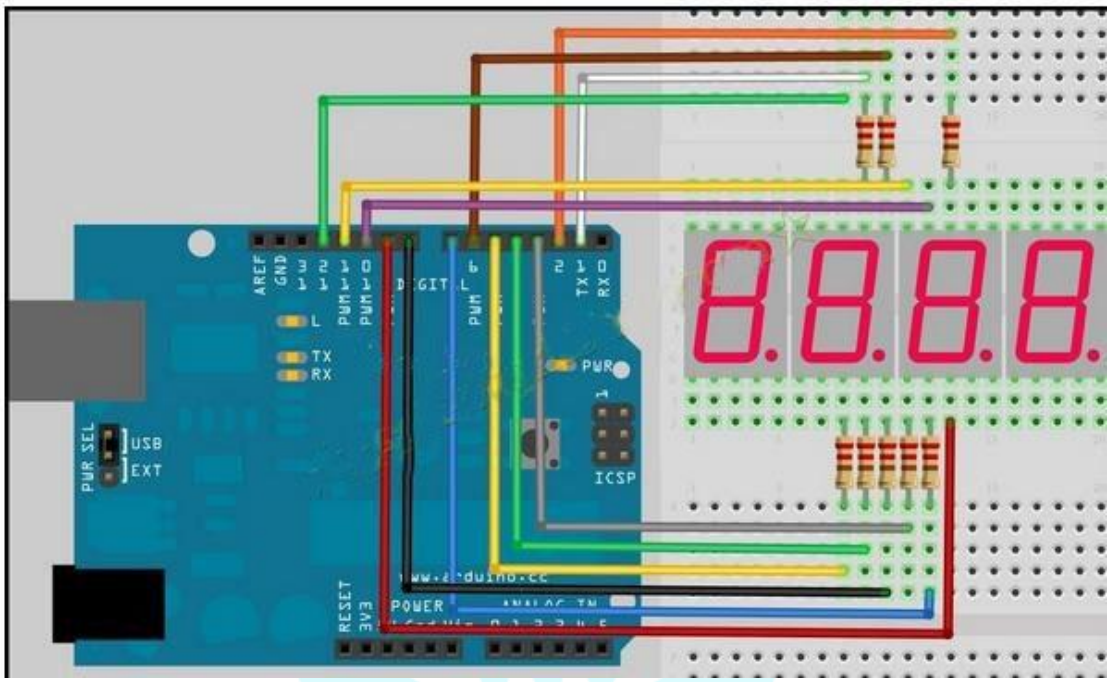
Experiment component

- 4-bit Nixie Tube : 1;
- 220Ω Resistance : 5;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram



Four Digits Displays Series



Example code:

```
//set anode interface
int a = 1;
int b = 2;
int c = 3;
int d = 4;
int e = 5;
int f = 6;
int g = 7;
int p = 8;
//set cathode interface
int d4 = 9;
int d3 = 10;
int d2 = 11;
int d1 = 12;
// Set variables
long n = 0;
int x = 100;
int del = 55;      // This number is fine-tuning of the clock
void setup()
{
   pinMode(d1, OUTPUT);
   pinMode(d2, OUTPUT);
   pinMode(d3, OUTPUT);
   pinMode(d4, OUTPUT);
   pinMode(a, OUTPUT);
```

```
    pinMode(b, OUTPUT);
    pinMode(c, OUTPUT);
    pinMode(d, OUTPUT);
    pinMode(e, OUTPUT);
    pinMode(f, OUTPUT);
    pinMode(g, OUTPUT);
    pinMode(p, OUTPUT);
}

void loop()
{
    clearLEDs();
    pickDigit(1);
    pickNumber((n/x/1000)%10);
    delayMicroseconds(del);

    clearLEDs();
    pickDigit(2);
    pickNumber((n/x/100)%10);
    delayMicroseconds(del);

    clearLEDs();
    pickDigit(3);
    dispDec(3);
    pickNumber((n/x/10)%10);
    delayMicroseconds(del);

    clearLEDs();
    pickDigit(4);
    pickNumber(n/x%10);
    delayMicroseconds(del);

    n++;

      if (digitalRead(13) == LOW)
      {
      n = 0;
      }
}

void pickDigit(int x)     //defing pickDigit(x), its role is turn on the dx port
{
    digitalWrite(d1, HIGH);
    digitalWrite(d2, HIGH);
```

```
  digitalWrite(d3, HIGH);
  digitalWrite(d4, HIGH);

  switch(x)
    {
    case 1:
    digitalWrite(d1, LOW);
    break;
    case 2:
    digitalWrite(d2, LOW);
    break;
    case 3:
    digitalWrite(d3, LOW);
    break;
  default:
    digitalWrite(d4, LOW);
    break;
    }
}

void pickNumber(int x)     //define pickNumber(x), Its role is to show digital x
{
  switch(x)
    {
  default:
    zero();
    break;
    case 1:
    one();
    break;
    case 2:
        two();
    break;
    case 3:
    three();
    break;
    case 4:
four();
    break;
    case 5:
    five();
    break;
    case 6:
        six();
```

```
        break;
        case 7:
              seven();
        break;
        case 8:
              eight();
        break;
        case 9:
        nine();
        break;
        }
}

void dispDec(int x)      // Set to open the decimal point
{
   digitalWrite(p, LOW);
}

void clearLEDs()    //clear the screen
{
   digitalWrite(a, LOW);
   digitalWrite(b, LOW);
   digitalWrite(c, LOW);
   digitalWrite(d, LOW);
   digitalWrite(e, LOW);
   digitalWrite(f, LOW);
   digitalWrite(g, LOW);
   digitalWrite(p, LOW);
}

void zero()      // Define the number 0 cathode pin switch
{
   digitalWrite(a, HIGH);
   digitalWrite(b, HIGH);
   digitalWrite(c, HIGH);
   digitalWrite(d, HIGH);
   digitalWrite(e, HIGH);
digitalWrite(f, HIGH);
   digitalWrite(g, LOW);
}

void one()
{
   digitalWrite(a, LOW);
```

```
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
}

void two()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, LOW);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, LOW);
  digitalWrite(g, HIGH);
}

void three()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, HIGH);
}

void four()
{
  digitalWrite(a, LOW);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}

void five()
{
  digitalWrite(a, HIGH);
```

```
  digitalWrite(b, LOW);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, LOW);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}

void six()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, LOW);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
  digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}

void seven()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
}

void eight()
{
  digitalWrite(a, HIGH);
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, HIGH);
digitalWrite(f, HIGH);
  digitalWrite(g, HIGH);
}

void nine()
{
  digitalWrite(a, HIGH);
```

