Lesson 3 Infrared Controlling Car

The points of section

Infrared remote control is a widely used method for remote control.

The car has been equipped with infrared receiver and thus allows it to be controlled using the infrared remote controller.

Learning parts:

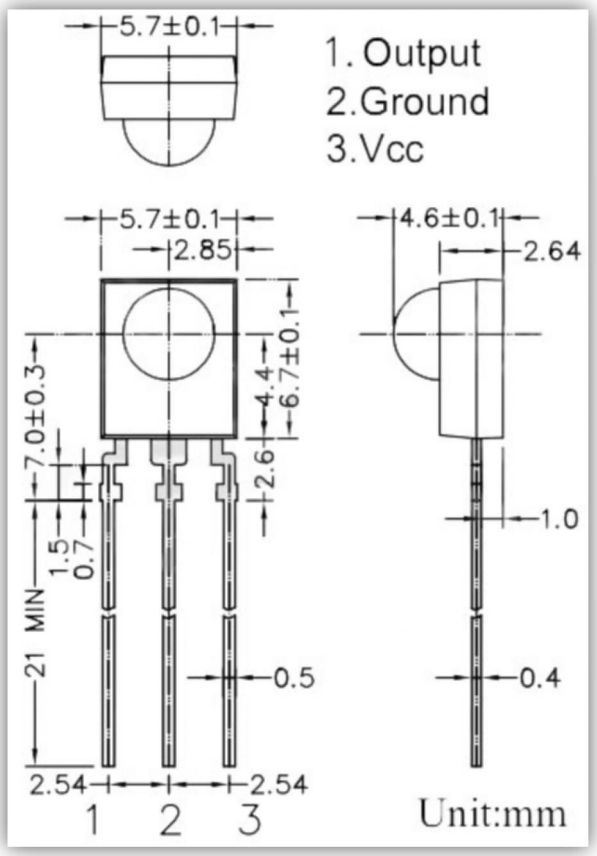
- *Understand the infrared remote controller and the receiver*
- *Understand the remote control principles*

Preparations:

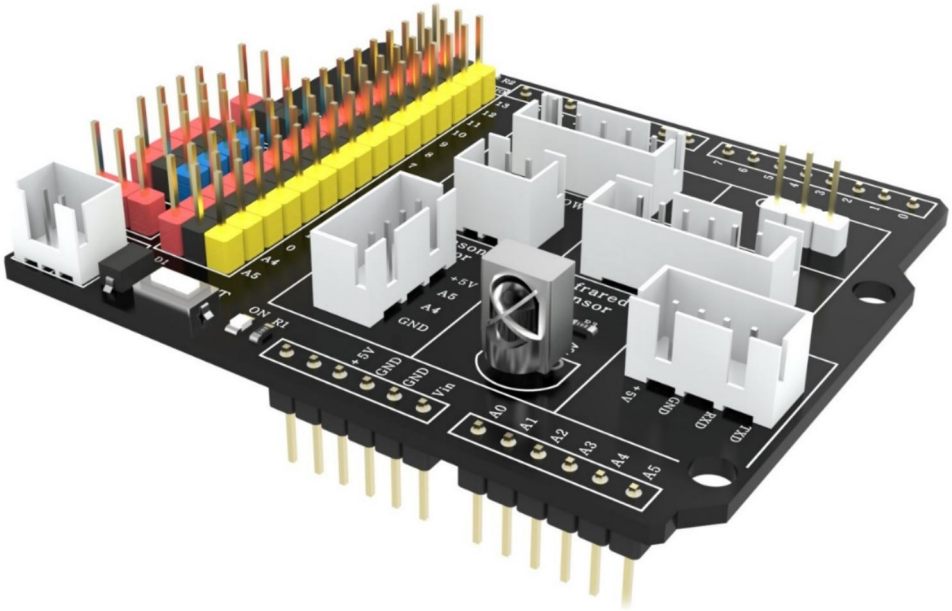
- *A car (with battery)*
- *A USB cable*
- *IR receiving module and IR remote*

