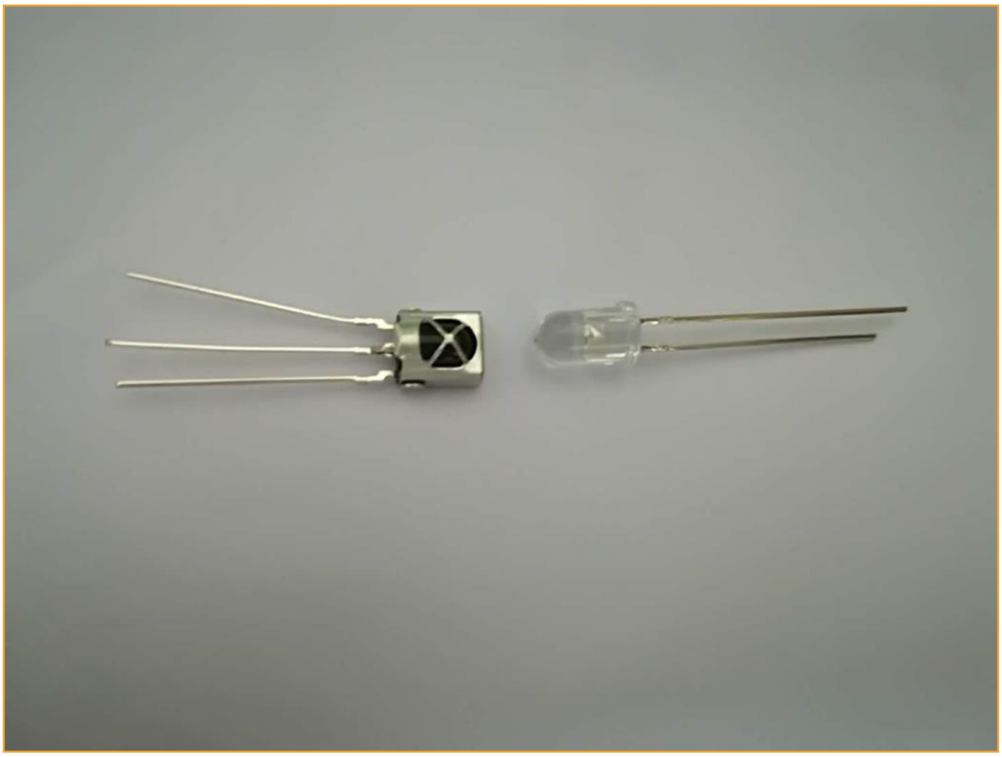
Infrared emission and infrared receiver circuit

Introduction

The infrared emission tube will be introduced this time. In fact, it is widely seen in our daily life. Now such a devices are widely used in many household appliances, such as air conditioning, television, DVD and so on, which is based on wireless remote sensing, but also a remote control, it is necessary to study its principle and how to use.

Infrared emitting diodes (IR LEDs), also known as infrared emitting diodes, belong to the class of diodes. It is a light emitting device that can directly convert electrical energy into near-infrared light (invisible light) and can radiate it out. It is mainly used in various photoelectric switches, touch screens, and remote control transmission circuits.