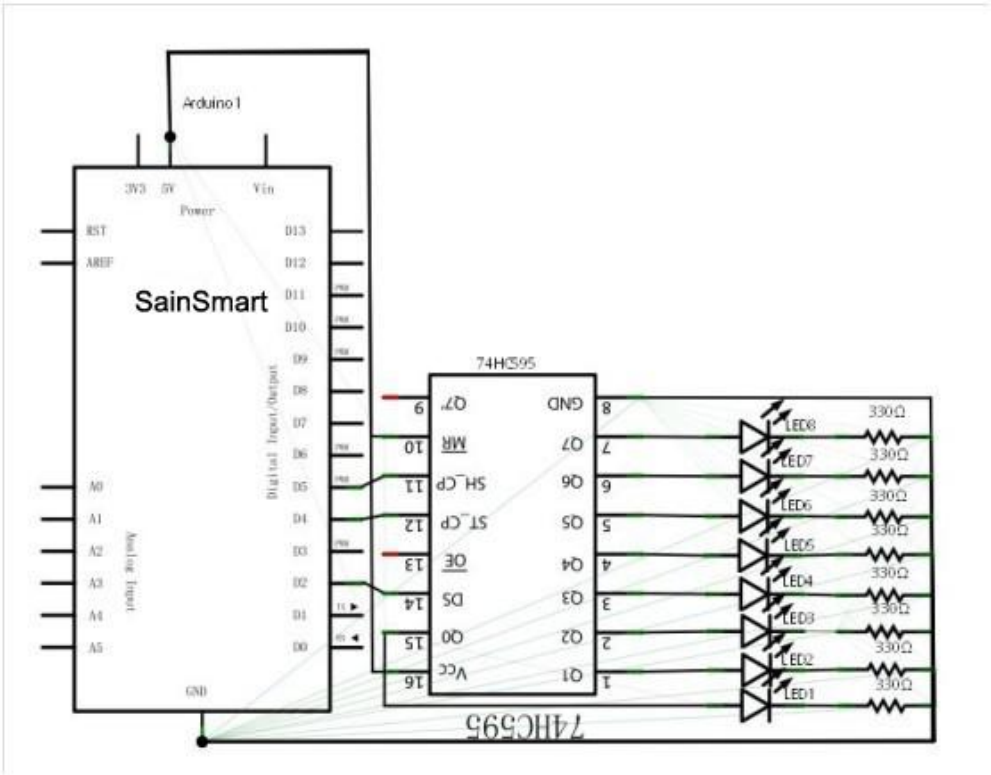
```
    digitalWrite(b, HIGH);
    digitalWrite(c, HIGH);
    digitalWrite(d, HIGH);
    digitalWrite(e, LOW);
    digitalWrite(f, HIGH);
    digitalWrite(g, HIGH);
}
```

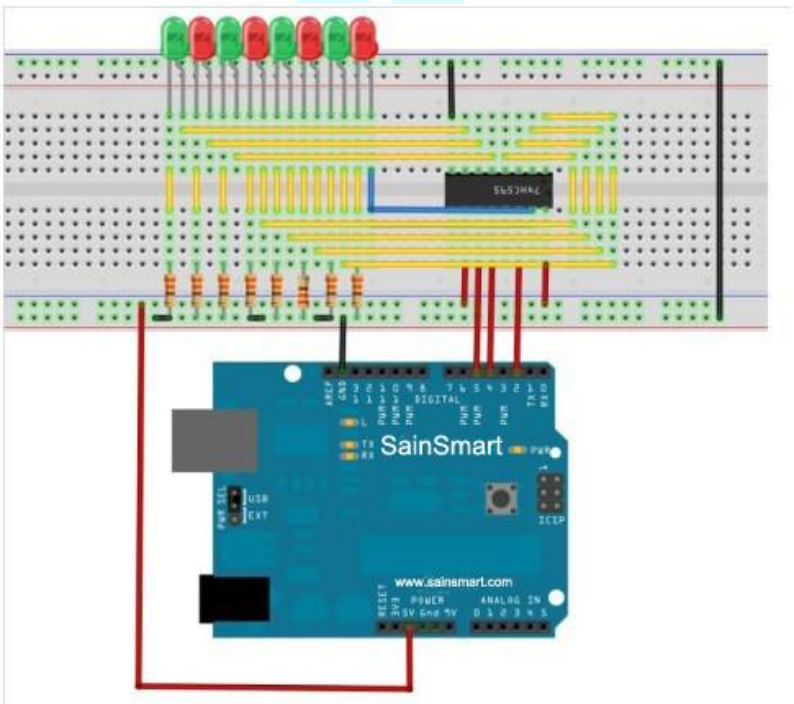
Chapter 12 74HC595

Experiment component

- 74HC595 : 1;
- Red M5 LED : 4;
- Green M5 LED : 4;
- 220Ω Resistance : 8;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram





Example code:

```
int data = 2;
int clock = 4;
int latch = 5;
int ledState = 0;
const int ON = HIGH;
const int OFF = LOW;
void setup()
{
pinMode(data, OUTPUT);
pinMode(clock, OUTPUT);
pinMode(latch, OUTPUT);
}
void loop()
{
int delayTime = 100;
for(int i = 0; i < 256; i++)
{
updateLEDs(i);
delay(delayTime);
}
}
void updateLEDs(int value)
{
digitalWrite(latch, LOW);
shiftOut(data, clock, MSBFIRST, value);
digitalWrite(latch, HIGH);
}
void updateLEDsLong(int value)
{
digitalWrite(latch, LOW);
for(int i = 0; i < 8; i++)
{
int bit = value & B10000000;
value = value << 1;
if(bit == 128){digitalWrite(data, HIGH);}
else{digitalWrite(data, LOW);}
digitalWrite(clock, HIGH);
delay(1);
digitalWrite(clock, LOW);
}
digitalWrite(latch, HIGH);
}
int bits[]={B00000001, B00000010, B00000100, B00001000, B00010000, B00100000,
```

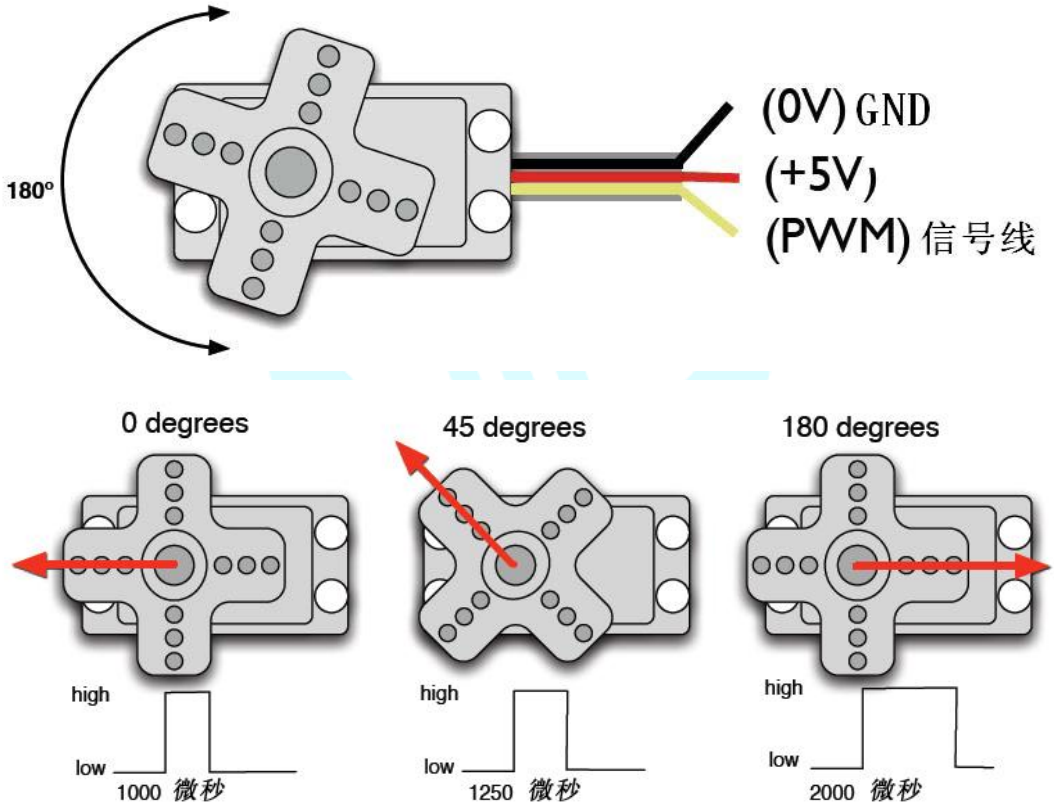
```
B01000000, B10000000};
int masks[] ={B11111110, B11111101, B11111011, B11110111, B11101111, B11011111,
B10111111, B01111111};
void changeLED(int led, int state)
{
ledState = ledState & masks[led];
if(state == ON){ledState = ledState | bits[led];}
updateLEDs(ledState);
}
```

