

## Lección 3 Control de infrarrojos coche



## Los 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 que sea controlado mediante el mando a distancia de infrarrojos..

Aprendiendo sobre las piezas:

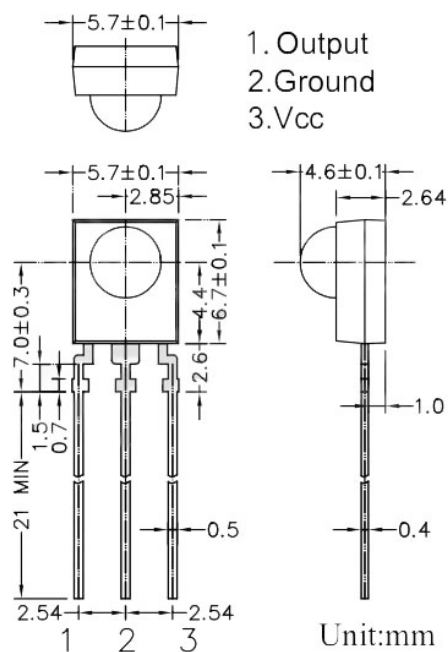
- ◆ Entender el mando a distancia por infrarrojos y el receptor
- ◆ Entender los principios del mando a distancia

Preparación:

- ◆ Un coche (con la batería)
- ◆ Un cable USB
- ◆ Módulo receptor IR y control remoto IR

## I . Módulo receptor IR y control remoto IR

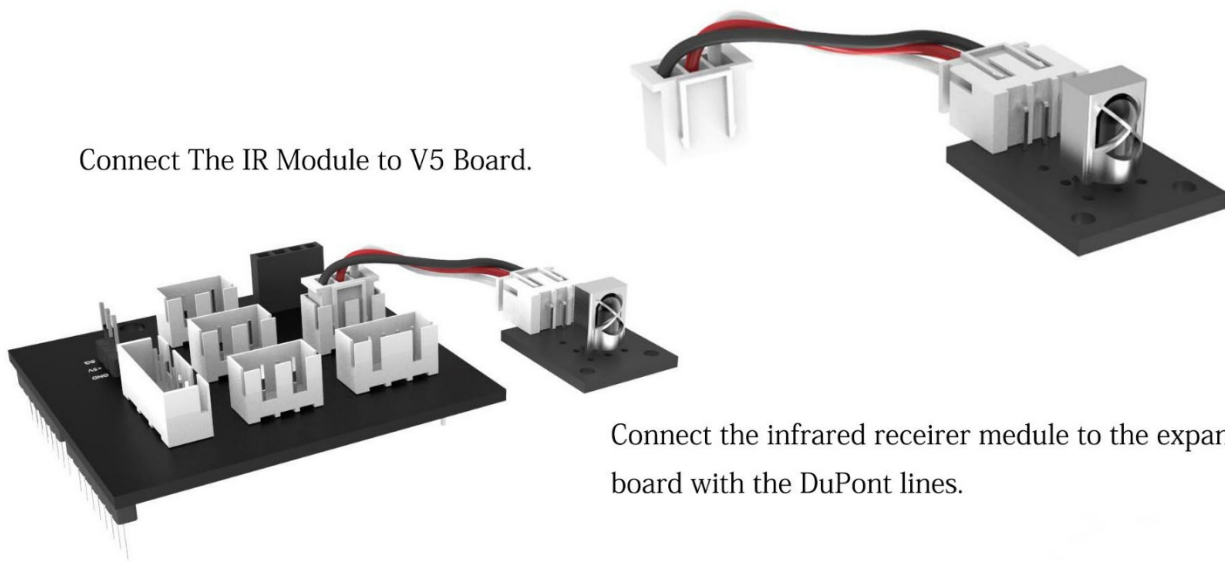
Los datos del sensor del receptor IR son los siguientes:



La conexión del módulo receptor es la siguiente:

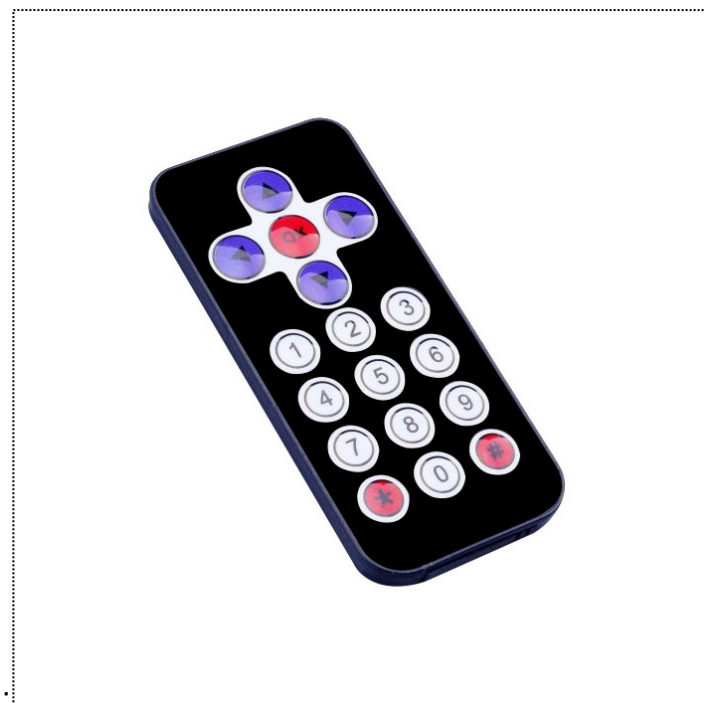


Connect The IR Module to V5 Board.