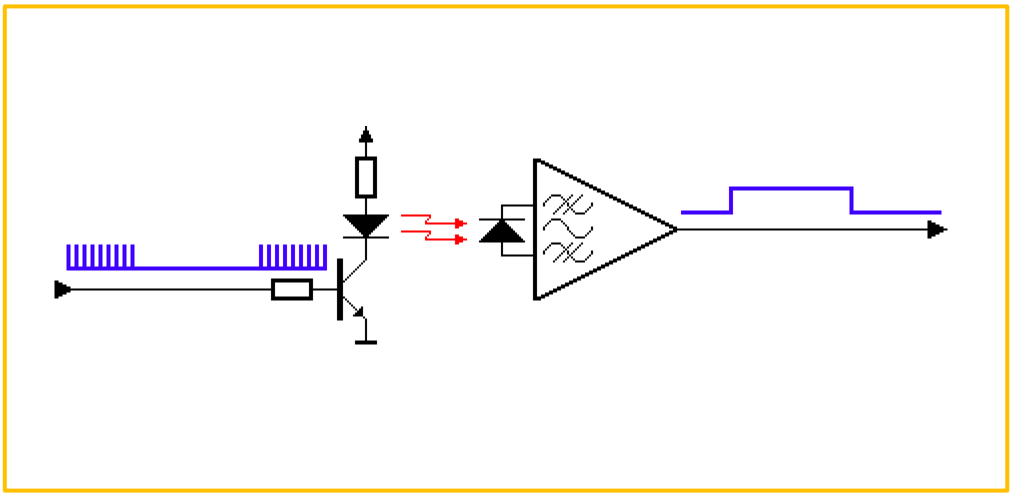
Features

Commonly used infrared light-emitting diodes (such as SE303, PH303), its shape and light-emitting diode LED similar, emit infrared light. The tube pressure drop is about 1.4v, and the working current is generally less than 20mA. In order to adapt to different operating voltages, a current limiting resistor is often used in the loop.

When infrared is emitted to control the corresponding controlled device, the distance controlled by it is proportional to the transmitted power.

In order to increase the infrared control distance, the infrared light emitting diode operates in a pulse state because the effective transmission distance of the pulsed light (modulated light) is proportional to the peak current of the pulse. Simply increasing the peak Ip as much as possible can increase the infrared light emission distance.

In this experiment, we want to use two Arduino motherboards, one infrared emitter tube and one integrated receiver.After wiring the circuit, we can do the test.