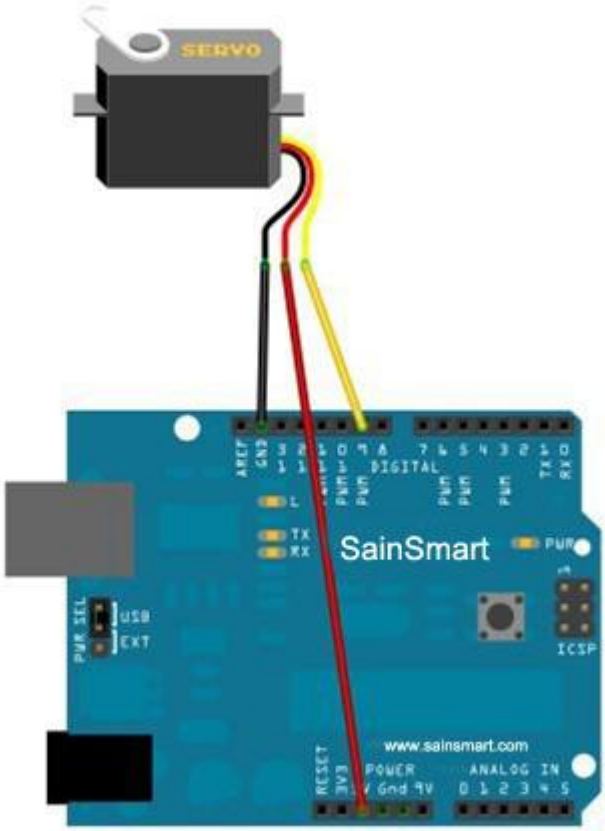
Chapter 13 Servo Motor

Experiment component

- RB-412 : 1;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram





Example code:

```
int servopin=9;// Defines a digital interface 9 to connect the servo signal line
int myangle;// Define the angular variable
int pulsewidth;// Defined variable pulse width
int val;
void servopulse(int servopin,int myangle)// Defines a pulse function
{
pulsewidth=(myangle*11)+500;// Angle will be transformed into pulse width value of 500-2480
digitalWrite(servopin,HIGH);// The servo interface level to high
delayMicroseconds(pulsewidth);//delay the microseconds of the pulse width value
digitalWrite(servopin,LOW);// The servo interface level to low
delay(20-pulsewidth/1000);
}
void setup()
{
pinMode(servopin,OUTPUT);// Set servo interface for the output interface
Serial.begin(9600);// Connected to the serial port, baud rate is 9600
Serial.println("servo=o_seral_simple ready" ) ;
}
void loop()//0-9 numbers will translate into 0-180 angle and the corresponding number of times the LED blinks
```

```
{
val=Serial.read();//Read the value of the serial port
if(val>'0'&&val<='9')
{
val=val-'0';//
val=val*(180/9);// Converts Numbers into perspective
Serial.print("moving servo to ");
Serial.print(val,DEC);
Serial.println();
for(int i=0;i<=50;i++) // Given enough time to make it to the steering gear specified Angle
{
servopulse(servopin,val);// Reference pulse function
}
}
}
```

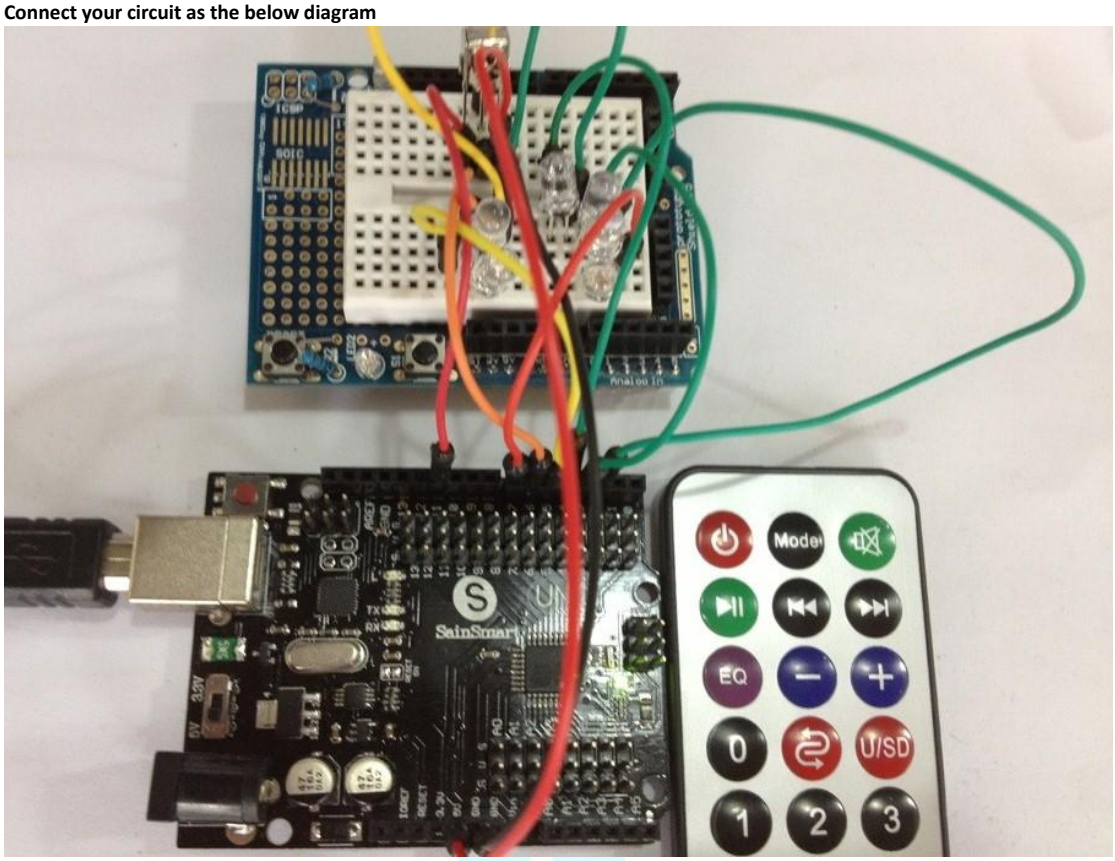
```
#include<Servo.h>//Define header files, there is one thing to note, you may directly click on the menu bar at the Arduino software, then Sketch>Importlibrary>Servo, call servo function. Or you may also directly enter the #include<Servo.h>, but noticed that there are no space between #include and <Servo.h>.
Servo myservo;// Define servo variable names
void setup()
{
myservo.attach(9);// Define servo interface (9, 10 can, shortcoming is only can control 2)
}
void loop()
{
myservo.write(90);// Set the steering gear rotating Angle
}
```

Chapter 14 Infrared remote control

Experiment component

- Infrared remote controller : 1;
- Infrared : 1;
- LED lamp : 6;
- 220Ω Resistance : 6;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram



