Lesson 4 Infrared Controlling Car

Main points

Infrared remote control is a widely used remote control method.The car is equipped with an infrared receiver and therefore allows it to be controlled by an infrared remote control.

First, Learning:

1. Learn the working principle of infrared control.

2. Learn how to use an infrared controller and an infrared receiver

Second, Preparations:

Smart car

USB cable

Infrared receiver and infrared remote controller

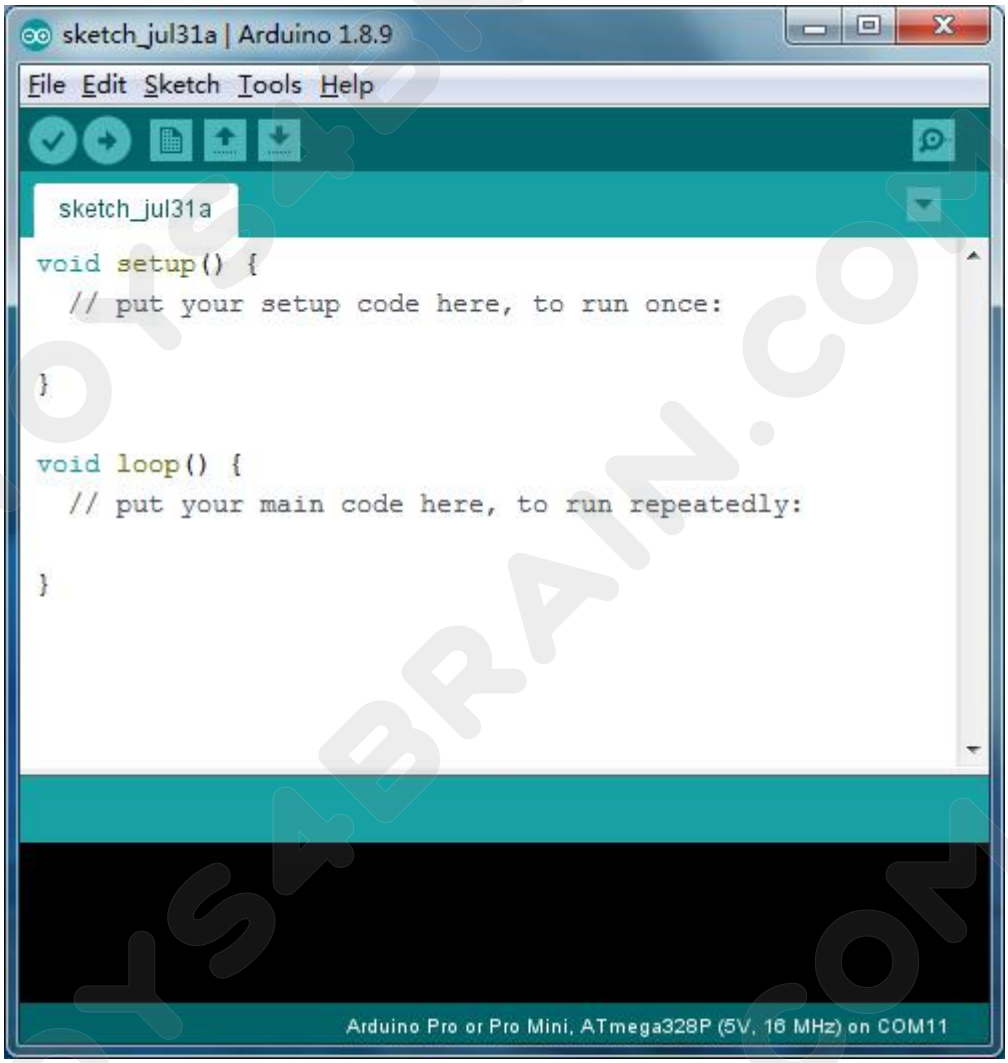
1、 IR Receiving Module and IR Remote

IR Receiving Module

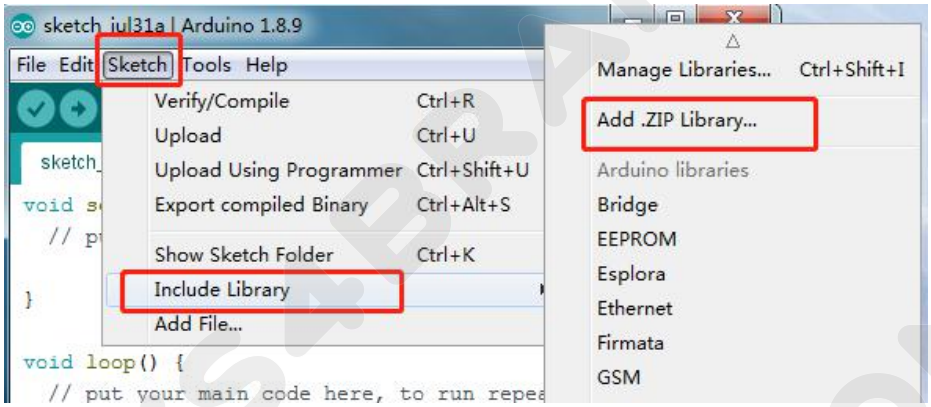
This is IR remote:

2. Program test

3. Because in this program, we need to use the library, so we need to add the library file first.Connect Arduino Nano to computer and turn on Arduino IDE.