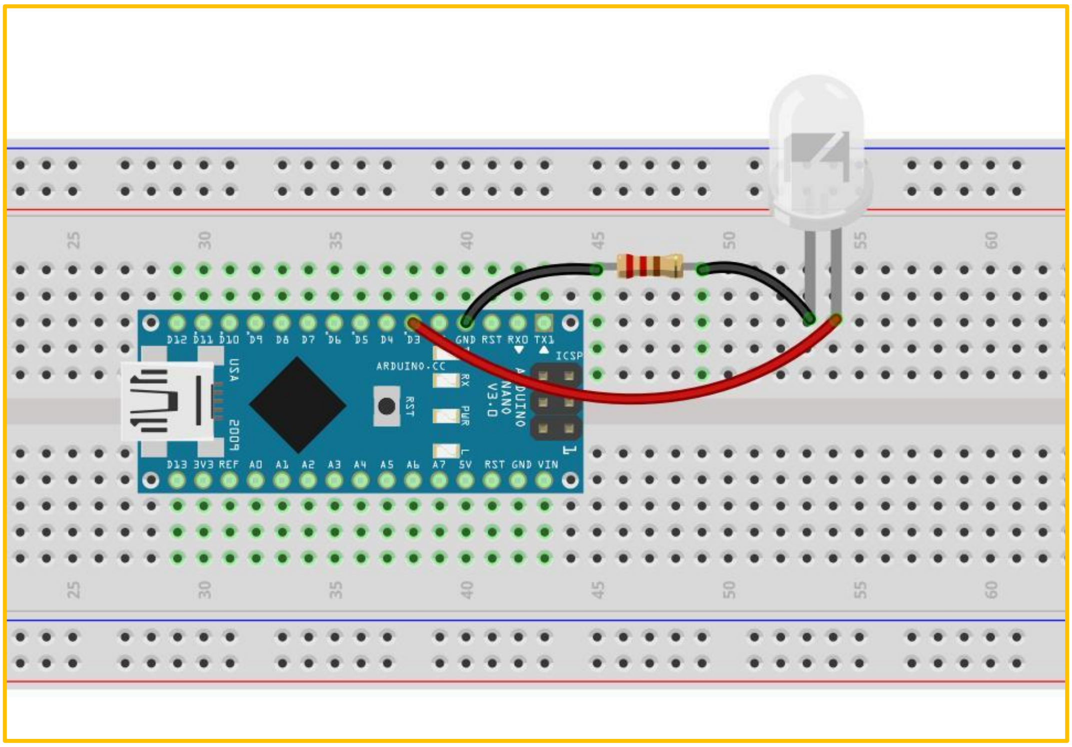
Component List

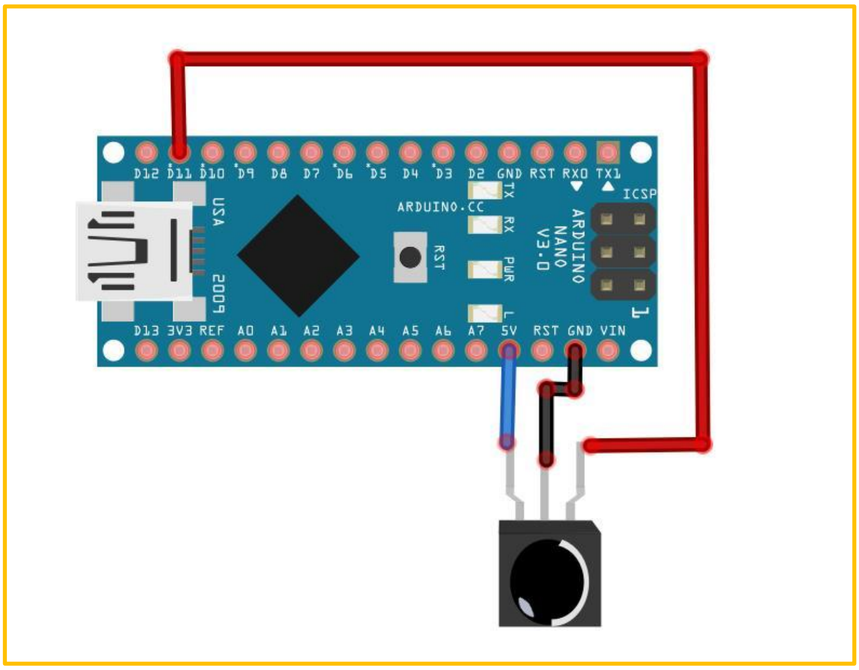
- Keywish Arduino Nano mainboard*2
- USB cable*2
- Infrared emission tube*1
- 220 Ω resistor*1
- Infrared receiver tube *1
- Some wires

Wiring of Circuit

Arduino Nano	transmitter
D3	S
GND	-

Arduino Nano	receiver
D11	Left
GND	Middle
+5V	Right


25
30
35
40
45
50
55
60
D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 GND RST RX0 TX1
ARDUINO.CC
ICSP
ARDUINO NANO V3.0
USA
2009
RST
RX
PWR
L
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN


D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
ICSP
ARDUINO.CC
TX
RX
PWR
L
ARDUINO NANO V3.0
USA
2009
RST
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN

Code

Transmitter:

```
#define ADD 0x00
int TRAM_PIN = 3; //define input pin on Arduino
void setup() {
   Serial.begin(9600);
   pinMode(TRAM_PIN, OUTPUT);
}
void loop() {
   uint8_t dat, temp;
   if ( Serial.available()) {
      temp = Serial.read();
      IR_Send38KHZ(280, 1);//Send 9ms start code
      IR_Send38KHZ(140, 0);//Send a 4.5ms result code

      IR_Sendcode(ADD);//User ID
      dat = ~ADD;
      IR_Sendcode(dat);//User ID reverse code

      IR_Sendcode(temp);//Operation code
      dat = ~temp;
      IR_Sendcode(dat);//Opcode inversion

      IR_Send38KHZ(21, 1);//Send end code
   }
   delay(200);
}
void IR_Send38KHZ(int x, int y) { //Generates 38KHZ infrared pulses
   for (int i = 0; i < x; i++) { //15=386US
      if( y == 1){
         digitalWrite(TRAM_PIN, 1);
         delayMicroseconds(9);
         digitalWrite(TRAM_PIN, 0);
         delayMicroseconds(9);
      }
      else {
         digitalWrite(TRAM_PIN, 0);
```