Example code:

```
#include <IRremote.h>
int RECV_PIN = 11;
int LED1 = 2;
int LED2 = 3;
int LED3 = 4;
int LED4 = 5;
int LED5 = 6;
int LED6 = 7;
long on1    = 0x00FFA25D;
long off1 = 0x00FFE01F;
long on2 = 0x00FF629D;
long off2 = 0x00FFA857;
long on3 = 0x00FFE21D;
long off3 = 0x00FF906F;
```

```
long on4 = 0x00FF22DD;
long off4 = 0x00FF6897;
long on5 = 0x00FF02FD;
long off5 = 0x00FF9867;
long on6 = 0x00FFC23D;
long off6 = 0x00FFB047;
IRrecv irrecv(RECV_PIN);
decode_results results;
// Dumps out the decode_results structure.
// Call this after IRrecv::decode()
// void * to work around compiler issue
//void dump(void *v) {
//    decode_results *results = (decode_results *)v
void dump(decode_results *results) {
   int count = results->rawlen;
   if (results->decode_type == UNKNOWN)
      {
        Serial.println("Could not decode message");
      }
   else
    {
      if (results->decode_type == NEC)
         { Serial.print("Decoded NEC: ");
         }
      else if (results->decode_type == SONY)
         {
           Serial.print("Decoded SONY: ");
         }
      else if (results->decode_type == RC5)
         {
           Serial.print("Decoded RC5: ");
         }
      else if (results->decode_type == RC6)
         {
           Serial.print("Decoded RC6: ");
         }
        Serial.print(results->value, HEX);
        Serial.print(" (");
        Serial.print(results->bits, DEC);
        Serial.println(" bits)");
   }
        Serial.print("Raw (");
        Serial.print(count, DEC);
        Serial.print("): ");
```

```
for (int i = 0; i < count; i++)
        {
          if ((i % 2) == 1) {
          Serial.print(results->rawbuf[i]*USECPERTICK, DEC);
        }
       else
        {
          Serial.print(-(int)results->rawbuf[i]*USECPERTICK, DEC);
        }
       Serial.print(" ");
        }
          Serial.println("");
        }

void setup()
 {
   pinMode(RECV_PIN, INPUT);
   pinMode(LED1, OUTPUT);
   pinMode(LED2, OUTPUT);
   pinMode(LED3, OUTPUT);
   pinMode(LED4, OUTPUT);
   pinMode(LED5, OUTPUT);
   pinMode(LED6, OUTPUT);
   pinMode(13, OUTPUT);
   Serial.begin(9600);
   irrecv.enableIRIn(); // Start the receiver
 }
int on = 0;
unsigned long last = millis();

void loop()
{
   if (irrecv.decode(&results))
    {
      // If it's been at least 1/4 second since the last
      // IR received, toggle the relay
      if (millis() - last > 250)
        {
          on = !on;
//            digitalWrite(8, on ? HIGH : LOW);
          digitalWrite(13, on ? HIGH : LOW);
          dump(&results);
        }
      if (results.value == on1 )
```

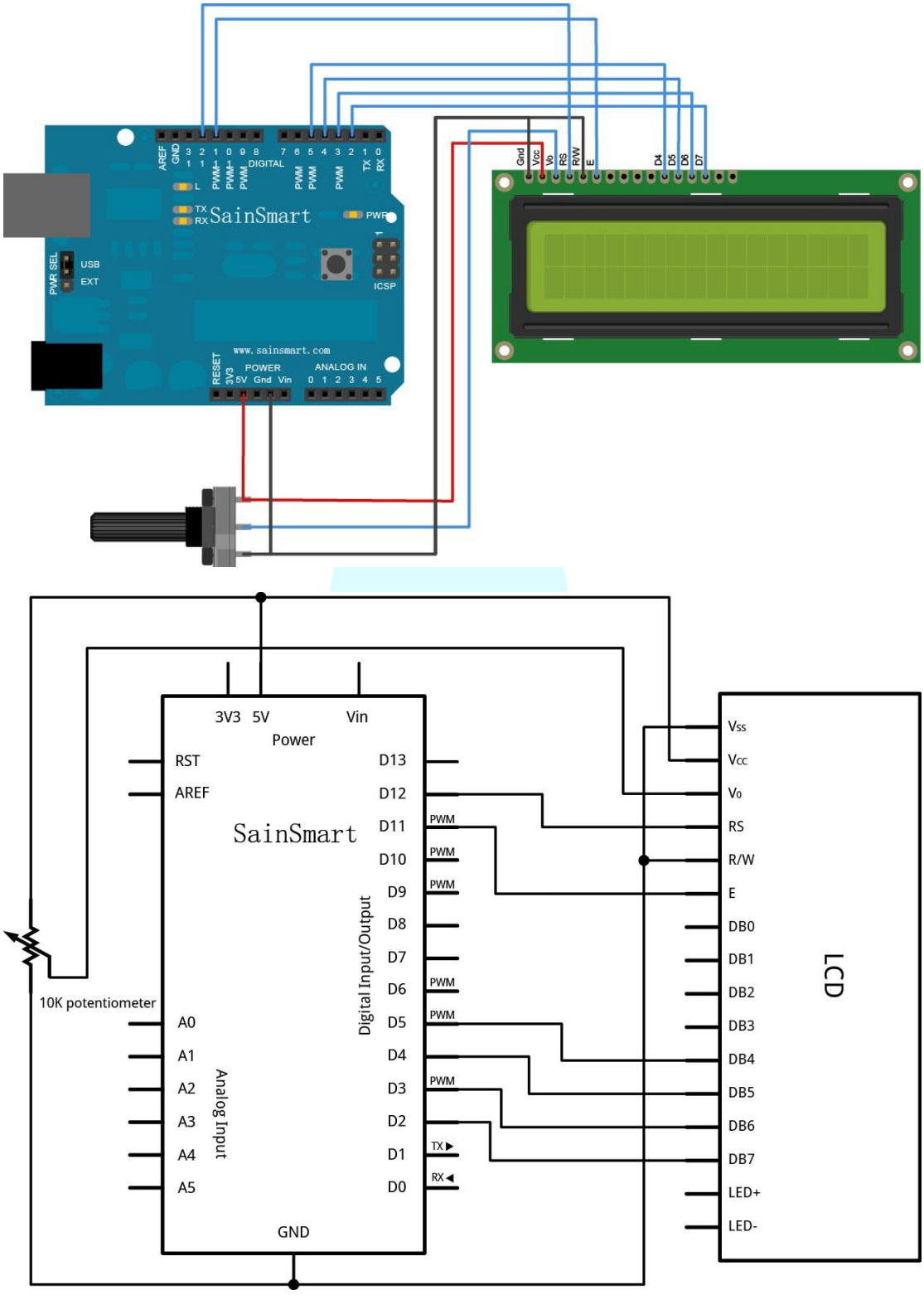
```
            digitalWrite(LED1, HIGH);
        if (results.value == off1 )
            digitalWrite(LED1, LOW);
        if (results.value == on2 )
            digitalWrite(LED2, HIGH);
        if (results.value == off2 )
            digitalWrite(LED2, LOW);
        if (results.value == on3 )
            digitalWrite(LED3, HIGH);
        if (results.value == off3 )
            digitalWrite(LED3, LOW);
        if (results.value == on4 )
            digitalWrite(LED4, HIGH);
        if (results.value == off4 )
            digitalWrite(LED4, LOW);
        if (results.value == on5 )
            digitalWrite(LED5, HIGH);
        if (results.value == off5 )
            digitalWrite(LED5, LOW);
        if (results.value == on6 )
            digitalWrite(LED6, HIGH);
        if (results.value == off6 )
            digitalWrite(LED6, LOW);
        last = millis();
        irrecv.resume(); // Receive the next value
    }
}
```