Ⅰ. IR Receiving Module and IR Remote

The data of IR receiver sensor is as below:



The connection of receiver module is as below:

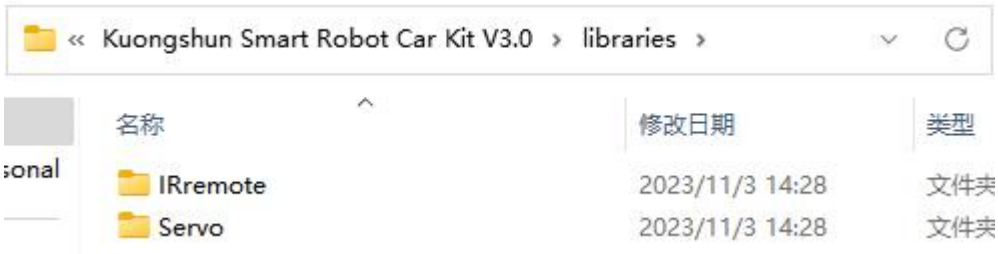


This is IR remote:

Ⅱ. Testing program

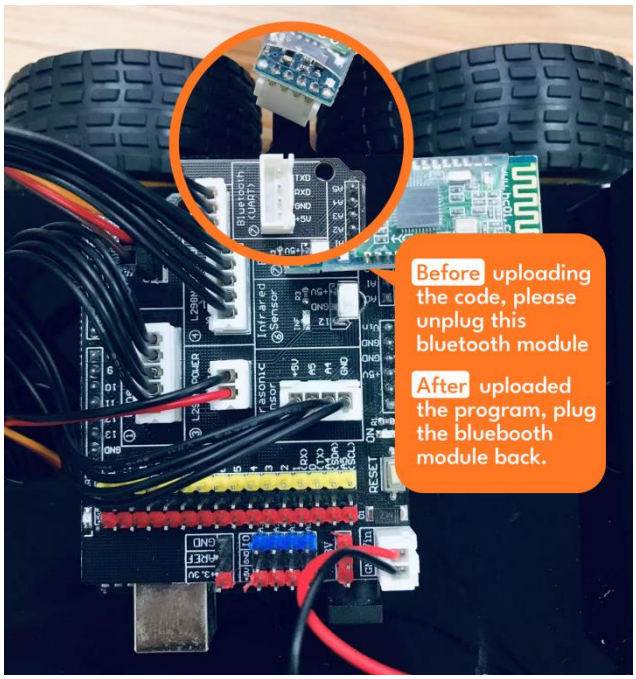
Because in this program, we need to use the library so that we need to add library file at first.

Find this folder IRremote in the following path and copy it (copying both folders allows you to skip this operation in the next lesson)

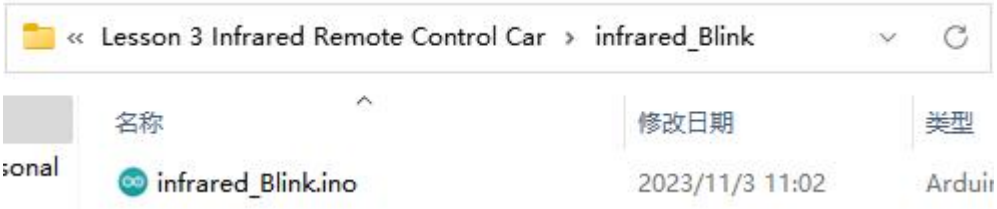


Find the following path and paste the IRremote folder you just copied into this path