Connect the infrared receiver module to the expansion board with the DuPont lines.

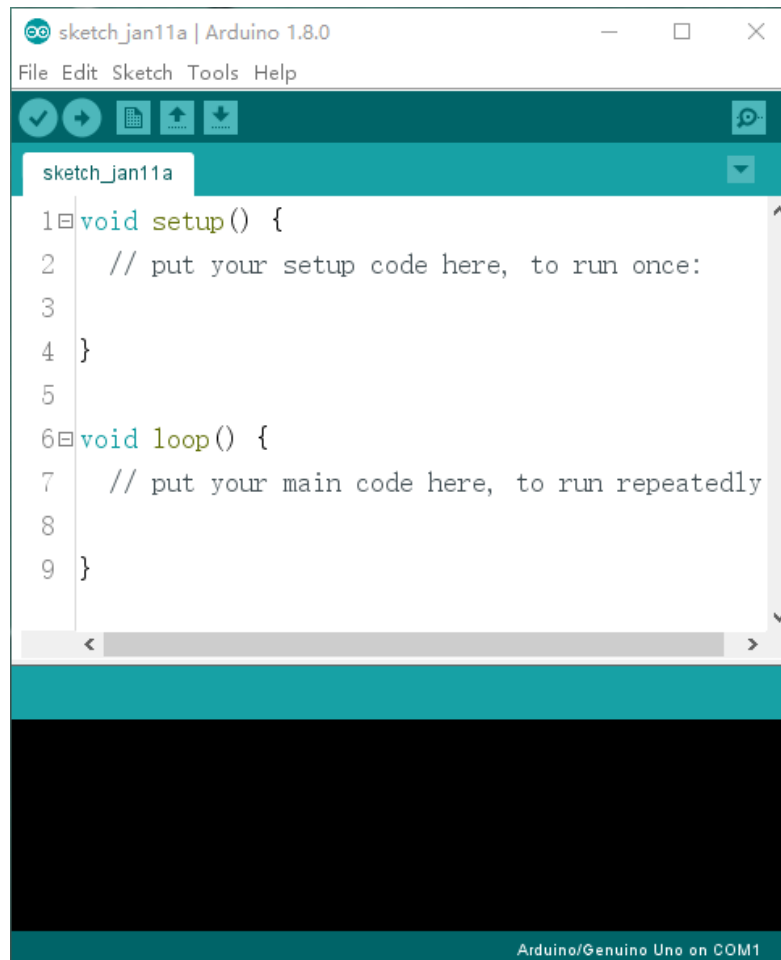
Esto es IR remoto:



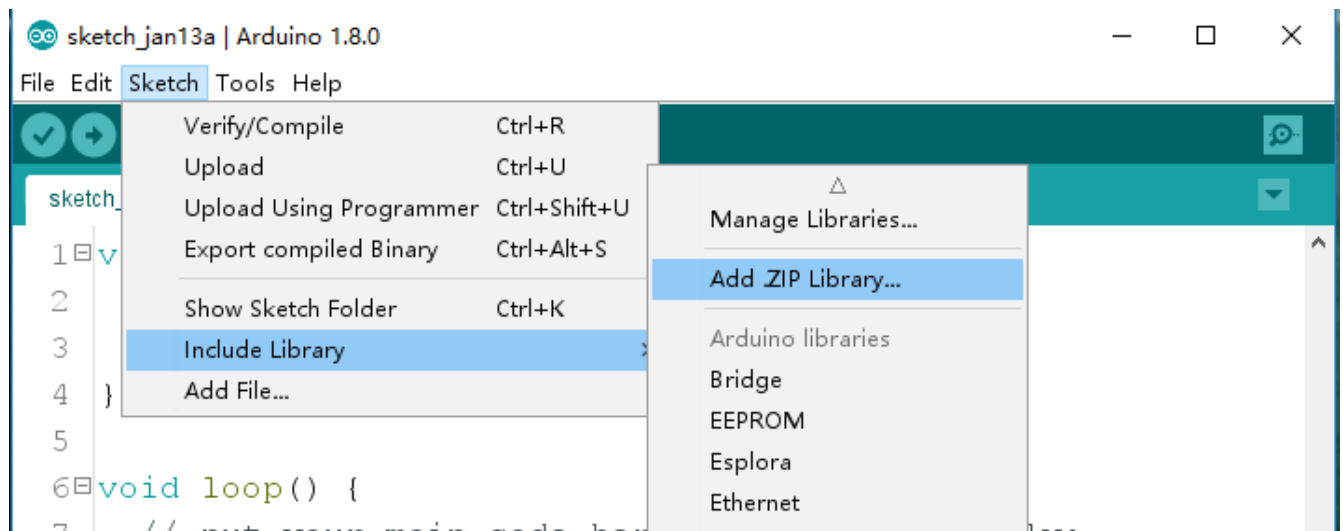
## II. Programa de pruebas

Porque en este programa, necesitamos usar la biblioteca de modo que necesitamos agregar el archivo de la biblioteca al principio.