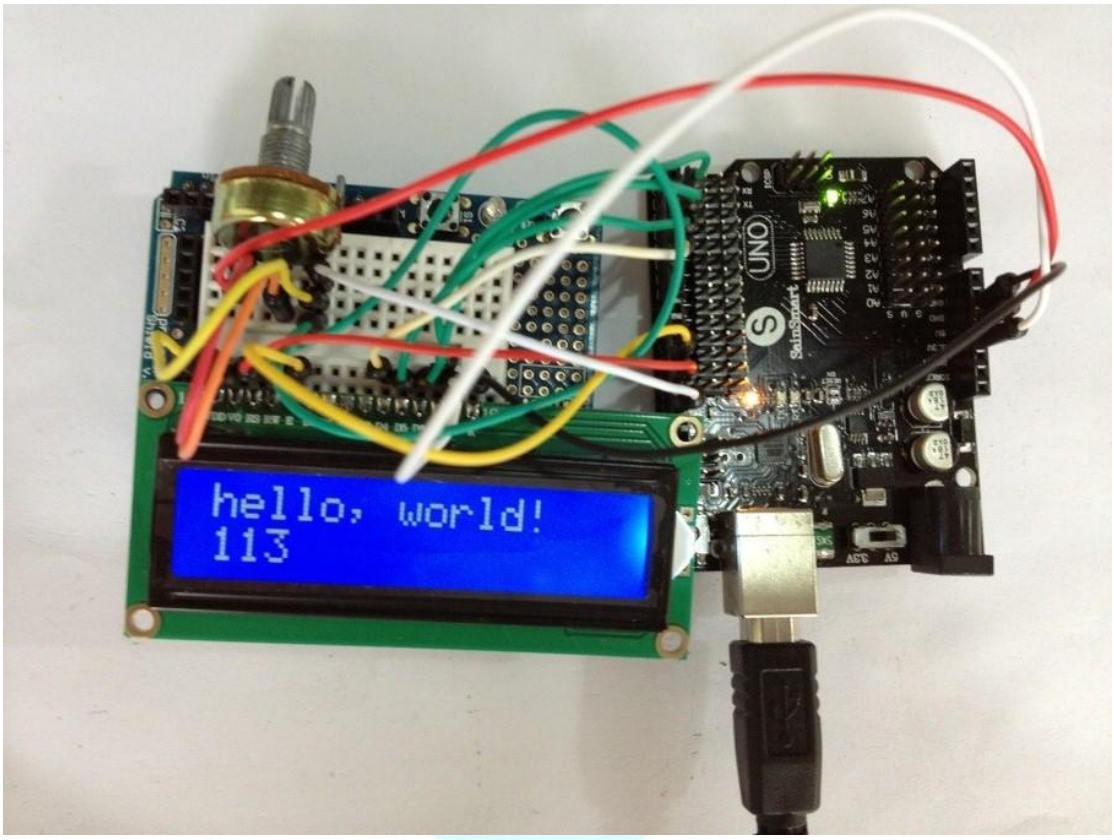
Chapter 15 1602LCD

Experiment component

- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram





Example code:

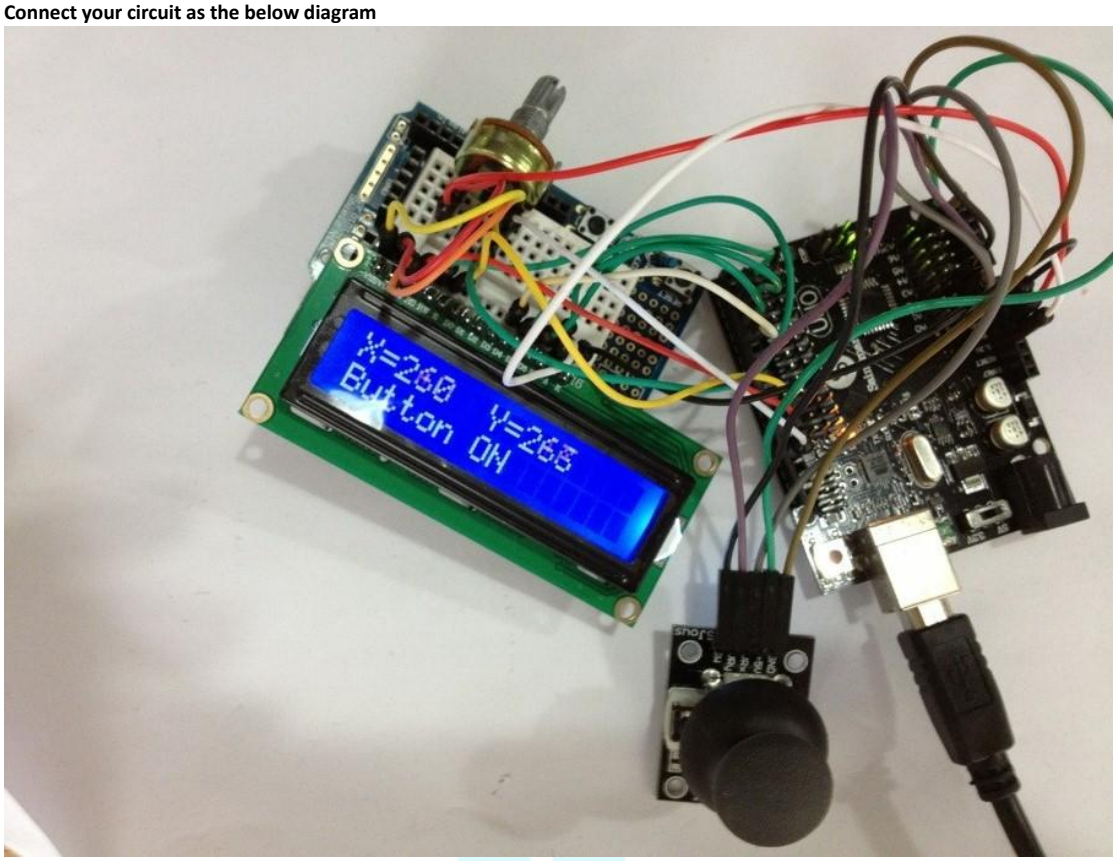
```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
   // set up the LCD's number of columns and rows:
   lcd.begin(16, 2);
   // Print a message to the LCD.
   lcd.print("hello, world!");
}

void loop() {
   // set the cursor to column 0, line 1
   // (note: line 1 is the second row, since counting begins with 0):
   lcd.setCursor(0, 1);
   // print the number of seconds since reset:
   lcd.print(millis()/1000);
}
```

Chapter 16 PS2 joystick

Experiment component

- SainSmart JoyStick Module : 1;
- 1602LCD : 1;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram



Example code:

```
#include <LiquidCrystal.h>      // Call comes Arduino library LiquidCrystal
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);//Set the interface
int xpotPin = 0;    // Set analog port 0 for the X signal input port
int ypotPin = 1;    // Set analog port 1 for the y signal input port
int bpotPin = 2;    // Set analog port 2 for the button signal input port
int xval=0;        // Set variable
int yval=0;
int bval=0;
void setup()
{
lcd.begin(16, 2);    // Initialize the LCD
delay(1000); //delay 1000ms
}
void loop ()
{
```