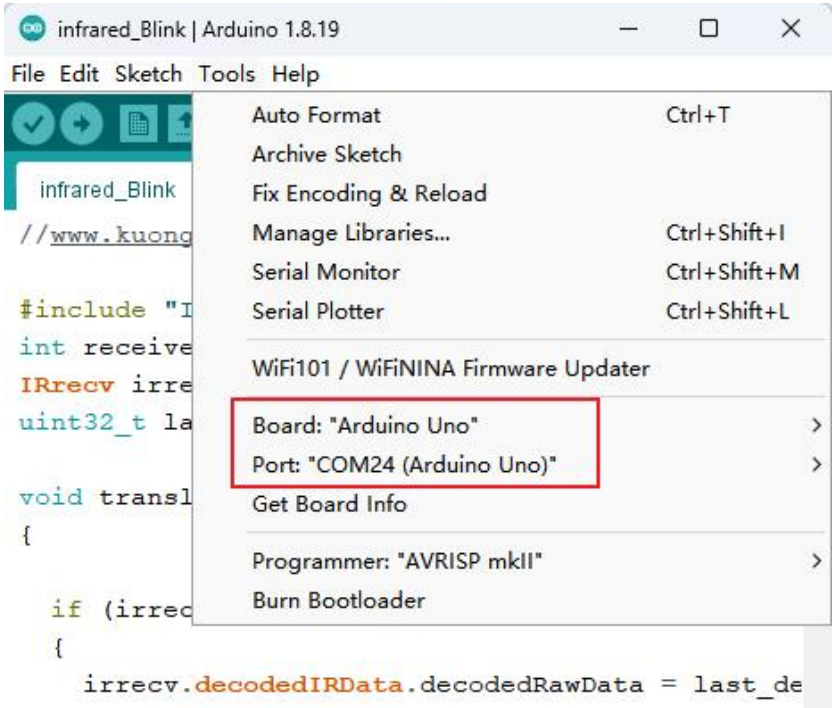
Connect the cart to the computer via the data cable and unplug the Bluetooth module, if you have already unplugged it, please put the Bluetooth module away and plug it back in after uploading the program in case you lose it!



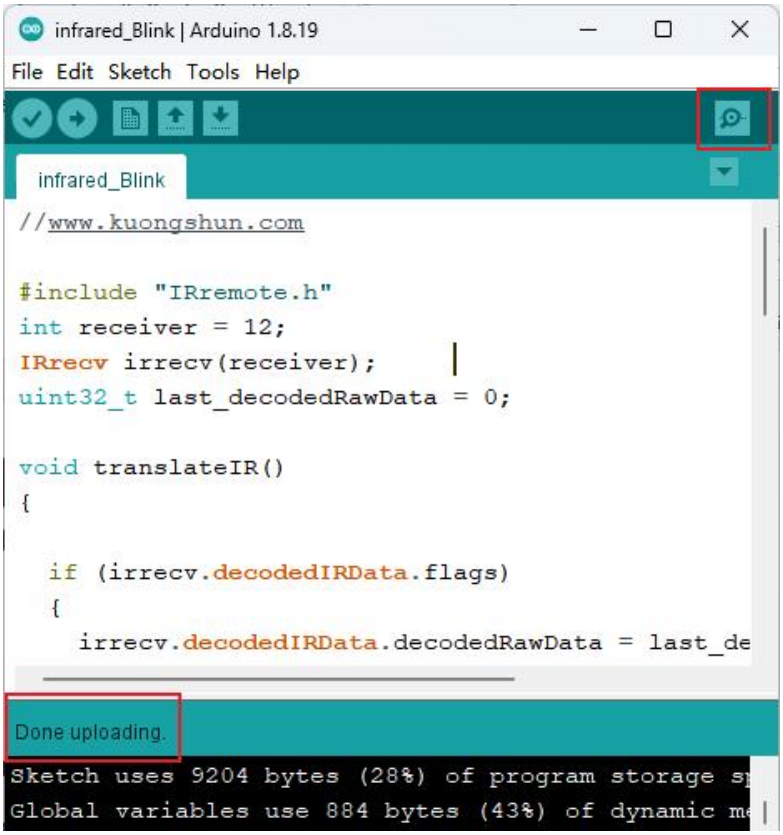
Open the infrared test program at the following path



