Lección 4 control del coche por infrarrojos

Puntos de esta sección

El control remoto por infrarrojos es un método ampliamente utilizado para el control remoto.

El coche ha sido equipado con receptor de infrarrojos y por lo tanto permite el controlarlo mediante un mando a distancia por infrarrojos

Partes a aprender:

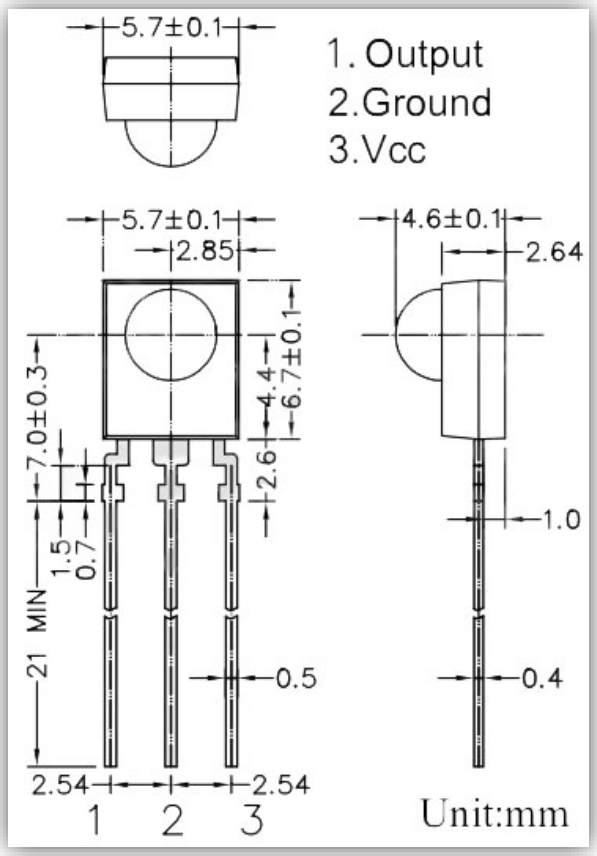
- *Comprender el funcionamiento de los módulos de emission y recepción infrarrojos*
- *Comprender las partes de control remoto*

Necesitaremos:

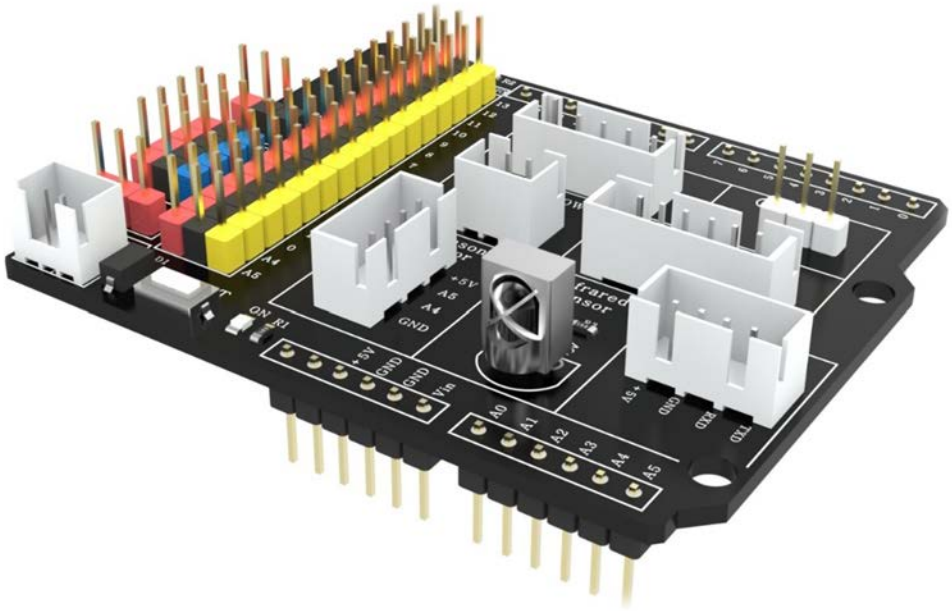
- *Coche con batería*
- *Cable USB*
- *Modulos de emission y recepción de infrarrojos*

Ⅰ. Modulos de emisión y recepción de infrarrojos

Los datos del sensor de recepción de infrarrojos (IR) son los siguientes



La conexión del modulo receptor es la siguiente:



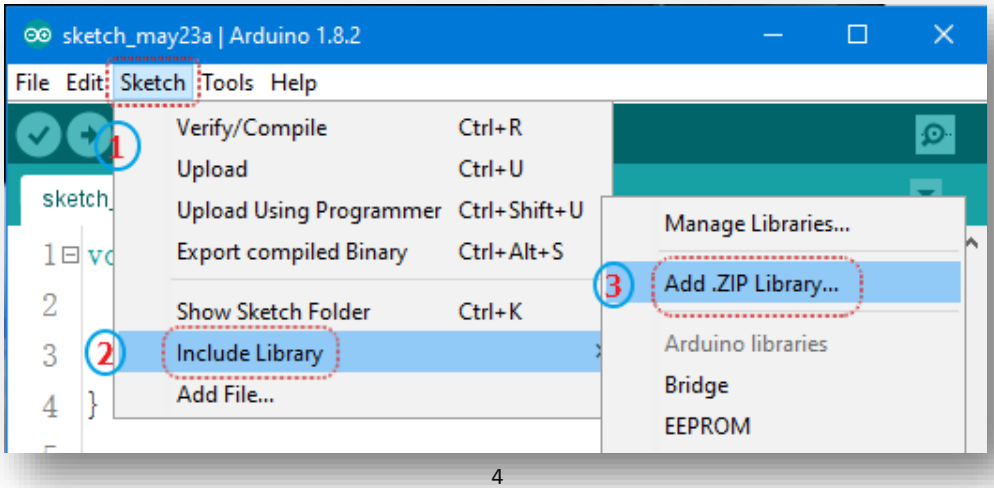
Este es el emisor de infrarrojos (mando):

Ⅱ. Testeo del programa

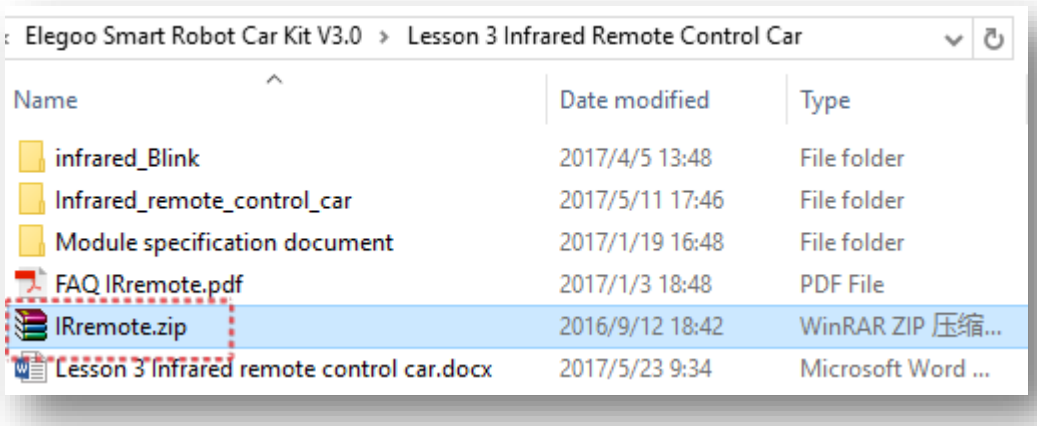
Para este programa necesitaremos, antes de nada, añadir un archive a la biblioteca

Conecte UNO al ordenador y abra el Arduino IDE.

pulse Sketch --> Include Library --> Add .ZIP Library… --> después selecione la librería como debajo.