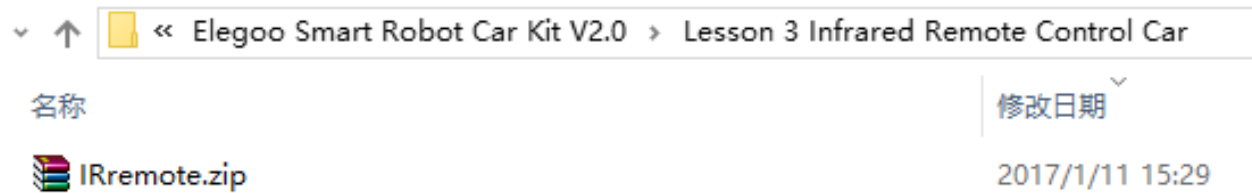
Abra el IDE de Arduino



Haga clic en Sketch-Incluir Biblioteca-Añadir .ZIP Library ... y luego seleccione la biblioteca como.

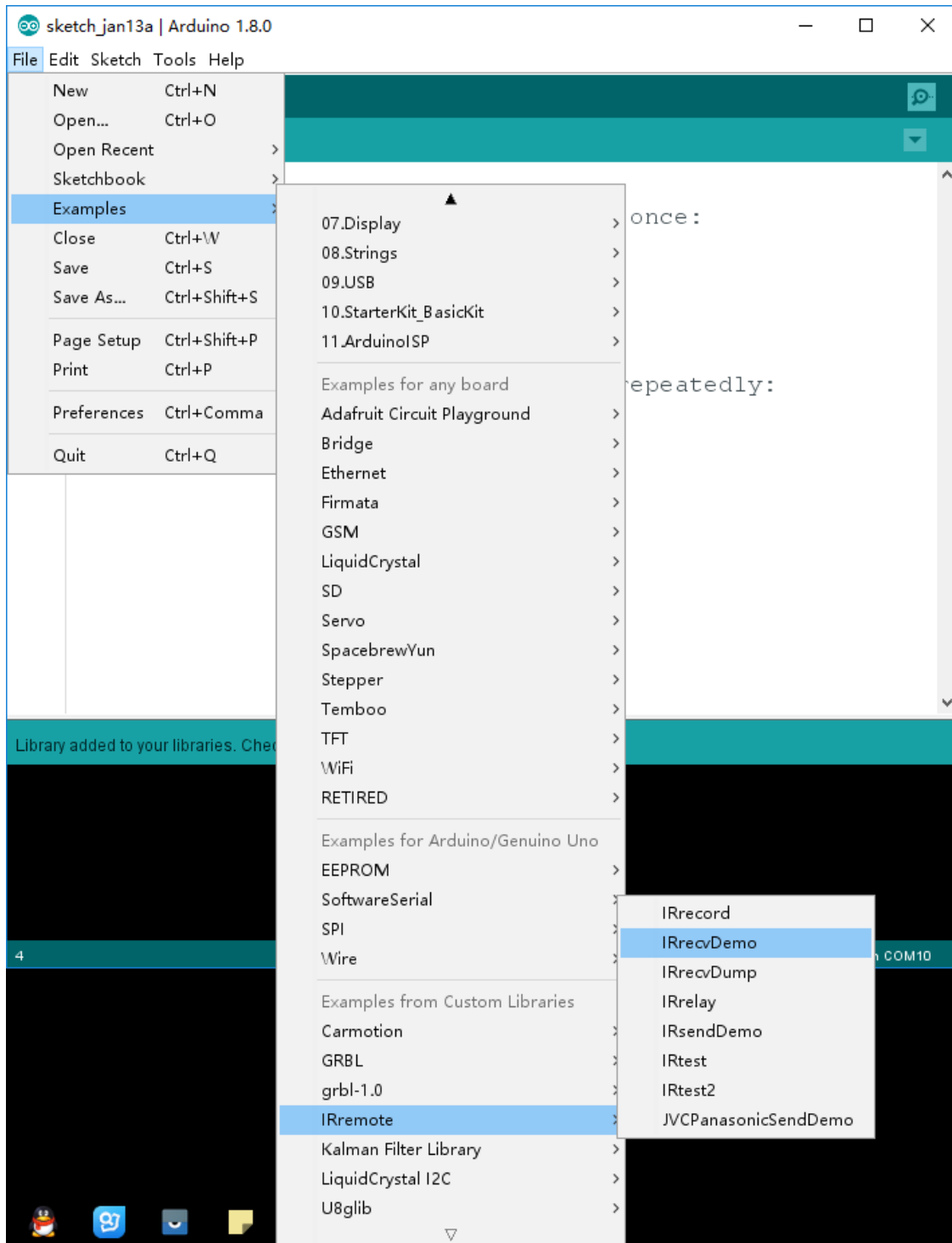


El nombre de archivo de la biblioteca ZIP debe ser IRremote.zip.

