

## Remote control

### The purpose of the experiment:

After uploading the **Remote control** program, place the car indoors and pull the curtains to block the outdoor lights. Align the infrared emitter of the infrared remote control with the infrared receiver at the rear of the BatCar, then press the numeric keypad of the infrared remote control to control the BatCar to complete the corresponding action.

### List of components required for the experiment:

BatCar\*1

USB data cable\*1

Infrared remote control\*1



### Experimental code analysis:

```
//=====
//=====
// Infrared Remote control Experiment
//=====
//=====
#include "../IRremote.h"
//=====
//Infrared Remote Control
//=====
int RECV_PIN = 2; // Set Infrared Remote port
```

```

IRrecv irrecv(RECV_PIN);
decode_results results; // Store infrared remote decode data
unsigned long last = millis();
int on = 0; // Infrared Received flag
long run_car = 0x00FF18E7; // key 2
long back_car = 0x00FF4AB5; // key 8
long left_car = 0x00FF10EF; // key 4
long right_car = 0x00FF5AA5; // key 6
long stop_car = 0x00FF38C7; // key 5
long left_turn = 0x00ff30CF; // key 1
long right_turn = 0x00FF7A85; // key 3
//=====
//Set motor port
//=====
int Left_motor_back = 9;
int Left_motor_go = 5;
int Right_motor_go = 6;
int Right_motor_back = 10;
int Right_motor_en = 8;
int Left_motor_en = 7;
void setup()
{
    //Initialize motor drive for output mode
    pinMode(Left_motor_go, OUTPUT);
    pinMode(Left_motor_back, OUTPUT);
    pinMode(Right_motor_go, OUTPUT);
    pinMode(Right_motor_back, OUTPUT);
    pinMode(13, OUTPUT); // Show Infrared Remote Control Received Signal
    irrecv.enableIRIn(); // Start the receiver
}
void run()
{
    digitalWrite(Right_motor_go, HIGH); // right motor go ahead
    digitalWrite(Right_motor_back, LOW);
    digitalWrite(Left_motor_go, HIGH); // set left motor go ahead
    digitalWrite(Left_motor_back, LOW);

```

```
}  
void brake() //stop  
{  
    digitalWrite(Right_motor_go,LOW);  
    digitalWrite(Right_motor_back,LOW);  
    digitalWrite(Left_motor_go,LOW);  
    digitalWrite(Left_motor_back,LOW);  
}  
void left()//turn left  
{  
    digitalWrite(Right_motor_go,HIGH); // right motor go ahead  
    digitalWrite(Right_motor_back,LOW);  
    digitalWrite(Left_motor_go,LOW);    // left motor stop  
    digitalWrite(Left_motor_back,LOW);  
}  
void spin_left()    //Left rotation  
{  
    digitalWrite(Right_motor_go,HIGH); // right motor go ahead  
    digitalWrite(Right_motor_back,LOW);  
    digitalWrite(Left_motor_go,LOW);    // left motor back off  
    digitalWrite(Left_motor_back,HIGH);  
}  
void right() //turn right  
{  
    digitalWrite(Right_motor_go,LOW);    // right motor stop  
    digitalWrite(Right_motor_back,LOW);  
    digitalWrite(Left_motor_go,HIGH);// left motor go ahead  
    digitalWrite(Left_motor_back,LOW);  
}  
void spin_right()    //Right rotation  
{  
    digitalWrite(Right_motor_go,LOW);    // right motor back off  
    digitalWrite(Right_motor_back,HIGH);  
    digitalWrite(Left_motor_go,HIGH);// left motor go ahead  
    digitalWrite(Left_motor_back,LOW);
```

```

}
void back() //back off
{
    digitalWrite(Right_motor_go,LOW); //right motor back off
    digitalWrite(Right_motor_back,HIGH);
    digitalWrite(Left_motor_go,LOW); //left motor back off
    digitalWrite(Left_motor_back,HIGH);
}
//=====================================================
void dump(decode_results *results)//Decode Infrared Remote Control
Received Signal
{
    int count = results->rawlen;
    if (results->decode_type == UNKNOWN)
    {
        brake();
    }
}
void loop()
{
    if (irrecv.decode(&results)) //receive infrared signal
    {
        if (millis() - last > 250) //make sure receive signal
        {
            on = !on;//Sign position reversal
            digitalWrite(13, on ? HIGH : LOW);//LED blink
            dump(&results);//decode
        }
        if (results.value == run_car )//key "2"
            run();//go ahead
        if (results.value == back_car )//key "8"
            back();//back off
        if (results.value == left_car )//key "4"
            left();//turn left
        if (results.value == right_car )//key "6"
            right();//turn right
    }
}