Click Sketch --> Include Library --> Add .ZIP Library... --> then select the library as blow



The file name of the ZIP library must beIRremote.zip。

Open zip file of tutorial materials: Lesson 4 Car Infrared Remote Control Mode-Infrared_remote_control_car——IRremote. And add it.

Then open the code file in the path"\Lesson 4infrared remote control car\ infrared_Blink \ infrared_Blink.ino" and upload the program to the controller board。

Code preview :

```
#include <IRremote.h>

//------- IR REMOTE CODES ---------//
#define FORWARD 16718055
#define BACK    16730805
#define LEFT    16734885
#define RIGHT   16716015
#define STOP    16726215
#define Buzzer  16738455

#define RECV_PIN  12

#define ENB 5    // Left  wheel speed
#define IN1 7    // Left  wheel forward
#define IN2 8    // Left  wheel reverse
#define IN3 9    // Right wheel reverse
#define IN4 11   // Right wheel forward
#define ENA 6    // Right wheel speed
#define carSpeed 200   // initial speed of car >=0 to <=255
int sound = 15;
int left_LED = 16;
int Right_LED = 17;

IRrecv irrecv(RECV_PIN);
decode_results results;
unsigned long val;
unsigned long preMillis;

void forward() {
  digitalWrite(ENA, HIGH);
  digitalWrite(ENB, HIGH);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  digitalWrite(left_LED,LOW);
```

```
    digitalWrite(Right_LED,LOW);
    Serial.println("go forward!");
}

void back(){
    digitalWrite(ENA, HIGH);
    digitalWrite(ENB, HIGH);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    digitalWrite(left_LED,HIGH);
    digitalWrite(Right_LED,HIGH);
    Serial.println("go back!");
}

void left(){
    analogWrite(ENA, carSpeed);
    analogWrite(ENB, carSpeed);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    digitalWrite(left_LED,LOW);
    digitalWrite(Right_LED,HIGH);
    Serial.println("go left!");
}

void right(){
    analogWrite(ENA, carSpeed);
    analogWrite(ENB, carSpeed);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    digitalWrite(left_LED,HIGH);
    digitalWrite(Right_LED,LOW);
    Serial.println("go right!");
}

void stop(){
    digitalWrite(ENA, LOW);
    digitalWrite(ENB, LOW);
    Serial.println("STOP!");
```

```
}


/************BLINKSOUND****/
void blinks()
{
  for(int i=0;i<5;i++)
  {
    digitalWrite(left_LED,HIGH);
    digitalWrite(Right_LED,HIGH);
    digitalWrite(sound,HIGH);
    delay(500);
    digitalWrite(left_LED,LOW);
    digitalWrite(Right_LED,LOW);
    digitalWrite(sound,LOW);
    delay(500);
  Serial.println("Blink!");
}
delay(1000);
   }
////////////////////////////////
void setup() {
   Serial.begin(9600);
   pinMode(IN1, OUTPUT);
   pinMode(IN2, OUTPUT);
   pinMode(IN3, OUTPUT);
   pinMode(IN4, OUTPUT);
   pinMode(ENA, OUTPUT);
   pinMode(ENB, OUTPUT);
   pinMode(left_LED,OUTPUT);
   pinMode(Right_LED,OUTPUT);
   pinMode(sound,OUTPUT);
   stop();
   irrecv.enableIRIn();
}
////////////////////////////////

void loop() {
   if (irrecv.decode(&results)){
      preMillis = millis();
      val = results.value;
      Serial.println(val);
      irrecv.resume();
      switch(val){
```

```
            case FORWARD: forward();    break;
            case BACK:      back();         break;
            case LEFT:      left();         break;
            case RIGHT:     right();        break;
            case STOP:      stop();         break;
            case Buzzer:    blinks();       break;
            default:                        break;
        }
    }
    else{
        if(millis() - preMillis > 500){
            stop();
            preMillis = millis();
        }
    }
}
```

After disconnecting the car and the computer, you can turn on the power switch and put the cart on the ground。

Press the "*" button and observe the car, and you will hear an alarm making by the buzzer of the car .

Introduction of principle：

1. Working principle

The universal infrared remote controlling system consists of two parts: sending and receiving, the sending part consists of an IR remote controller, the receiving part consists of an infrared receiving tube. The signals that sent by IR remote controlling is a serial of binary pulse code. In order to be free from distraction of other infrared signals during wireless transportation, it's general to modulate it at given carrier frequency, and then launch it through infrared emitted phototransistor. Infrared receiving tube filters out other noise waves, only receives signals of given frequency and restores them to binary pulse code that is demodulation. Built-in receiving tube transform light signals that are sent from infrared light-emitting diode to weak electric signals, signals are enlarged through amplifier inside IC, and through automatic gain controlling, band-pass filtering, demodulation, wave shaping and be restored to original encoding sent by remote control, recognize the circuit by coding that is input to electric appliance through signal output pin of infrared receiving module.