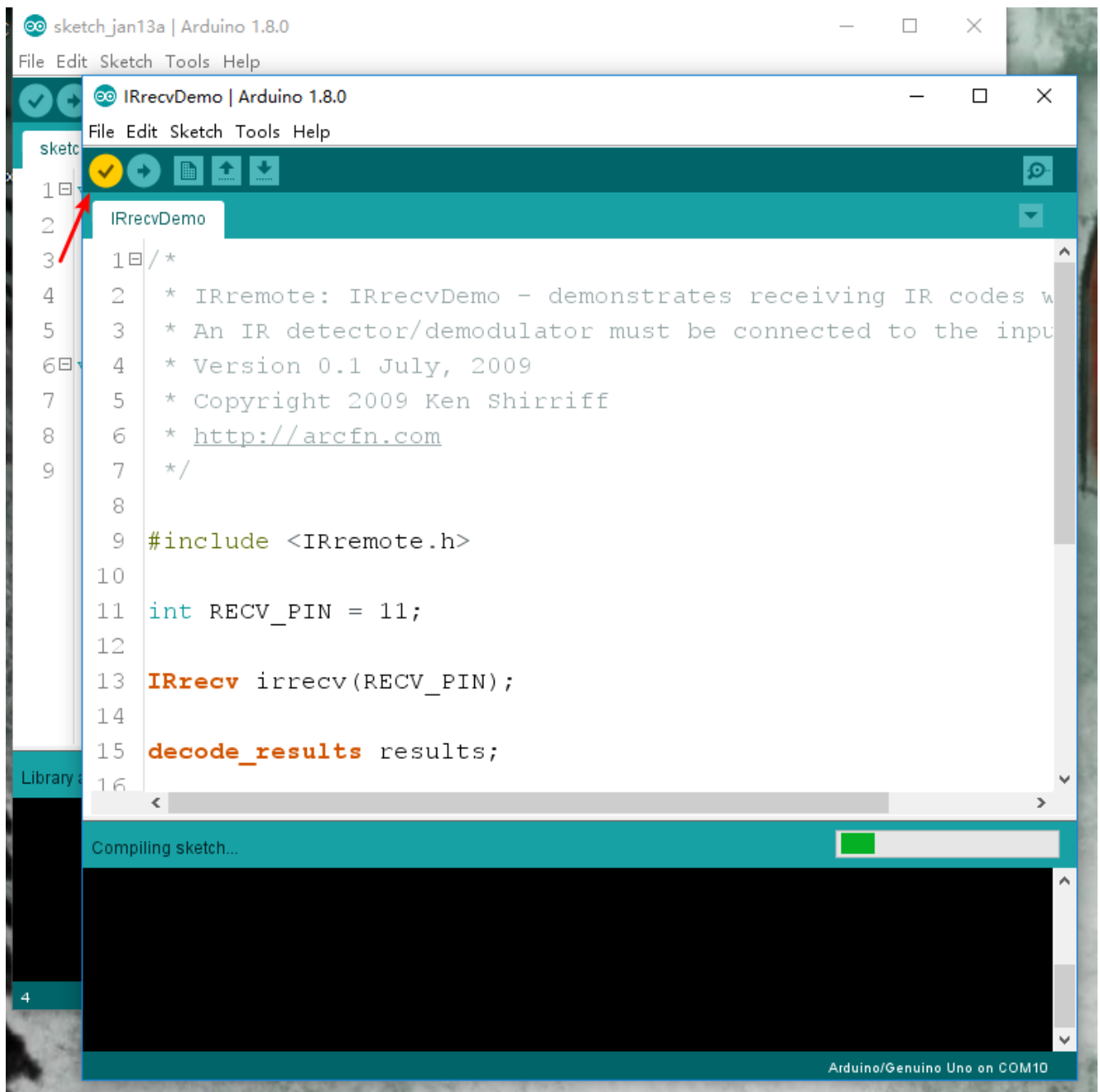


Debe ser compilado con este archivo de biblioteca, que es un archivo de biblioteca especialmente modificado.

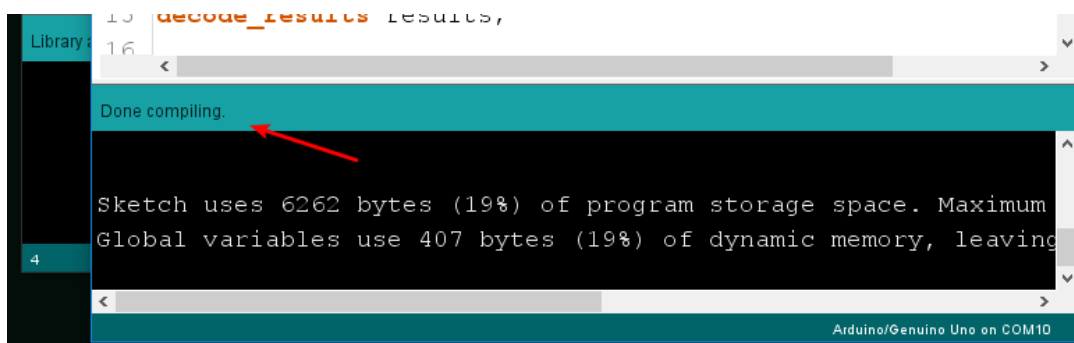
Seleccione un ejemplo IRremote



Haga clic en el botón de compilación.