```

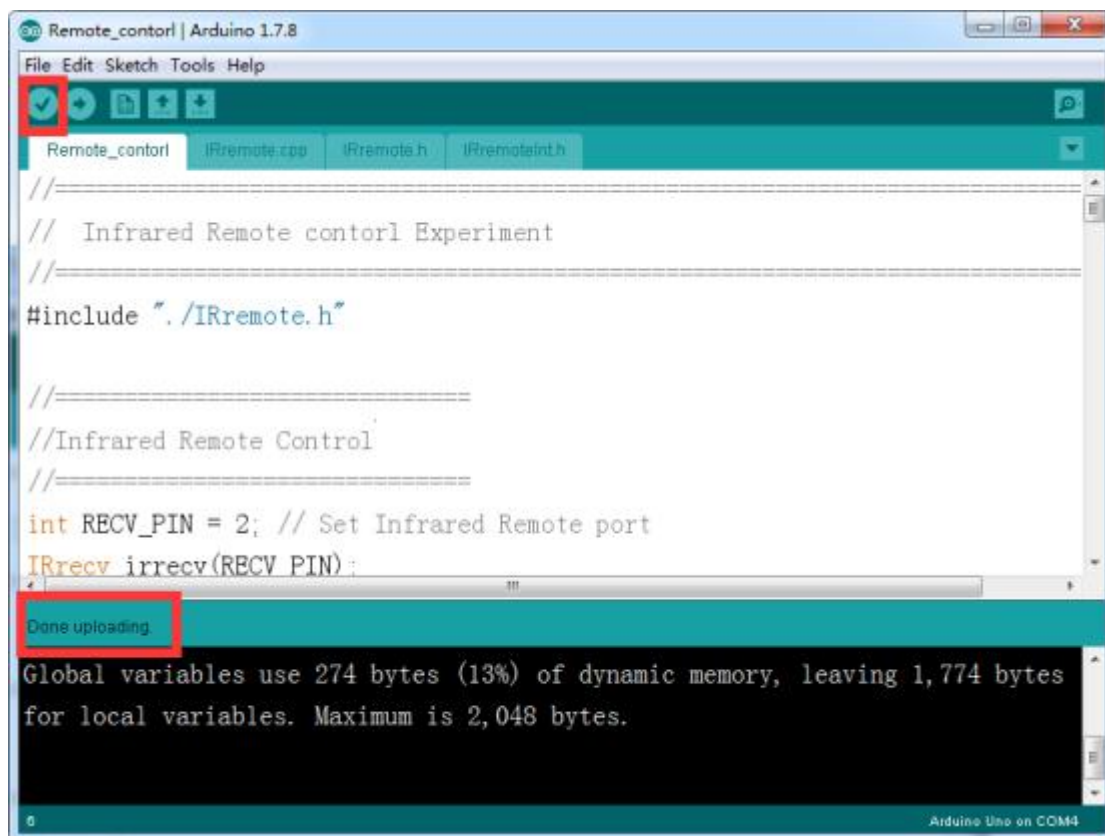
```

if (results.value == stop_car )//key "5"
  brake();//stop
if (results.value == left_turn )//key "1"
  spin_left();//Left rotation