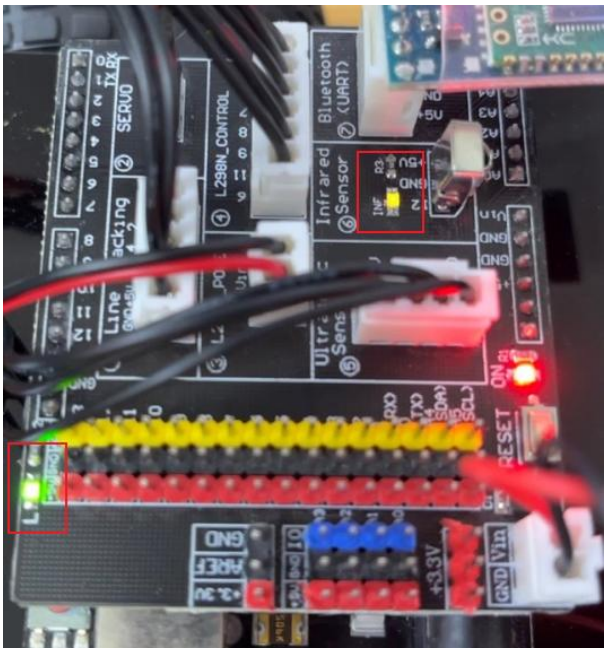
After confirming that the board and port are Arduino UNO, click the Upload button to upload the program.



Appears to upload successfully can verify the infrared function, click on the upper right corner of the serial port monitor or look at the trolley expansion board can see the results

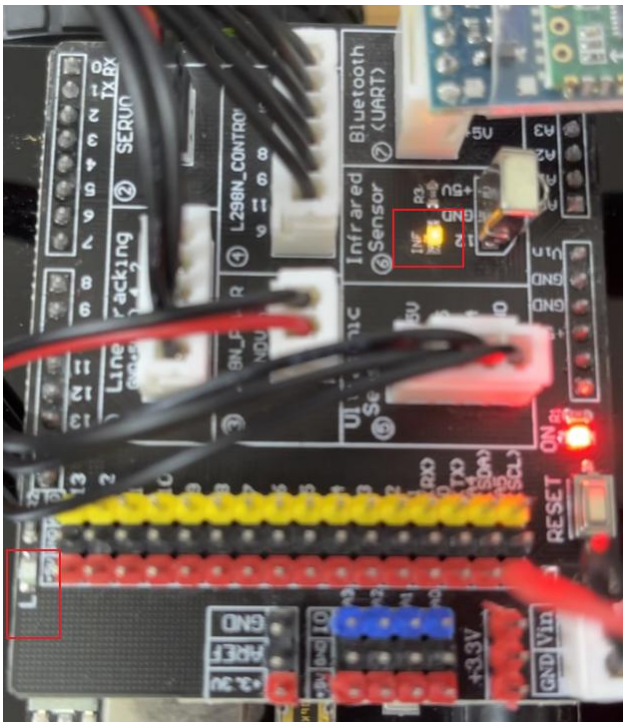


When 1 is pressed, the lamp on the side of the infrared receiver head will blink, and the lamp on pin 13 in the lower left corner of the figure will light up