Realizada compilación. Si no, la biblioteca IRremote no se ha instalado correctamente. Vaya a agregar la biblioteca IRremote nuevamente.





Abre el fichero infrared\_Blink\infrared\_Blink.ino

<div> <div> <div></div> <div> <div></div> <div></div> </div> </div> <div> <div>« Lesson 3 Infrared Remote Control Car » infrared_Blink</div> </div> </div>		
名称	修改日期	类型
<div> <div></div> <div>infrared_Blink.ino</div> </div>	2017/1/5 11:57	Ardui

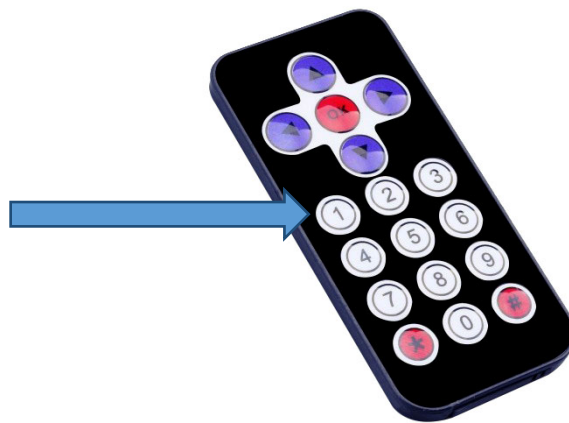
El código es el siguiente:

```
#include <IRremote.h> //Infrared Library
int receiverpin = 12; //Infrared signal receiving pin
int LED=13; //define LED pin
volatile int state = LOW; //define default input mode
unsigned long RED;
#define L 16738455
IRrecv irrecv(receiverpin); //initialization
decode_results results; //Define structure type
void setup() {
  pinMode(LED, OUTPUT); //initialize LED as an output
  Serial.begin(9600); // debug output at 9600 baud
  irrecv.enableIRIn(); // Start receiving
}
void stateChange()
{
  state = !state;
  digitalWrite(LED, state);
}
void loop() {
  if (irrecv.decode(&results))
  {
    RED=results.value;
    Serial.println(RED);
    irrecv.resume(); // Receive the next value
    delay(150);
    if(RED==L)
```

```
{  
  stateChange();  
}  
}  
}
```

Cargue el programa en la tarjeta controladora ROMEO. Después de desconectar el coche al ordenador, puede encender el interruptor de encendido y poner el coche en el suelo.

Presione el botón "1" mirando hacia el coche, observe el coche, y usted encontrará el LED en el coche apagar.



### **III. Introducción principal**

#### **1. Funcionamiento principal**

El sistema de control remoto infrarrojo universal consta de dos partes: el envío y la recepción, la parte de envío consta de un mando a distancia IR, la parte de recepción consta de un tubo receptor de infrarrojos. 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 general para sincronizarlo en la frecuencia correcta, y luego lanzarlo a través de fototransistor emitido por infrarrojos. El tubo de recepción infrarrojo filtra otras ondas de ruido, sólo recibe señales de frecuencia dada y las restaura a código de pulso binario que es demodulación. El tubo de recepción incorporado transforma las señales luminosas que se envían desde el diodo emisor de luz infrarroja a señales eléctricas débiles, las señales se agrandan a través del amplificador dentro del CI y mediante el control automático de ganancia, el filtro de paso de banda, la desmodulación, 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 remoto por infrarrojos**

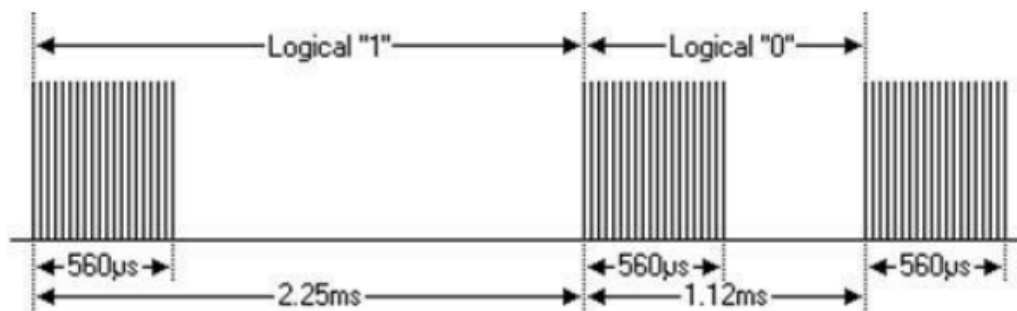
El esquema de codificación del control remoto IR combinado es: Protocolo NEC.