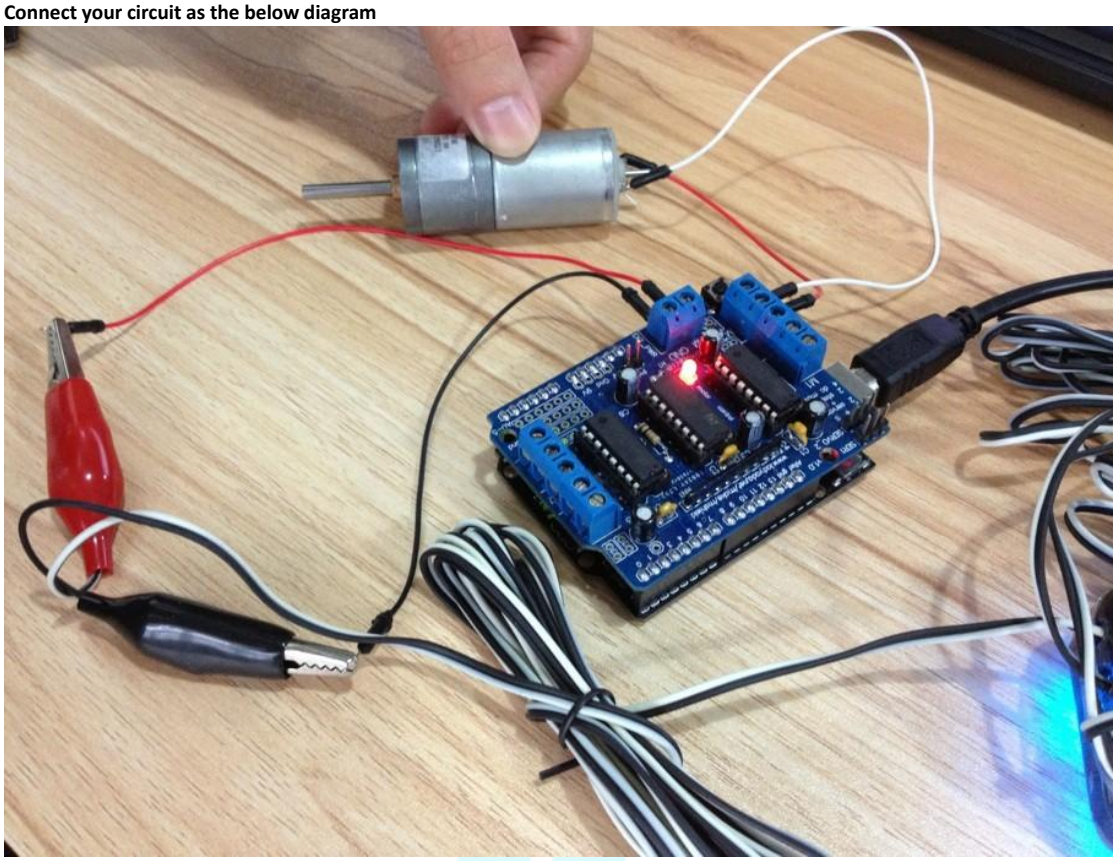
```
xval = analogRead(xpotPin);      //xval variable is from 0 to the value read from the signal port
yval = analogRead(ypotPin);      // yval variable is from 1 to the value read from the signal port
bval = analogRead(bpotPin);      //bval variable is from 2 to the value read from the signal port
lcd.clear(); //cls
lcd.setCursor(0, 0) ;      // Set the cursor position to the first position of the first row
lcd.print("X=");           // The screen display text X =
lcd.print(xval);
lcd.setCursor(7, 0) ;    // Set the cursor position to the eighth position of the first row
lcd.print("Y=");           // The screen display text Y =
lcd.print(yval);
if (bval<500)
{
lcd.setCursor(0, 1) ;      // Set the cursor position to the first position of the second line
lcd.print("Button ON"); // The screen to display text Button ON
}
else
{
lcd.setCursor(0, 1) ;
lcd.print("Button OFF"); // The screen to display text Button OFF
}
delay(100);      // Delay 0.1 seconds, here is the refresh rate


}
```

Chapter 17 Stepper motor

Experiment component

- SainSmart Stepper motor : 1;
- SainSmart L293D Motor Drive Shield : 1;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram



Example code:

```
// MakeShields Motor shield library
// copyright MakeShields, 2010
// this code is public domain, enjoy!

#include <MSMotorShield.h>
#include <Servo.h>

// DC motor on M2
MS_DCMotor motor(2);
// DC hobby servo
Servo servo1;
// Stepper motor on M3+M4 48 steps per revolution
MS_Stepper stepper(48, 2);

void setup() {
  Serial.begin(9600);           // set up Serial library at 9600 bps
  Serial.println("Motor party!");
```

```
  // turn on servo
  servo1.attach(9);

  // turn on motor #2
  motor.setSpeed(200);
  motor.run(RELEASE);
}

int i;

// Test the DC motor, stepper and servo ALL AT ONCE!
void loop() {
  motor.run(FORWARD);
  for (i=0; i<255; i++) {
    servo1.write(i);
    motor.setSpeed(i);
    stepper.step(1, FORWARD, INTERLEAVE);
    delay(3);
  }

  for (i=255; i!=0; i--) {
    servo1.write(i-255);
    motor.setSpeed(i);
    stepper.step(1, BACKWARD, INTERLEAVE);
    delay(3);
  }

  motor.run(BACKWARD);
  for (i=0; i<255; i++) {
    servo1.write(i);
    motor.setSpeed(i);
    delay(3);
    stepper.step(1, FORWARD, DOUBLE);
  }

  for (i=255; i!=0; i--) {
    servo1.write(i-255);
    motor.setSpeed(i);
    stepper.step(1, BACKWARD, DOUBLE);
    delay(3);
  }
}
```

Chapter 18 RGB module

Experiment component

- RGB module : 1;
- Bread board: 1;
- Jumper wires.

Example code:

```
int ledPin = 13; // LED is connected to digital pin 13
int redPin = 11;    // R petal on RGB LED module connected to digital pin 11
int greenPin = 9;    // G petal on RGB LED module connected to digital pin 9
int bluePin = 10;    // B petal on RGB LED module connected to digital pin 10

void setup()
{
        pinMode(ledPin, OUTPUT); // sets the ledPin to be an output
        pinMode(redPin, OUTPUT); // sets the redPin to be an output
        pinMode(greenPin, OUTPUT); // sets the greenPin to be an output
        pinMode(bluePin, OUTPUT); // sets the bluePin to be an output
}

void loop()    // run over and over again
{
         // Basic colors:
         color(255, 0, 0); // turn the RGB LED red
         delay(1000); // delay for 1 second
         color(0,255, 0); // turn the RGB LED green
         delay(1000); // delay for 1 second
         color(0, 0, 255); // turn the RGB LED blue
         delay(1000); // delay for 1 second

         // Example blended colors:
         color(255,255,0); // turn the RGB LED yellow
         delay(1000); // delay for 1 second
         color(255,255,255); // turn the RGB LED white
         delay(1000); // delay for 1 second
         color(128,0,255); // turn the RGB LED purple
         delay(1000); // delay for 1 second
         color(0,0,0); // turn the RGB LED off
         delay(1000); // delay for 1 second
}

void color (unsigned char red, unsigned char green, unsigned char blue)        // the color generating function
{
         analogWrite(redPin, 255-red);
         analogWrite(bluePin, 255-blue);
```

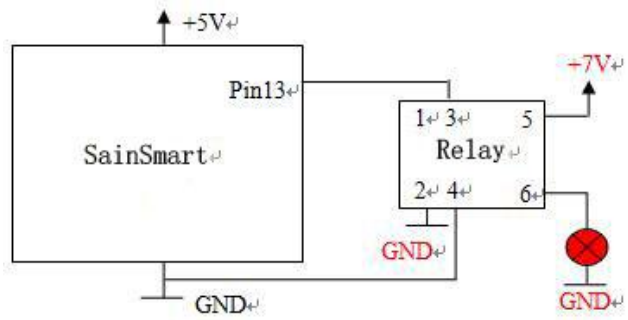
```
        analogWrite(greenPin, 255-green);
}
```

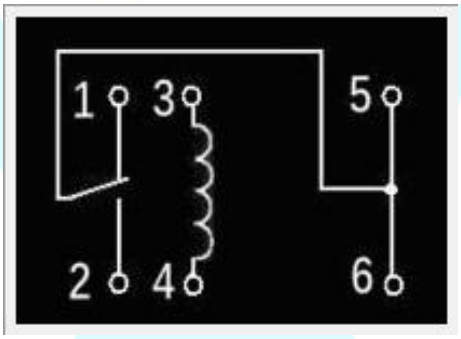
Chapter 19 Relay

Experiment component

- RGB module : 1;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram





Example code:

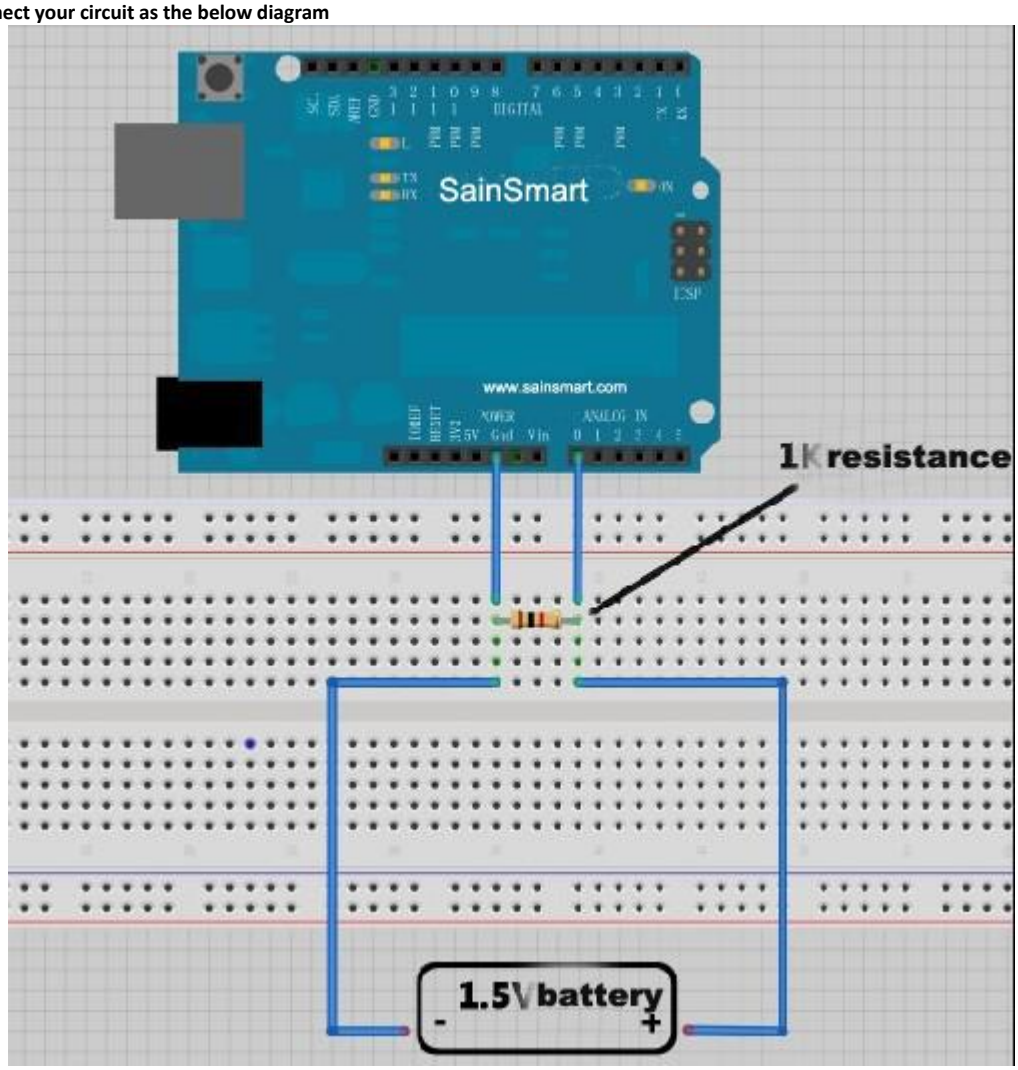
```
int jdqPin=13;
void setup()
{
   pinMode(jdqPin,OUTPUT);
   Serial.begin(9600);
}
void loop()
{
digitalWrite(jdqPin,HIGH);
delay(1000);
digitalWrite(jdqPin,LOW);
delay(1000);
}
```

Chapter 20 Voltmeter

Experiment component

- SainSmart Prototype shield : 1;
- 1kΩ Resistance : 1;
- Bread board: 1;
- Jumper wires.

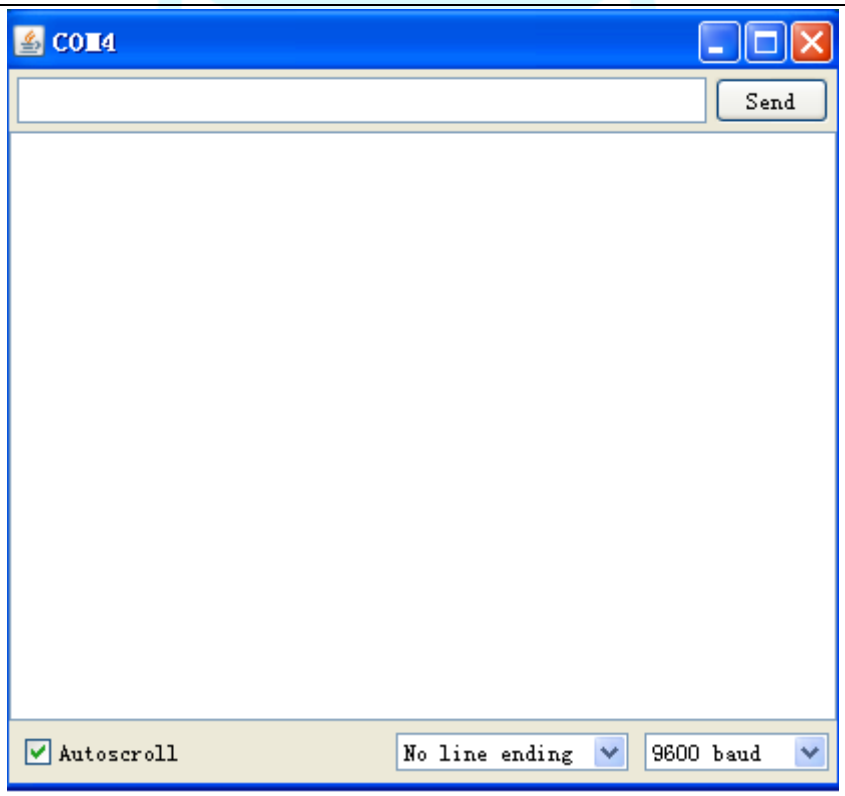
Connect your circuit as the below diagram



Example code:

```
float temp;     // Create a float variable temp as a storage space to store data preparation
void setup()
{
   Serial.begin(9600);          // Using 9600 baud rate serial communication
}
void loop()
{


   int V1 = analogRead(A0); // A0 port to read from the voltage data into newly created variable of type integer V1, analog port voltage measurement range is 0-5V returns a value of 0-1024
   float vol = V1*(5.0 / 1023.0); // We will be converted into the actual value of V1 voltage value into a float variable vol
```

```
  if (vol == temp)   // This part of the Classifying are used to filter duplicate data, only by the of times voltage values and the last of varying be conducted only after output
  {
    temp = vol;                  //After comparison, the ratio of this value into a variable temp for comparison
  }
  else
  {
    Serial.print(vol);                          // Serial output voltage value, and not a newline
    Serial.println(" V");                       // Serial output characters V, and line breaks
    temp = vol;
    delay(1000);                                // Output after the completion of the wait 1 seconds, is used to control the data refresh rate.
  }
}
```