if (results.value == right_turn )//key "2"
  spin_right();//Right rotation
last = millis();
irrecv.resume(); // Receive the next value
}
}

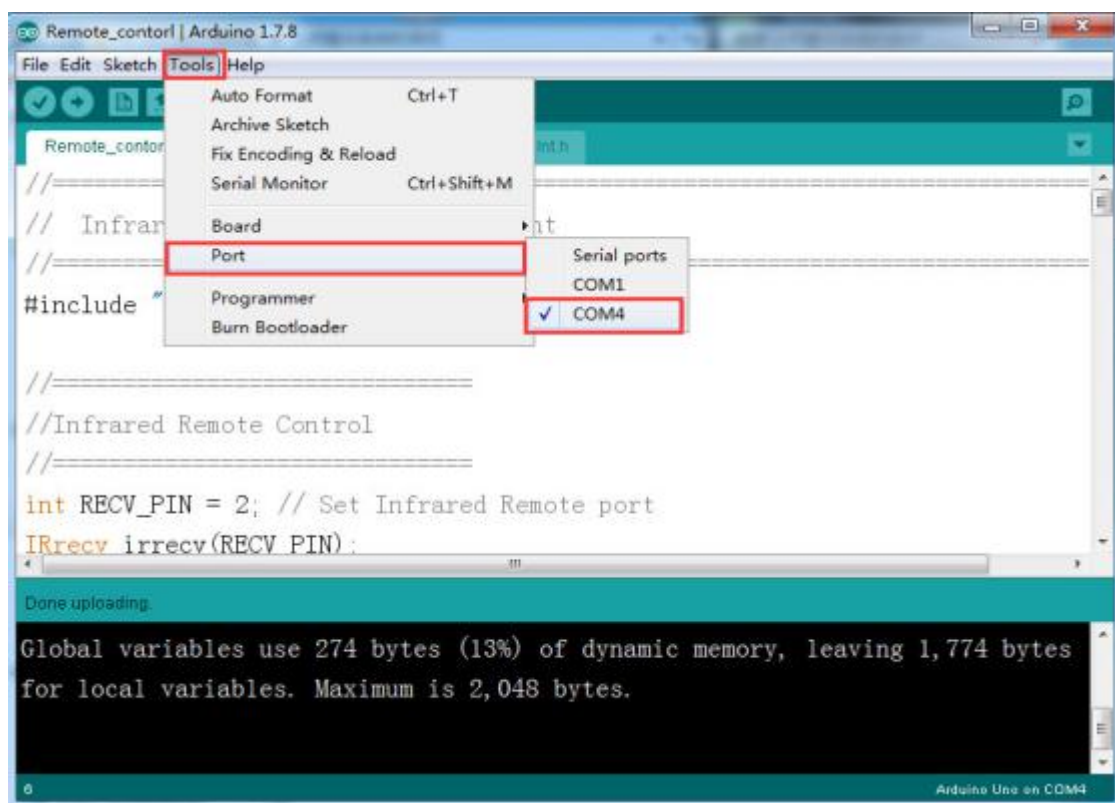
```