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

Características:

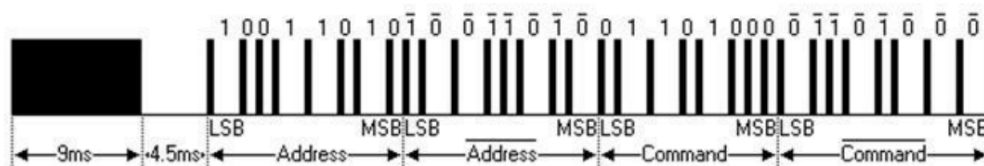
- (1) 8 bits de dirección, 8 bits de pedido
- (2) El bit de dirección y el bit de orden se transmiten dos veces para garantizar la fiabilidad
- (3) Modulación de la posición del pulso
- (4) La frecuencia del portador es 38kHz
- (5) El tiempo de cada bit es 1.125ms o 2.25ms

Las definiciones de 0 y 1 lógicos son las siguientes:



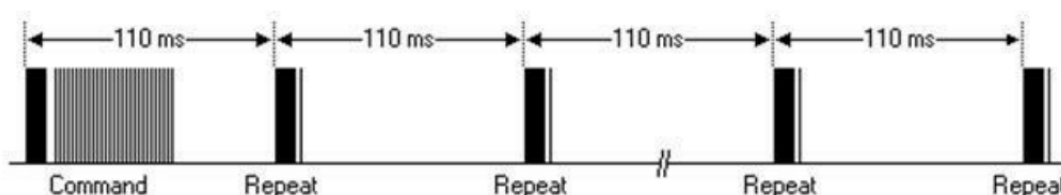
El protocolo es el siguiente:

Presione instantáneamente afloje el pulso de transmisión:



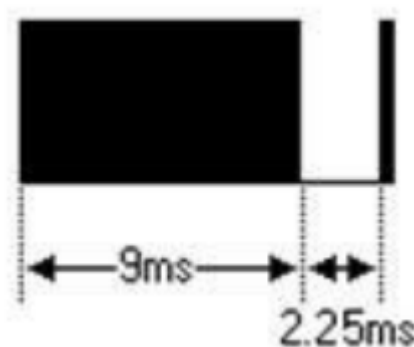
Nota: Este es el protocolo de envío de LSB (bit menos significativo) en primer lugar. La dirección de transporte del impulso anterior es 0x59, el orden es 0x16. Un mensaje comienza desde un nivel alto de 9ms, el siguiente es un nivel bajo de 4.5ms, (código de guía de formularios de dos niveles) ya través 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 lo contrario, se puede utilizar para confirmar los mensajes de recepción que se utilizarán. Tiempo de envío total es fijo, si no está interesado en ella, puede ignorar la fiabilidad de invertir, y puede ampliar la dirección y el orden en 16 bits! Porque el hecho de que la longitud de repetir si cada bit es opuesto.

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 todavía 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, debe anotar el tiempo cuando no hay señales de infrarrojos, su terminal de salida es de alto nivel, 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, entender el programa con la forma de onda visto.

### 3. La idea de programar un coche de control remoto

De acuerdo con la característica del código NEC y la onda 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 el 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 (incluido código de pedido de 8 bits y búsqueda de orden de 8 bits), código de repetición (compuesto de pulso de 9ms , 2,25 ms, 560°).

Explote 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 del 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:

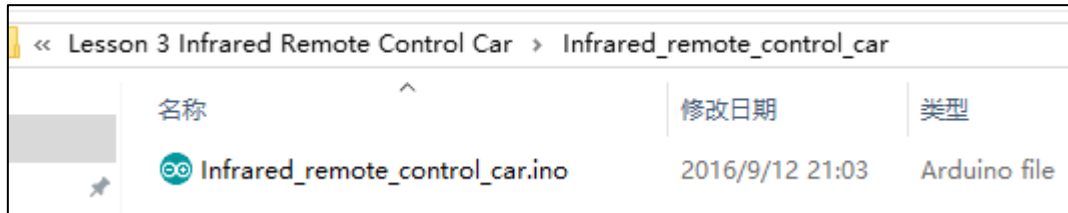
Carácter de control remoto	valor clave
Botón rojo medio	16712445

Por encima del triángulo	16736925
Abajo del triángulo	16754775
Triángulo izquierdo	16720605
Triángulo rectángulo	16761405