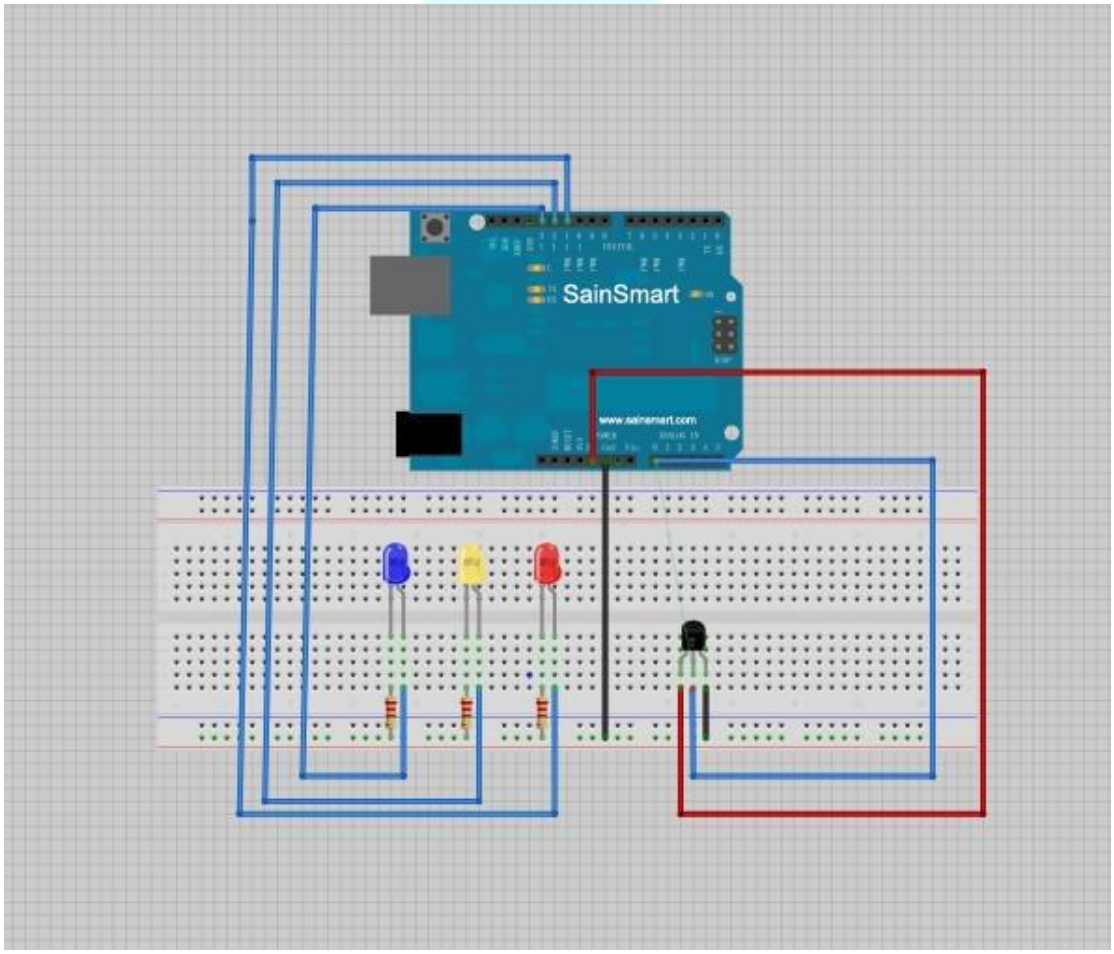


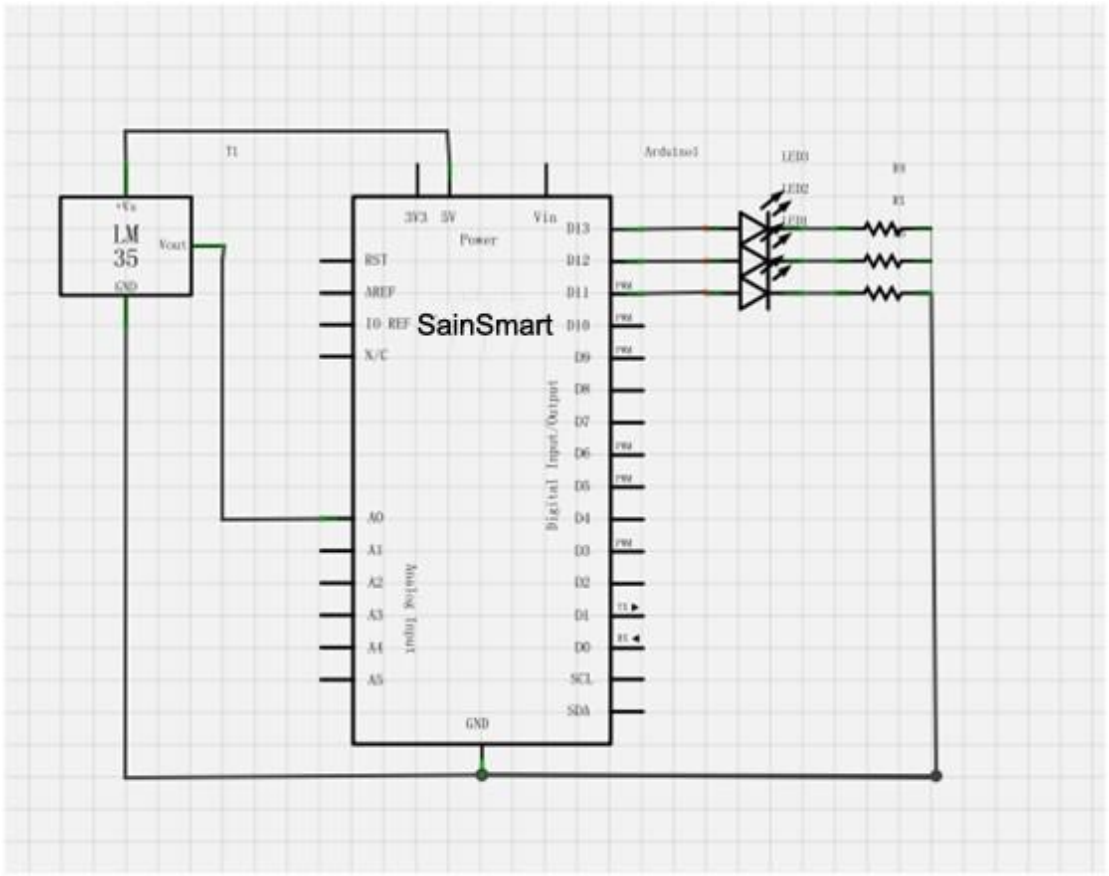
Chapter 21 Temperature Sensor

Experiment component

- LM35 Temperature sensor : 1;
- Red M5 LED : 1;
- Yellow M5 LED : 1;
- Blue M5 LED : 1;
- 220Ω Resistance : 3;
- Bread board: 1;
- Jumper wires.

Connect your circuit as the below diagram





Example code:

```
void setup() {
   pinMode(13, OUTPUT);
   pinMode(12, OUTPUT);
   pinMode(11, OUTPUT);
}
void loop() {
   int vol = analogRead(A0) * (5.0 / 1023.0*100);    //read LM35 temperature
if (vol<=31)                                          //Low temperature zone temperature Settings, as well as the led display
{
   digitalWrite(13, HIGH);
   digitalWrite(12, LOW);
   digitalWrite(11, LOW);
}
else if (vol>=32 && vol<=40)
  {
     digitalWrite(13, LOW);

   digitalWrite(12, HIGH);
   digitalWrite(11, LOW);
}
```

```
else if (vol>=41)                                    //High temperature setting
{
     digitalWrite(13, LOW);
   digitalWrite(12, LOW);
   digitalWrite(11, HIGH);
}
 }
```