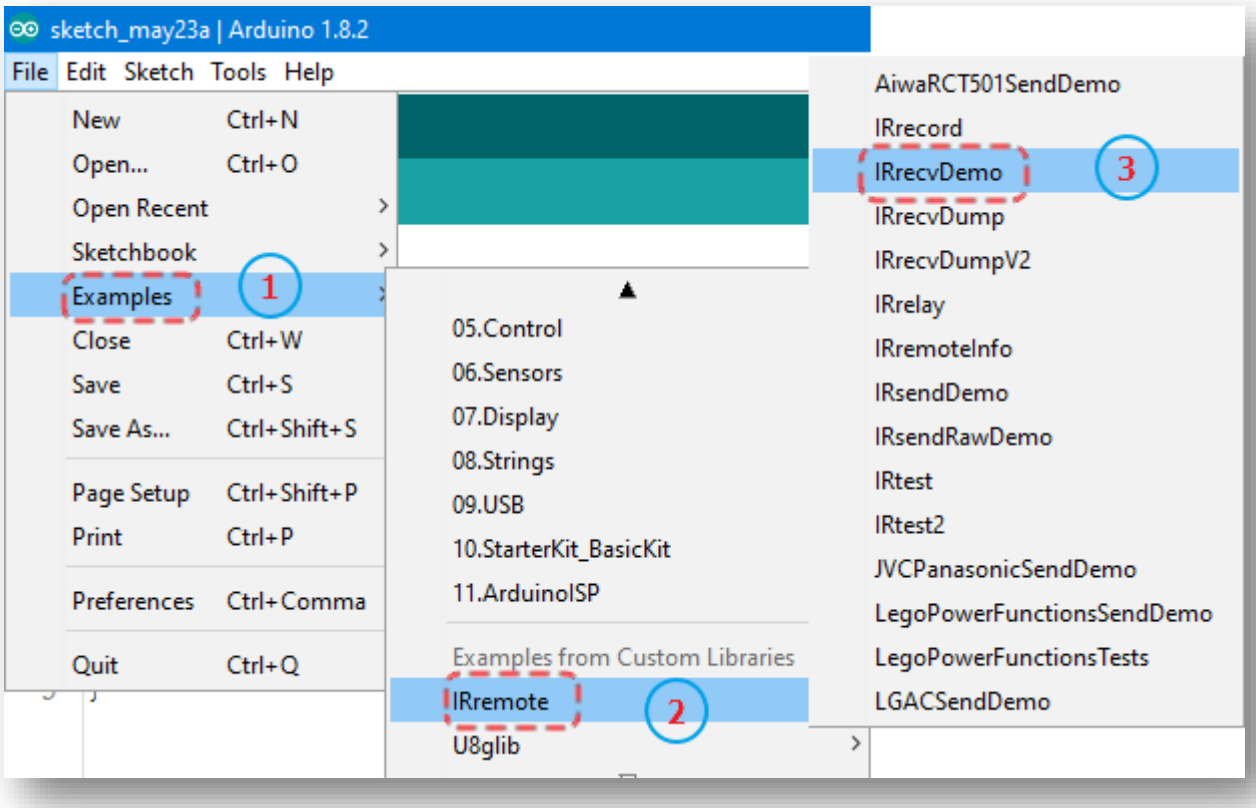


El nombre del archivo zip que contiene la libreria es “IRremote.zip”.

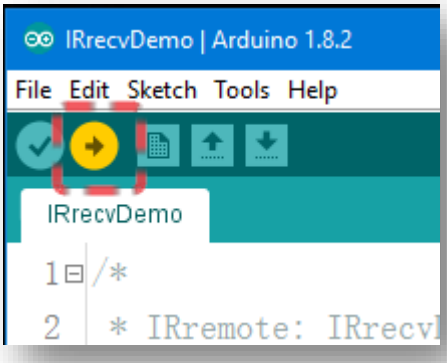


Debe ser compilado con este archivo de biblioteca, que es un archivo de biblioteca especialmente modificado.

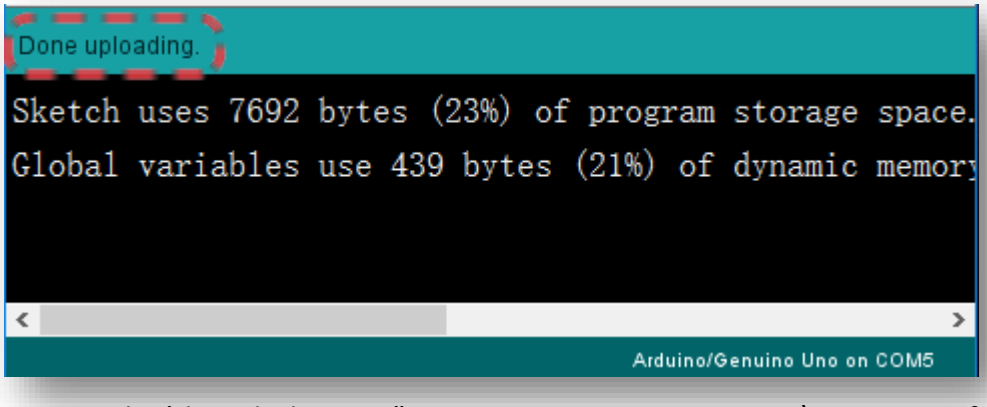
Seleccione el ejemplo IRremote



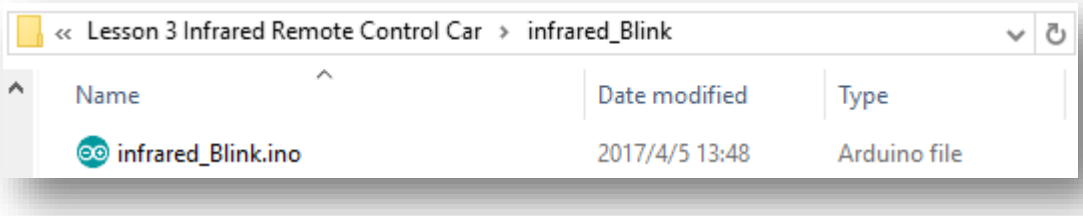
Pulse el botón de compilar



Si la compilación no se completa correctamente significará que la biblioteca IRremote no se ha instalado correctamente y deberá volver a instalarla.