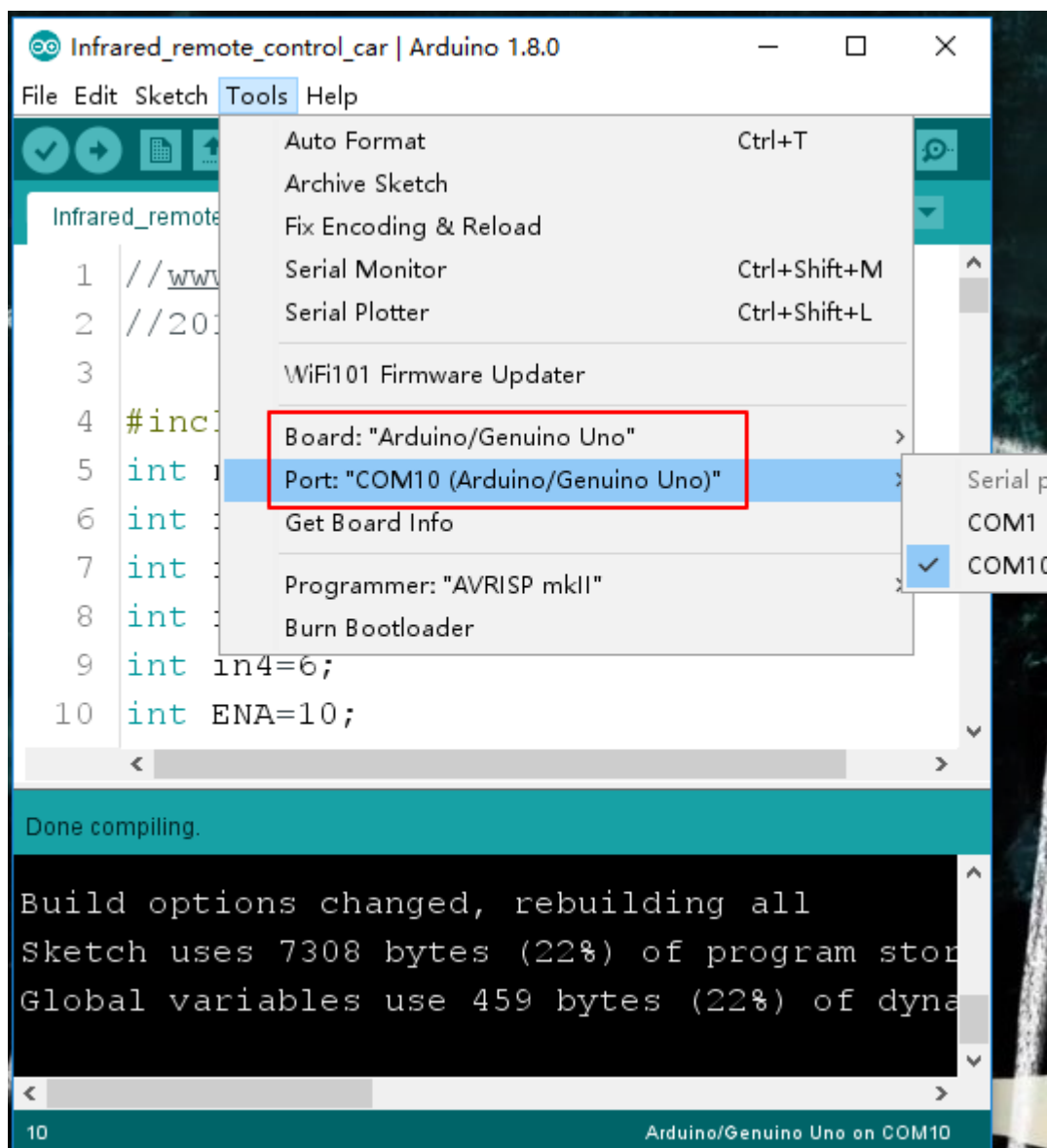


## IV. Hacer un coche de control remoto

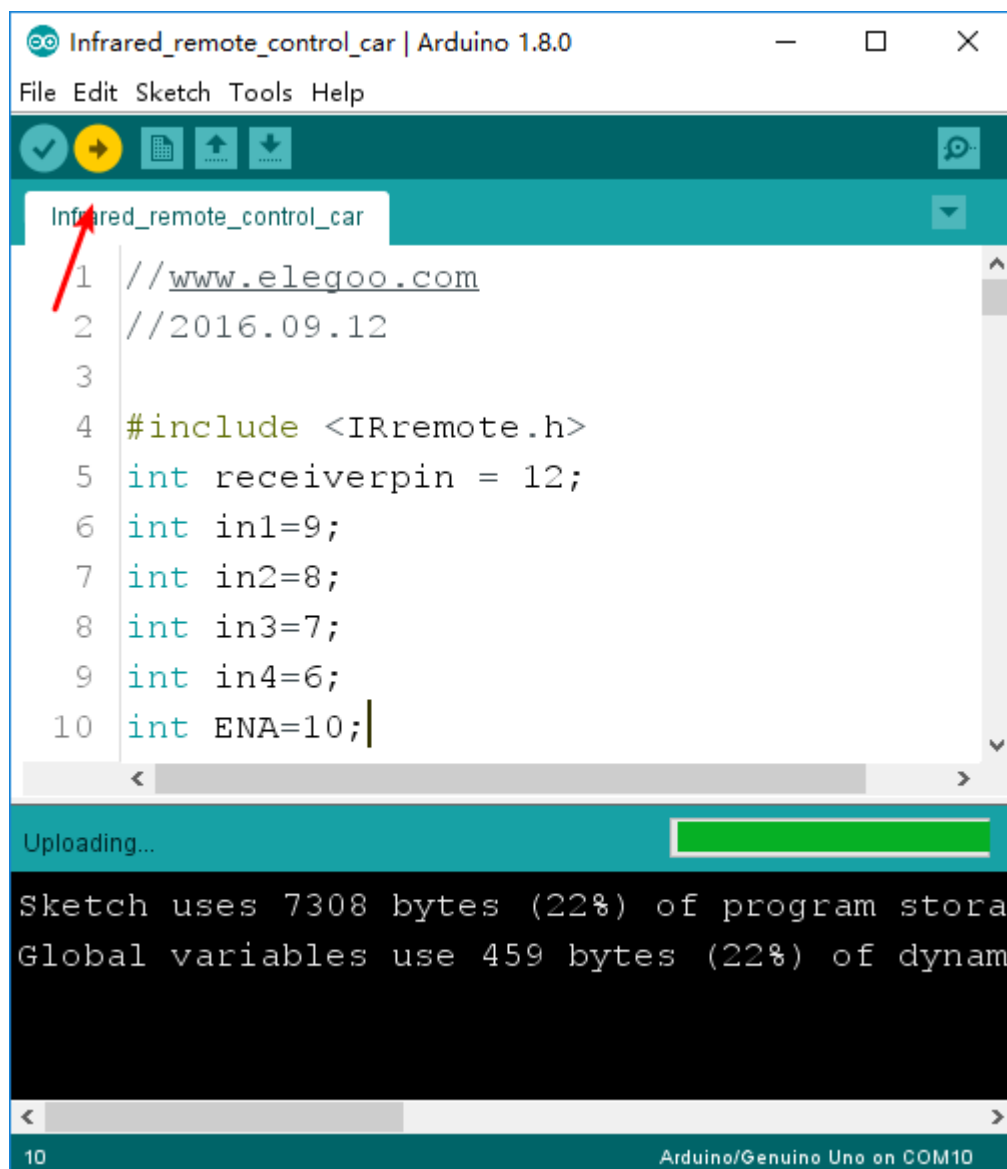
Abriremos el programa -> como abajo Al coche, abierto el fichero  
Infrared\_remote\_control\_car\ Infrared\_remote\_control\_car.ino



Select the Arduino Uno Board and Serial port.



Presione el botón de subir



```

1 //www.elegoo.com
2 //2016.09.12
3
4 #include <IRremote.h>
5 int receiverpin = 12;
6 int in1=9;
7 int in2=8;
8 int in3=7;
9 int in4=6;
10 int ENA=10;
  
```

Uploading...

Sketch uses 7308 bytes (22%) of program storage  
Global variables use 459 bytes (22%) of dynamic memory

10 Arduino/Genuino Uno on COM10

Cuando termine de cargar, desconectando el automóvil al ordenador. A continuación, encienda el interruptor de encendido y poner el coche en el suelo. Presione el botón en el control remoto y usted



puede ver el movimiento del coche según lo que usted ordena.



Voila, ahora puedes jugar con el coche de control de IR felizmente.

El código es como abajo:

```
#include <IRremote.h>
int receiverpin = 12;
int in1=6;
int in2=7;
int in3=8;
int in4=9;
int ENA=5;
int ENB=11;
int ABS=150;
unsigned long RED;
#define A 16736925

#define B 16754775

#define X 16712445

#define C 16720605

#define D 16761405

IRrecv irrecv(receiverpin);
decode_results results;
```

```
void _mForward()
{