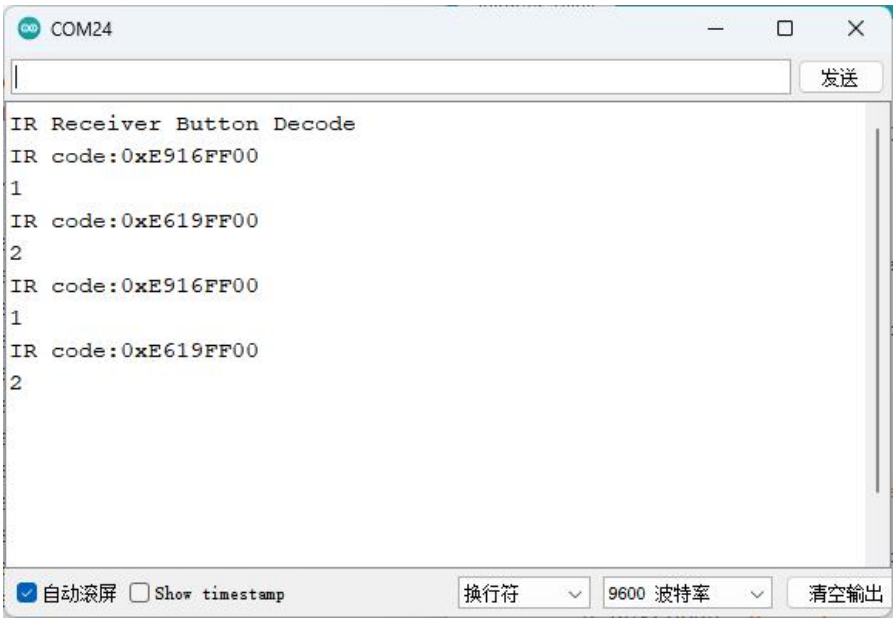


When 2 is pressed, the light on the side of the IR receiver head will blink and the pin 13 light in the

lower left corner of the figure will go out



Open the serial monitor to view, we can see that we pressed the 1 and 2 and their infrared code is printed on the screen.(This infrared data is more suitable for single click, try not to long press, long press will appear "REPEAT!" and recognition is not successful)



Code preview :

```
//www.kuongshun.com

#include "IRremote.h"
int receiver = 12;
IRrecv irrecv(receiver);
uint32_t last_decodedRawData = 0;

void translateIR()
{

  if (irrecv.decodedIRData.flags)
  {
    irrecv.decodedIRData.decodedRawData = last_decodedRawData;
    Serial.println("REPEAT!");
  } else
  {
    Serial.print("IR code:0x");
    Serial.println(irrecv.decodedIRData.decodedRawData, HEX);
  }
  switch (irrecv.decodedIRData.decodedRawData)
  {
    case 0xB946FF00: Serial.println("UP"); break;
    case 0xEA15FF00: Serial.println("DOWN"); break;
    case 0xBB44FF00: Serial.println("LEFT"); break;
    case 0xBC43FF00: Serial.println("RIGHT");    break;
    case 0xBF40FF00: Serial.println("OK");    break;
    case 0xAD52FF00: Serial.println("0");    break;
    case 0xE916FF00: Serial.println("1");digitalWrite(13, HIGH);break;
    case 0xE619FF00: Serial.println("2");digitalWrite(13, LOW);break;
    case 0xF20DFF00: Serial.println("3");    break;
    case 0xF30CFF00: Serial.println("4");    break;
    case 0xE718FF00: Serial.println("5");    break;
    case 0xA15EFF00: Serial.println("6");    break;
    case 0xF708FF00: Serial.println("7");    break;
    case 0xE31CFF00: Serial.println("8");    break;
    case 0xA55AFF00: Serial.println("9");    break;
    case 0xBD42FF00: Serial.println("*");    break;
    case 0xB54AFF00: Serial.println("#");    break;
    default:
      Serial.println(" other button   ");
  }
```

```
  last_decodedRawData = irrecv.decodedIRData.decodedRawData;
  delay(100);
}

void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  Serial.println("IR Receiver Button Decode");
  irrecv.enableIRIn();
}

void loop()
{
  if (irrecv.decode())
  {
    translateIR();
    irrecv.resume();
  }
}
```