Abra el archive con el código de la ruta ”\Elegoo Smart Robot Car Kit V3.0\Lesson 4 Infrared Remote Control Car\infrared_Blink\infrared_Blink.ino” y cargue el programa en la placa de control.



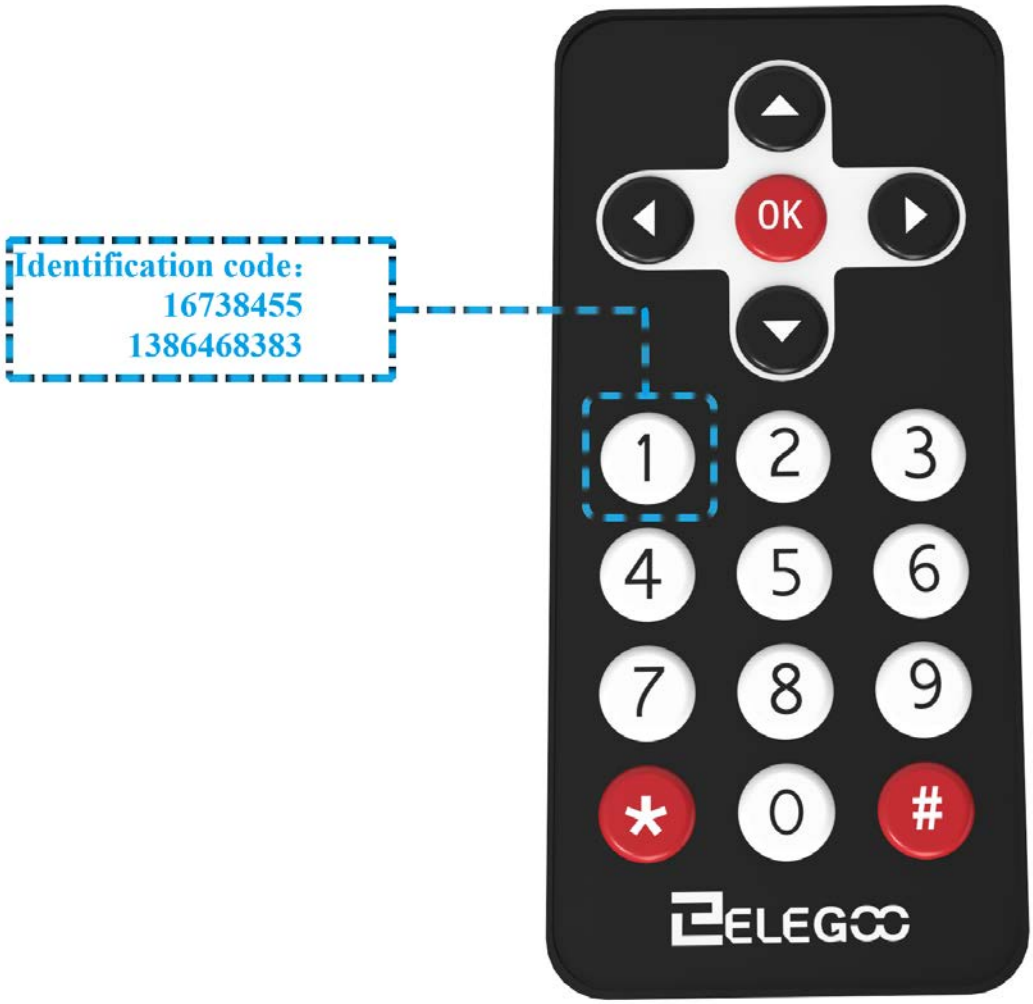
Vista previa del código :

```
//www.elegoo.com

#include <IRremote.h>

#define RECV_PIN  12       //Infrared signal receiving pin
#define LED      13       //define LED pin
#define L        16738455
#define UNKNOWN_L 1386468383

bool state = LOW;          //define default input mode
unsigned long val;

IRrecv irrecv(RECV_PIN);   //initialization
decode_results results;   //Define structure type

void stateChange() {
  state = !state;
  digitalWrite(LED, state);
}
```

```
void setup() {
  pinMode(LED, OUTPUT); //initialize LED as an output
  Serial.begin(9600);  // debug output at 9600 baud
  irrecv.enableIRIn();  // Start receiving
}

void loop() {
  if (irrecv.decode(&results)) {
    val = results.value;
    Serial.println(val);
    irrecv.resume();     // Receive the next value
    delay(150);
    if(val == L || val == UNKNOWN_L) {
      stateChange();
    }
  }
}
```

Tras desconectar el coche del pc puede encenderlo y ponerlo en el suelo.