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(in1,HIGH); //digital output
    digitalWrite(in2,LOW);
    digitalWrite(in3,LOW);
    digitalWrite(in4,HIGH);
    Serial.println("go forward!");
}

void _mBack()
{
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    Serial.println("go back!");
}

void _mleft()
{
    analogWrite(ENA,ABS);
    analogWrite(ENB,ABS);
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    Serial.println("go left!");
}

void _mright()
{

```

```
    analogWrite(ENA,ABS);
    analogWrite(ENB,ABS);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    digitalWrite(in3,LOW);
    digitalWrite(in4,HIGH);
    Serial.println("go right!");
}

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

void setup() {
    // put your setup code here, to run once:
    pinMode(in1,OUTPUT);
    pinMode(in2,OUTPUT);
    pinMode(in3,OUTPUT);
    pinMode(in4,OUTPUT);
    pinMode(ENA,OUTPUT);
    pinMode(ENB,OUTPUT);
    pinMode(receiverpin,INPUT);
    Serial.begin(9600);
    _mStop();
    irrecv.enableIRIn();
}

void loop() {
    if (irrecv.decode(&results))
    {
        RED=results.value;
```

```
Serial.println(RED);  
irrcv.resume();  
delay(150);  
if(RED==A)  
{  
  _mForward();  
}  
  
else if(RED==B)  
{  
  _mBack();  
}  
  
else if(RED==C)  
{  
  _mleft();  
}  
  
else if(RED==D)  
{  
  _mright();  
}  
  
else if(RED==X)  
{  
  _mStop();  
}  
  
}  
}
```