Ⅲ. 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

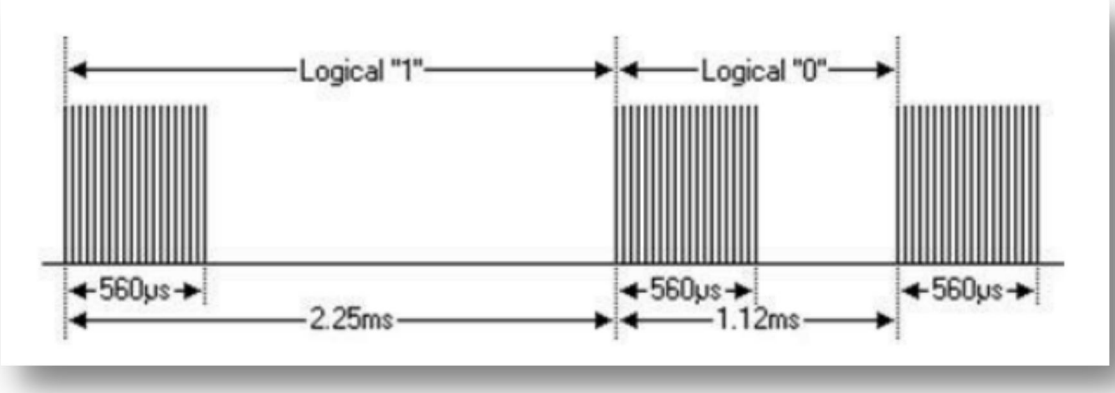
（2）Address bit and order bit are transmitted twice in order to guarantee reliability

（3）Pulse position modulation

（4）Carrier frequency is 38kHz

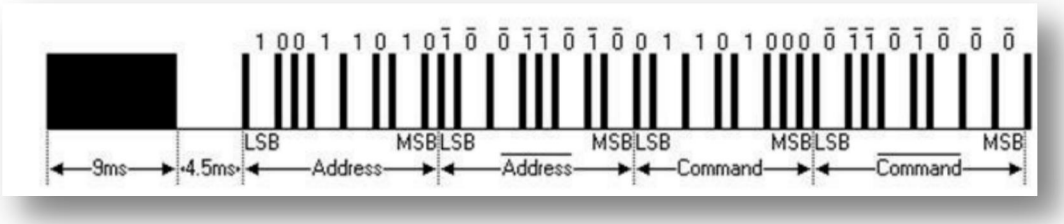
（5）Time of every bit is 1.125ms or 2.25ms

Definitions of logical 0 and 1 are as below:



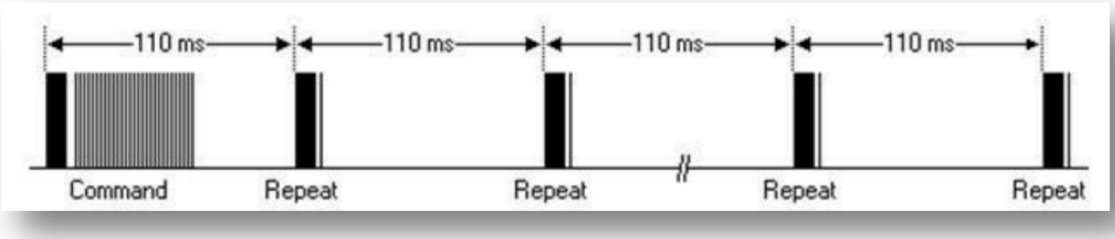
The protocol is as below:

Press instant loosen transmission pulse:



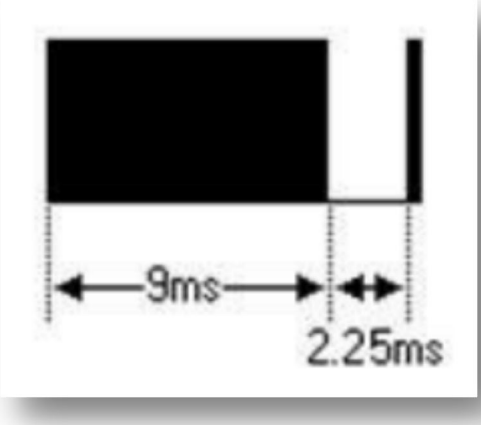
Note: This is protocol of sending LSB (least-significant bit) firstly. Transport address of the above pulse is 0x59, order is 0x16. One message starts from a high level of 9ms, the follow is a low level of 4.5ms, (Two level forms guidance code) and through address code and order code. Address and order 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, is 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