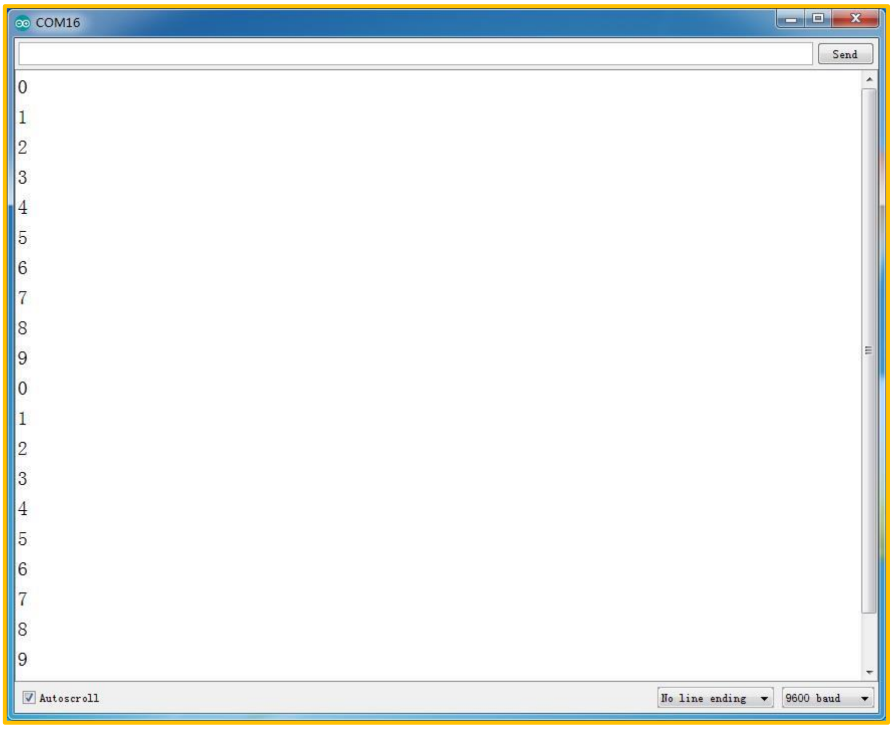
```
          delayMicroseconds(20);
       }
    }
}
void IR_Sendcode(uint8_t x) {
   for (int i = 0; i < 8; i++) {
      if ( (x&0x01) == 0x01) {
         IR_Send38KHZ(23, 1);
         IR_Send38KHZ(64, 0);
       }
       else {
          IR_Send38KHZ(23, 1);
          IR_Send38KHZ(21, 0);
       }
   x = x >> 1;
   }
}
```

Receiver:

```
#include <IRremote.h>
int RECV_PIN = 11; //define input pin on Arduino
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup() {
   Serial.begin(9600);
   irrecv.enableIRIn(); // Start the receiver
}
void loop() {
   if ( irrecv.decode( &results)) {
      if ( results.value == 16747635) {
         Serial.println("1");  //Received the number 1. Afterwards and so on
      }
      else if ( results.value == 16731315) {
         Serial.println("2");
      }
      else if ( results.value == 16763955) {
         Serial.println("3");
      }
      else if ( results.value == 16723155) {
         Serial.println("4");
      }
      else if ( results.value == 16755795) {
         Serial.println("5");
      }
      else if ( results.value == 16739475) {
         Serial.println("6");
      }
      else if ( results.value == 16772115) {
         Serial.println("7");
      }
      else if ( results.value == 16719075) {
         Serial.println("8");
      }
      else if ( results.value == 16751715) {
         Serial.println("9");
      }
      else if ( results.value == 16714995) {
         Serial.println("0");
      }
      irrecv.resume(); // Receive the next value
   }
}
```

Experiment Result

This time we use two Arduino mainboards, when downloading the codes, you have to know which is the transmitter code and which is receiver code. If the download is wrong, there will be no result. After downloading the codes, we open the Serial Monitor window, if you can see the following displayed data, it shows that you succeed.



```
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9
```