Presione el botón “1” mirando hacia al coche. Si observa el coche se dará cuenta que el LED “L” se encontrará apagado.



Ⅲ. Introducción al principio

1. Principio de funcionamiento

El sistema de control remoto infrarrojo universal consta de dos partes: envío y recepción, la parte de envío consta de un mando a distancia IR, mientras que la parte receptora consta de un tubo receptor infrarrojo. Las señales enviadas por control remoto IR son una serie de códigos de impulsos binarios. Con el fin de estar libre de la interferencia de otras señales de infrarrojos durante el transporte inalámbrico, es usual modularlo en la frecuencia portadora dado, y luego lanzarlo a través de infrarrojos emitidos desde el fototransistor. El tubo de recepción infrarrojo filtra otras ondas de ruido, dado que sólo recibe señales de frecuencia dada y las restaura a código de pulso binario que es de modulación. El tubo receptor incorporado transforma las señales luminosas que se envían desde el diodo emisor de luz infrarrojo a señales eléctricas débiles, las señales se agrandan con el amplificador dentro del IC, y con el control automático de la ganancia, el filtrado de la venda-paso, la demodulación, codificación enviada por control remoto, reconocer el circuito mediante la codificación que se introduce en el aparato eléctrico a través de la salida de señal pin del módulo de recepción de infrarrojos.

2. Protocolo de control por infrarrojos

El esquema de codificación del control remoto IR combinado es: Protocolo NEC.

A continuación, vamos a aprender qué es el protocolo NEC.

caracteristicas:

(1) 8 bits de dirección, 8 bits de pedido

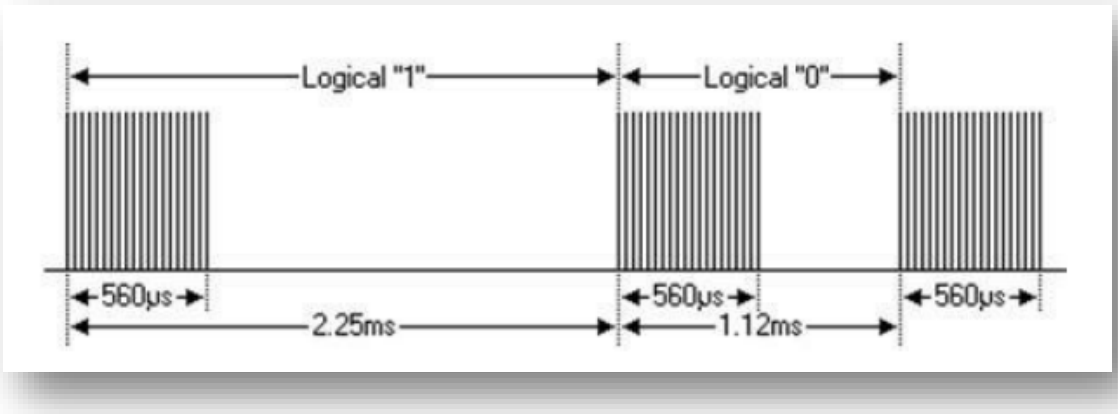
(2) El bit de dirección y el bit de orden se transmiten dos veces para garantizar la fiabilidad

(3) Modulación de la posición del pulso

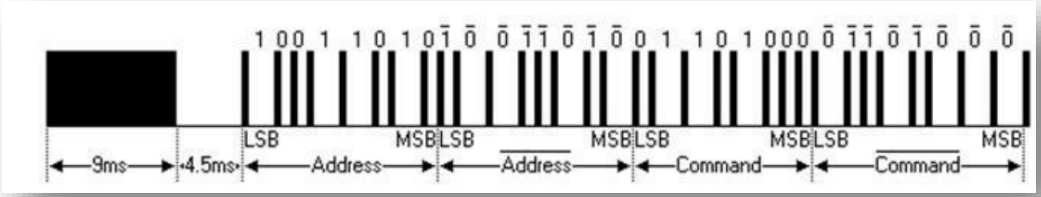
(4) La frecuencia del portador es 38kHz

(5) El tiempo de cada bit es 1.125ms o 2.25ms

Definitions of logical 0 and 1 are as below:

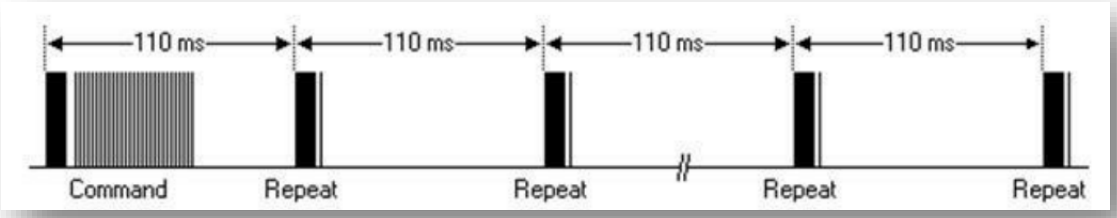


El protocol es el siguiente:



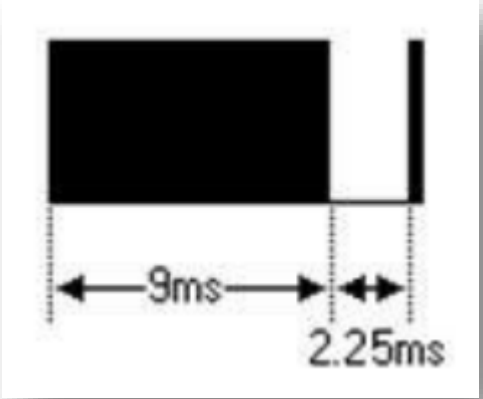
Nota: Este es el protocolo de envío de LSB (bit menos significativo ó least-significant bit) en primer lugar. La dirección de transporte del impulso anterior es 0x59, el orden es 0x16. Un mensaje comienza desde un nivel alto de 9ms, el siguiente es un nivel bajo de 4.5ms, (código de guía de formularios de dos niveles) y a través de código de dirección y código de pedido. La dirección y el pedido se transmiten dos veces. En la segunda vez, todos los bits se invierten. Se puede utilizar para confirmar los mensajes de recepción que se utilizarán. El tiempo de envío total es fijo, si no está interesado en él, puede ignorar la fiabilidad de invertir, y puede ampliar la dirección y el orden en 16 bits! Porque el hecho se repite si la longitude de cada bit es opuesta.

Presione el pulso transmitido aflojado después de un tiempo.



Una vez que se ha enviado un comando, incluso si se pulsa el botón de control remoto. Cuando se pulsa el botón, el impulso de los primeros 110 ms es diferente del anterior, el código duplicado se transmite después de cada 110 ms. El código duplicado se compone de un pulso de alto nivel de 9ms y un nivel bajo de 2.25 y un alto nivel de 560µs.

Repetir el pulso:



Nota: Después de que la forma de onda de impulso entre en la integración del sensor, debido al hecho de que la integración del sensor debe ser decodificada, la señal magnificada y el plástico, usted debe notar que cuando no hay señales infrarrojas, su terminal de salida es nivel alto y e snivel bajo cuando hay señales. Así que el nivel de la señal de salida es opuesto al terminal de transmisión. Todo el mundo puede ver el pulso del receptor a través del osciloscopio y entender el programa vista la forma de onda de la señal.

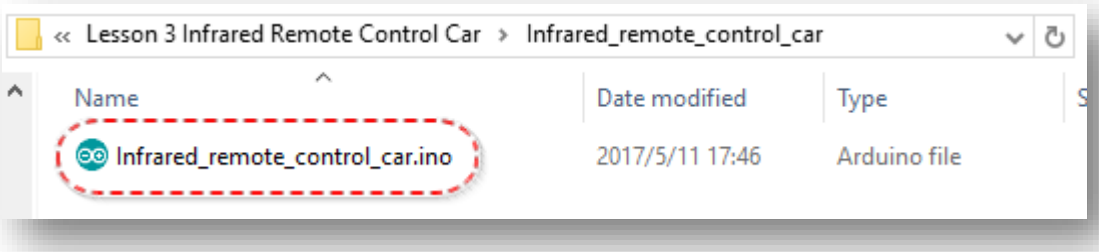
3. La idea de programar el coche de control remoto

De acuerdo con la característica del código NEC y la onda del extremo de recepción, este experimento divide la onda del extremo de recepción en cuatro partes: código principal (pulso de 9ms y 4,5ms), código de dirección (incluyendo código de dirección de 8 bits y 8 bits dirección de búsqueda),
Código de dirección de 16 bits (incluyendo código de dirección de 8 bits y búsqueda de dirección de 8 bits), código de pedido de 16 bits (incluyendo código de pedido de 8 bits y búsqueda de orden de 8 bits), código de repetición (compuesta de pulso de 9ms , 2,25ms, 560us).
Aproveche el temporizador para probar el nivel alto y bajo de la onda recibida, distinguiéndose según el tiempo probado: lógico "01", "1" lógico, impulso de avance, pulso de repetición. Código de dirección y código de dirección se juzgan si es correcto, no se almacenan, debido al hecho de que el código de pedido de cada clave es diferente, la acción se lleva a cabo por código de pedido.
Durante el experimento en el coche, sólo tenemos que controlar el coche para ir hacia adelante y hacia atrás, girar a la izquierda y la derecha, y detener, lo que significa que necesitaríamos 5 claves y el valor de ellos son los siguientes:

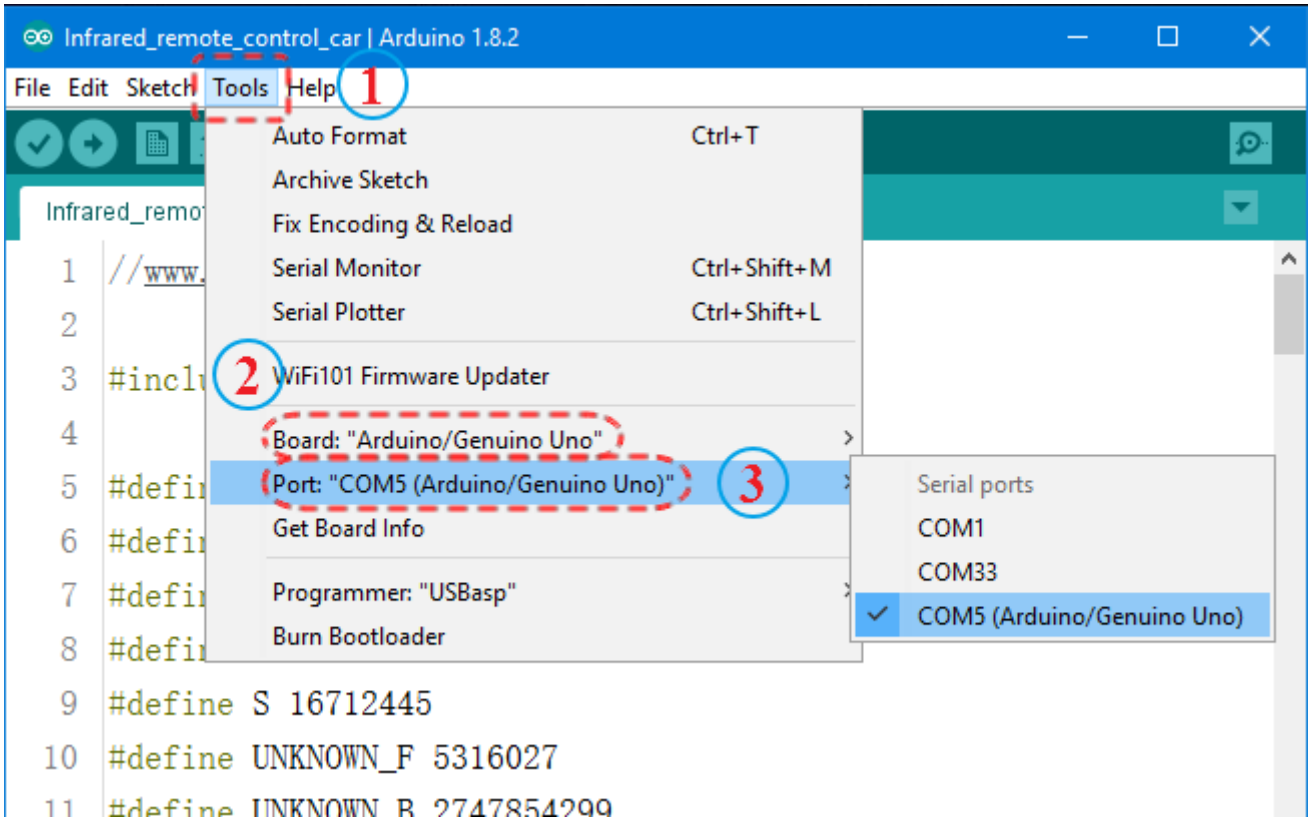
Botón de control remoto	Valor de tecla
Botón rojo del medio	16712445, 3622325019
Triángulo hacia adelante	16736925, 5316027
Triángulo hacia atrás	16754775, 2747854299
Triángulo izquierda	16720605, 1386468383
Triángulo derecha	16761405, 553536955

IV. Hacer un coche de control remoto

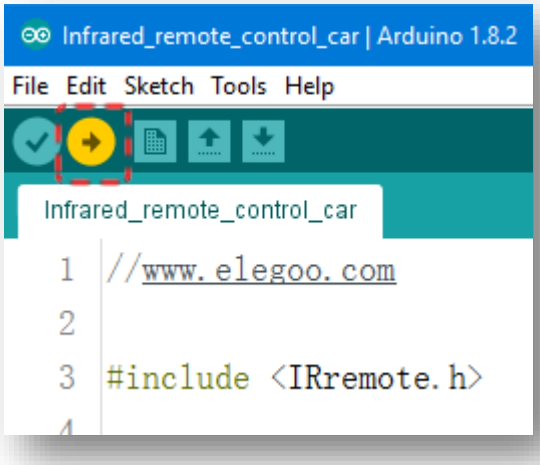
Abrir el archive con el código en la ruta"\Elegoo Smart Robot Car Kit V3.0\Infrared_remote_control_car\ Infrared_remote_control_car.ino" y después subir el código al coche como se muestra a continuación:



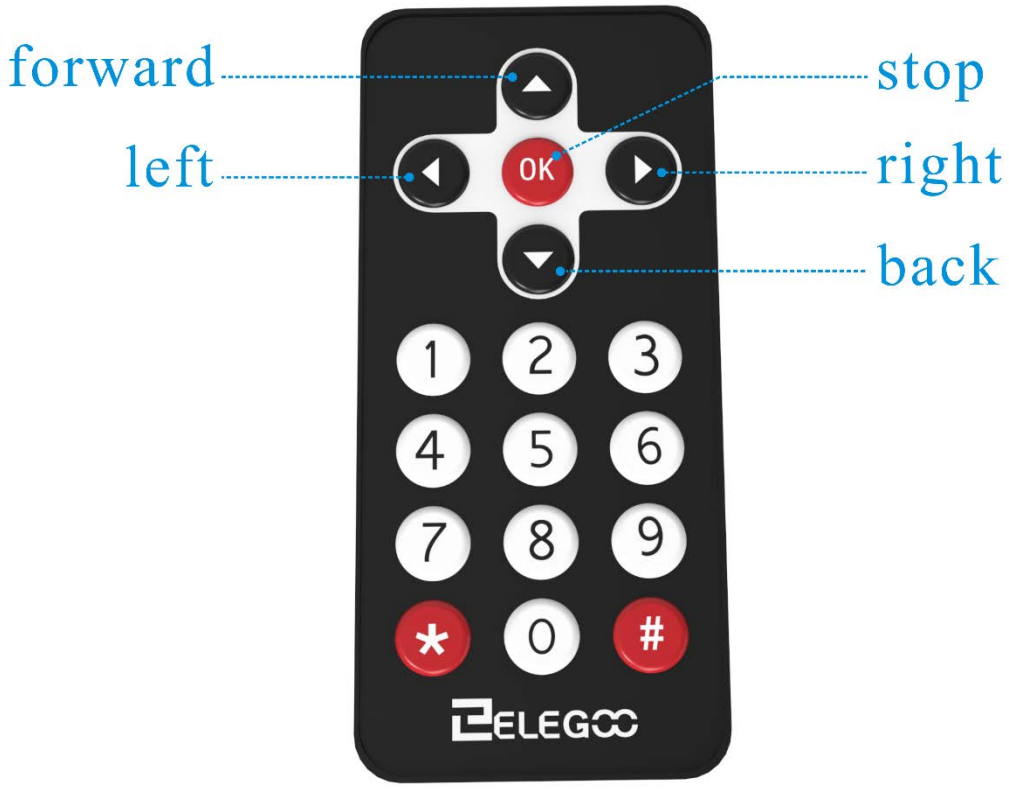
Seleccionar el Arduino Uno Board y Serial port (Puerto serie).



Pulsar el botón upload (cargar)



Cuando termine de cargar, desconectando el automóvil del ordenador. A continuación, encienda el interruptor y coloque el vehículo en el suelo. Presione el botón en el mando a distancia y podrá ver el movimiento del coche en consecuencia a como usted ordena.



A partir de ahora ya podrá jugar con el coche y el control por infrarrojos.

Vista previa del código:

```
//www.elegoo.com

#include <IRremote.h>

#define F 16736925
#define B 16754775
#define L 16720605
#define R 16761405
#define S 16712445
#define UNKNOWN_F 5316027
#define UNKNOWN_B 2747854299
#define UNKNOWN_L 1386468383
#define UNKNOWN_R 553536955
#define UNKNOWN_S 3622325019

#define RECV_PIN  12
```

```
#define ENA 5
#define ENB 6
#define IN1 7
#define IN2 8
#define IN3 9
#define IN4 11
#define carSpeed 150

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

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);
  stop();
  irrecv.enableIRIn();
}

void loop() {
  if (irrecv.decode(&results)){
    preMillis = millis();
    val = results.value;
    Serial.println(val);
    irrecv.resume();
    switch(val){
      case F:
      case UNKNOWN_F: forward(); break;
      case B:
      case UNKNOWN_B: back(); break;
      case L:
      case UNKNOWN_L: left(); break;
      case R:
      case UNKNOWN_R: right();break;
      case S:
      case UNKNOWN_S: stop(); break;
      default: break;
    }
  }
  else{
```

```
    if(millis() - preMillis > 500){
      stop();
      preMillis = millis();
    }
  }
}
```