### Experimental steps:

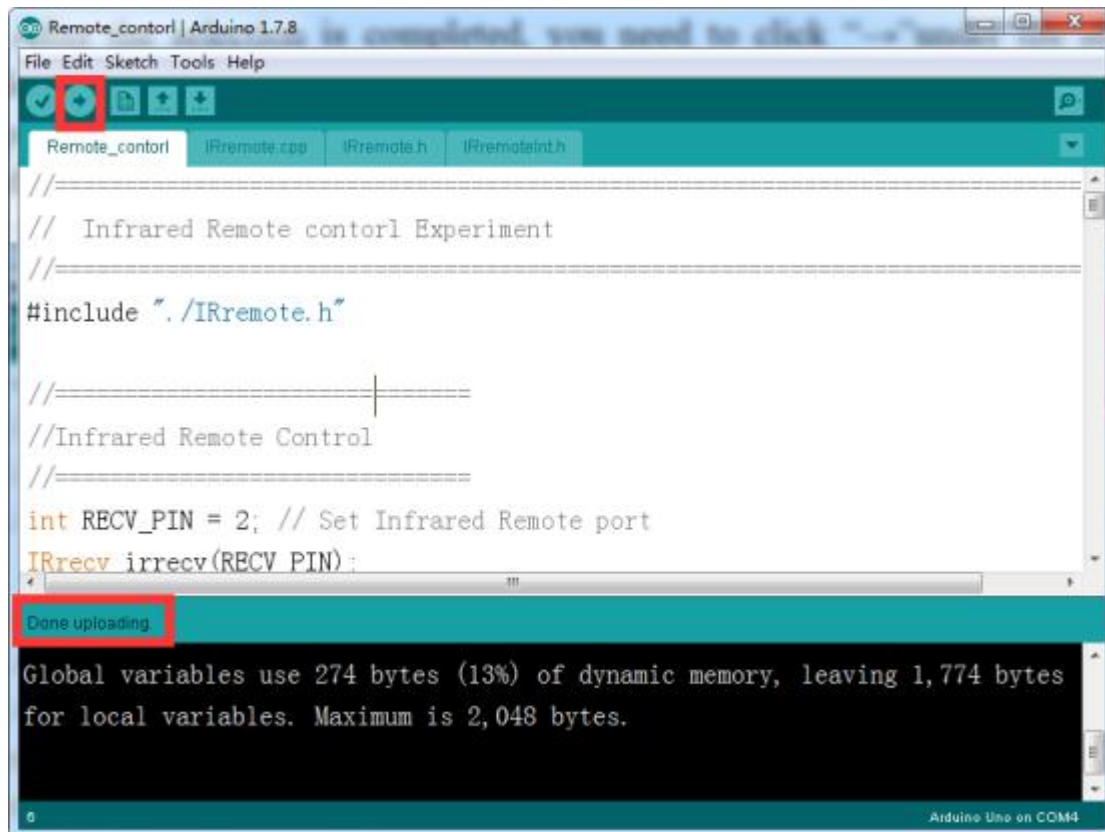
1. We need to open the code of this experiment: **Remote\_contorl.ino**, click“√” under the menu bar to compile the code, and wait for the word **"Done compiling "** in the lower right corner, as shown in the figure below.



2. In the menu bar of Arduino IDE, we need to select **【Tools】 --- 【Port】** --- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.



3. After the selection is completed, you need to click “→” under the menu bar to upload the code to the Arduino UNO board. When the word “**Done uploading**” appears in the lower left corner, the code has been successfully uploaded to the Arduino UNO board, as shown in the figure below.



4. After the program of **Remote control** is uploaded, put the BatCar indoors and pull the curtains to block the outdoor lights. Align the infrared remote control with the infrared receiver at the rear of the BatCar. Press the digital button on the infrared remote control to control the BatCar to complete the corresponding action. The following is the user code value corresponding to the infrared remote control.

| Remote control button | Corresponding user code value | The program controls the action of the BatCar |
|-----------------------|-------------------------------|---|
| 0                     | 0x00FF6897                    | No control action                             |
| —                     | 0x00FF9867                    | No control action                             |
| C                     | 0x00FFB04F                    | No control action                             |
| 1                     | 0x00ff30CF                    | Left rotation                                 |
| 2                     | 0x00FF18E7                    | Forward                                       |
| 3                     | 0x00FF7A85                    | Right rotation                                |
| 4                     | 0x00FF10EF                    | Turn left                                     |
| 5                     | 0x00FF38C7                    | Brake   |
| 6                     | 0x00FF5AA5                    | Turn right                                    |
| 7                     | 0x00FF42BD                    | No control action                             |
| 8                     | 0x00FF4AB5                    | Backward                                      |
| 9                     | 0x00FF52AD                    | No control action                             |

User code : 00FF