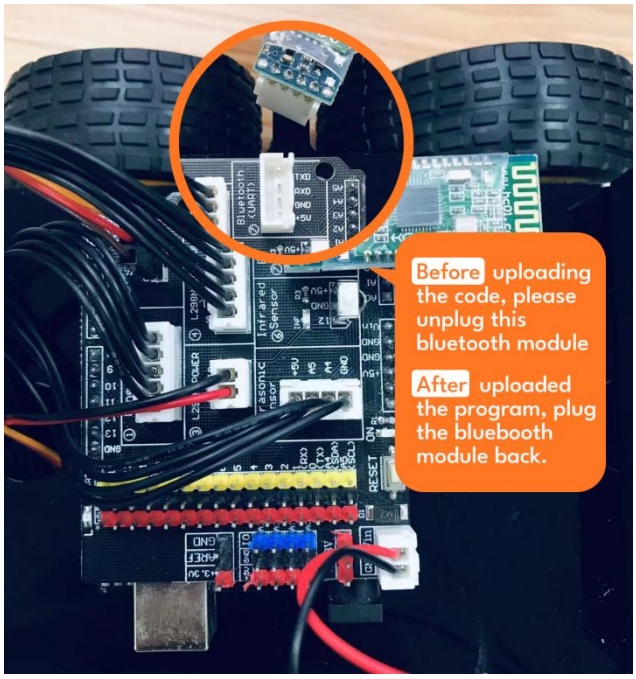
According to the characteristic of NEC code and wave of receiving-end, this experiment divides wave of receiving-end into four parts: leading code (Pulse of 9ms and 4.5ms)、address code (including 8-bit address code and 8-bit address fetch)、
16-bit address code (including 8-bit address code and 8-bit 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
Middle red button	16712445, 3622325019
Above triangle	16736925, 5316027
Below triangle	16754775, 2747854299
Left triangle	16720605, 1386468383
Right triangle	16761405, 553536955

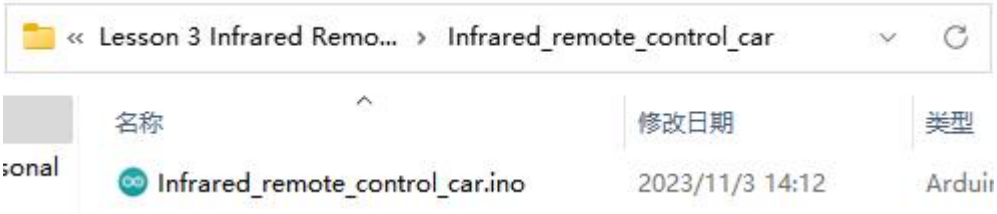
IV. Make a remote controlling car

To make a remote control cart we need to swipe the complete infrared control cart program into the car.

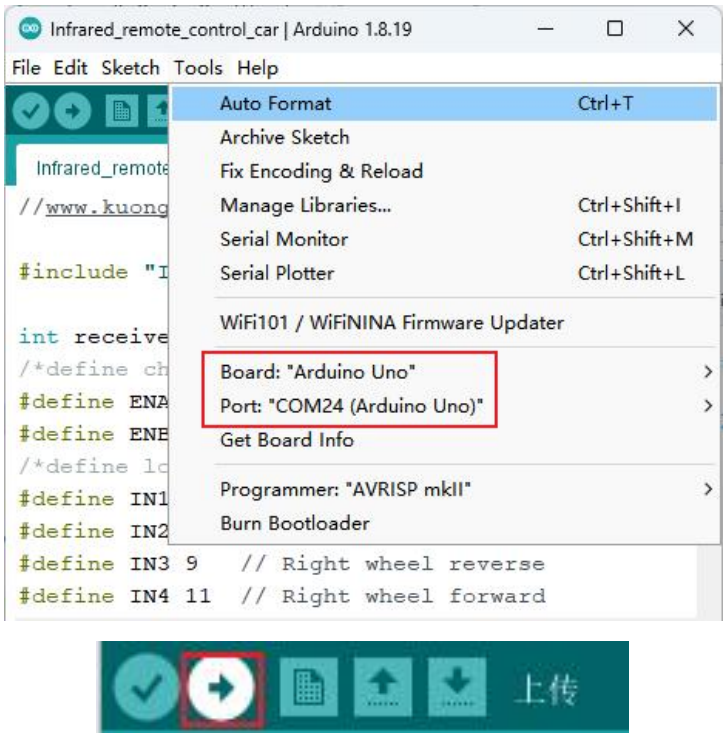
Connect the cart to the computer via the data cable and unplug the Bluetooth module, if you have already unplugged it, please put the Bluetooth module away and plug it back in after uploading the program in case you lose it!