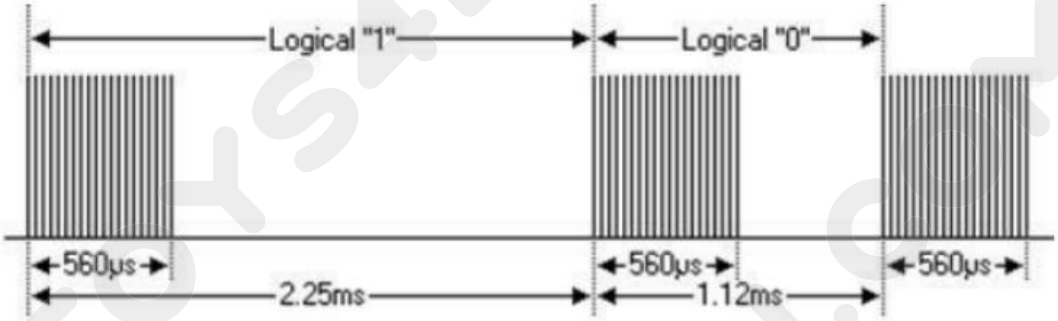
2、 Protocol of infrared remote controlling

The coding scheme of matched IR remote controlling is: NEC protocol.

Next, let's learn what NEC protocol is.

Features:

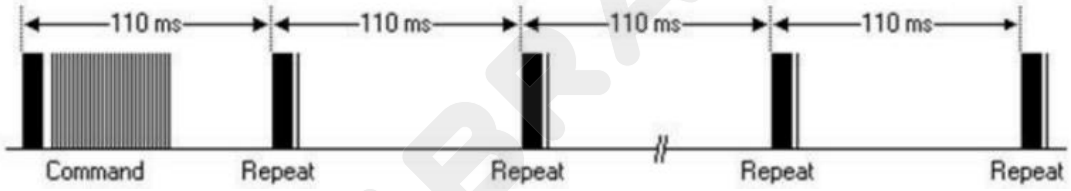
（1）8 address bit, 8 order bit
（2）Address bit and order bit are transmitted twice in order to guarantee reliability
（3）Pulse position modulation
（4）Carrier frequency is 38kHz
（5）Time of every bit is 1.125ms or 2.25ms
Definitions of logical 0 and 1 are as below:



The protocol is as below:
Press instant loosen transmission pulse:

Note: This is protocol of sending LSB (least-significant bit) firstly. Transport address of the above pulse is 0x59, order is 0x16. One message starts from a high level of 9ms, the follow is a low level of 4.5ms, (Two level forms guidance code) and through address code and order code. Address and order are transmitted twice. In the second time, all bits are inverted the opposite, can be used to confirm the receiving messages to be used. Total sending time is fixed, if you are not interested in it, you can ignore reliability of invert, and can expand address and order at 16 bit! Because the fact that length repeat if every bit is opposite.

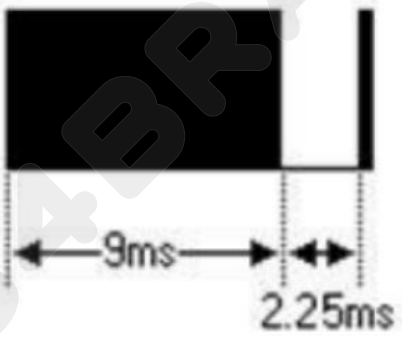
Press transmitted pulse loosened after a time.



Once a command was sent, even if the button of remote controlling is pressed.
When button is still pressed, the pulse of first 110ms is different from above,

duplicated code is transmitted after every 110ms. Duplicated code is made up of a high level pulse of 9ms and a low level of 2.25 and a high level of 560μs.

Repeat pulse：



Note: After impulse waveform enters into integration of sensor, owing to the fact that integration of sensor should be decoded, signal magnified and plastic, you should note the time when there are no infrared signals, its output terminal is high level or low level when there are signals. So the level of output signal is opposite to transmitting terminal. Everybody can see receiver pulse through oscilloscope, understand program with wave form seen.

3. The idea of programming remote control car

According to the characteristic of NEC code and wave of receiving-end, this experiment divides wave of receiving-end into four parts: leading code (Pulse of 9ms and 4.5ms)、 address code (including 8-bit address code and 8-bit address fetch)、 16-bit address code (including 8-bit address code and 8-bit address fetch)、 16-bit order code(including 8-bit order code and 8-bit order fetch)、repeat code(be made up of pulse of 9ms、2.25ms、560us).

Exploit the timer to test high level and low level of wave received, being distinguished according to the time tested: logical“01”、 logical”1”、leading pulse、repeat pulse. Leading code and address code are judged whether correct, not be stored, owing to the fact that order code of each key is different, action is carried out by order code.During car experiment, we just need to control the car to go forward and backward ,turn left and right, and stop ,which means we would need 5 keys and the value of them are as below:

Remote control character key value