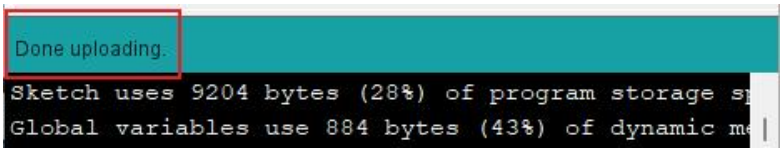
Open the infrared test program at the following path



After confirming that the board and port are Arduino UNO, click the Upload button to upload the program.



When done uploading, disconnecting the car to the computer. Then turn on the power switch and put the car on the ground. Press the button on the remote and you can see the car move accordingly as you command.



Code preview:

```
//www.kuongshun.com

#include "IRremote.h"

int receiver = 12;
/*define channel enable output pins*/
#define ENA 5   // Left  wheel speed
#define ENB 6   // Right wheel speed
/*define logic control output pins*/
#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 carSpeed 200  // initial speed of car >=0 to <=255

IRrecv irrecv(receiver);
uint32_t last_decodedRawData = 0;

void translateIR() // takes action based on IR code received
{
  if (irrecv.decodedIRData.flags)// Check if it is a repeat IR code
  {
    irrecv.decodedIRData.decodedRawData = last_decodedRawData;    //set the
current decodedRawData to the last decodedRawData
    Serial.println("REPEAT!");
  } else//output the IR code on the serial monitor
  {
    Serial.print("IR code:0x");
    Serial.println(irrecv.decodedIRData.decodedRawData, HEX);
  }
  switch (irrecv.decodedIRData.decodedRawData)//map the IR code to the remote key
  {
    case 0xB946FF00: Serial.println("UP");forward();break;
    case 0xEA15FF00: Serial.println("DOWN");back();break;
    case 0xBB44FF00: Serial.println("LEFT");left();break;
    case 0xBC43FF00: Serial.println("RIGHT");right();break;
    case 0xBF40FF00: Serial.println("OK");stop();break;
    case 0xAD52FF00: Serial.println("0");    break;
    case 0xE916FF00: Serial.println("1");digitalWrite(13, HIGH);break;
    case 0xE619FF00: Serial.println("2");digitalWrite(13, LOW);break;
    case 0xF20DFF00: Serial.println("3");    break;
    case 0xF30CFF00: Serial.println("4");    break;
```

```
    case 0xE718FF00: Serial.println("5");    break;
    case 0xA15EFF00: Serial.println("6");    break;
    case 0xF708FF00: Serial.println("7");    break;
    case 0xE31CFF00: Serial.println("8");    break;
    case 0xA55AFF00: Serial.println("9");    break;
    case 0xBD42FF00: Serial.println("*");    break;
    case 0xB54AFF00: Serial.println("#");    break;
    default:
    Serial.println(" other button   ");
  }// End Case
  last_decodedRawData = irrecv.decodedIRData.decodedRawData;//store the last
decodedRawData
  delay(100); // Do not get immediate repeat
} //END translateIR

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

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

void setup() {
  Serial.begin(9600);
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
  pinMode(ENA,OUTPUT);
  pinMode(ENB,OUTPUT);
  pinMode(13, OUTPUT);
  stop();
  irrecv.enableIRIn();// Start the receiver
}

void loop() {
  if (irrecv.decode()) // have we received an IR signal?
  {
    translateIR();
    irrecv.resume(); // receive the next value
  }
}
```