Lezione 4 Controllo a infrarossi

I punti di questa sezione

Il telecomando a infrarossi è un metodo ampiamente utilizzato per il controllo remoto. L'auto è stata dotata di ricevitore a infrarossi e quindi consente l'uso del telecomando a infrarossi.

Imapreremo

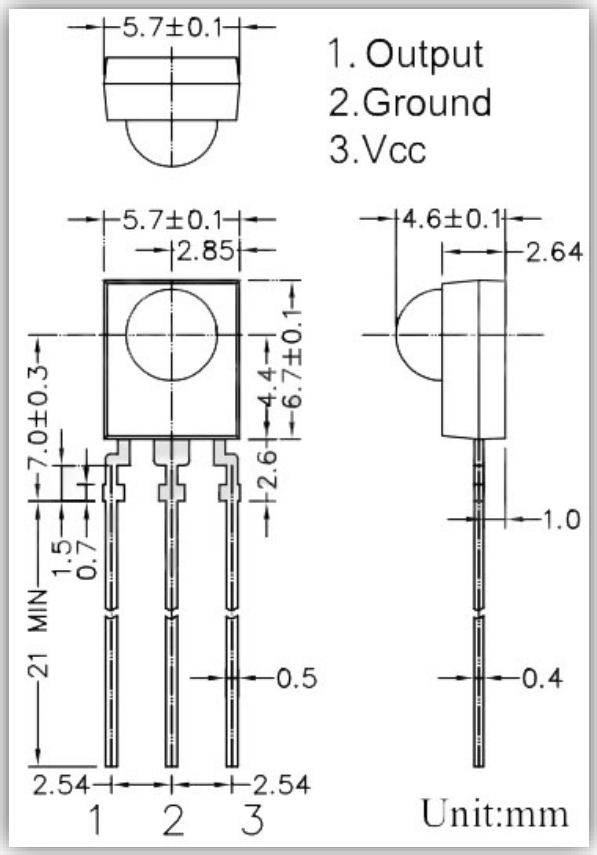
- *Capire il telecomando e il ricevitore infrarossi*
- *Capire I principi del controllo remoto*

Prepara:

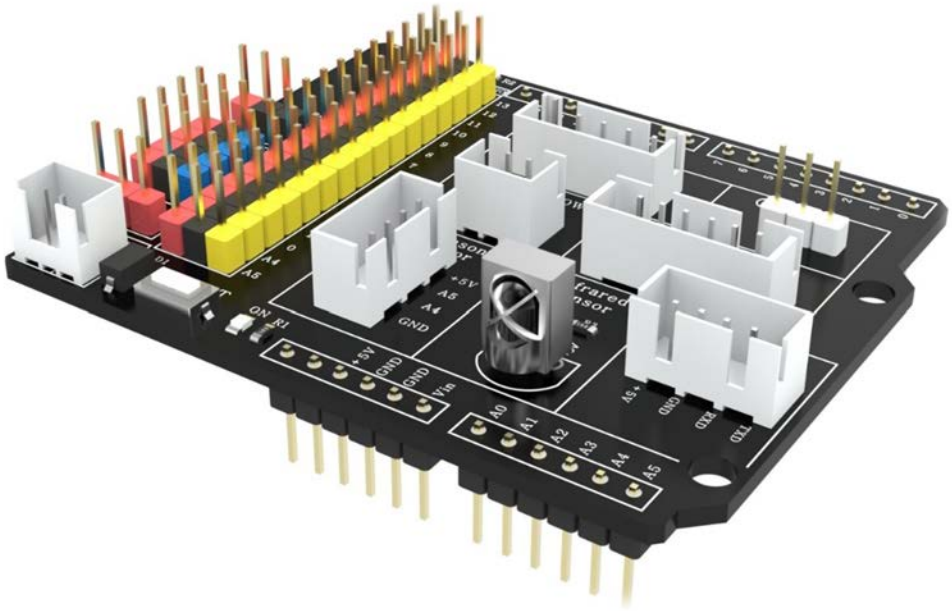
- *L'auto con batteria*
- *Un cavo USB*
- *Modulo ricevitore IR e telecomando IR*

Ⅰ. Modulo ricevitore IR e telecomando IR

I dati del sensore del ricevitore IR sono i seguenti:



La connessione del modulo ricevitore è la seguente:

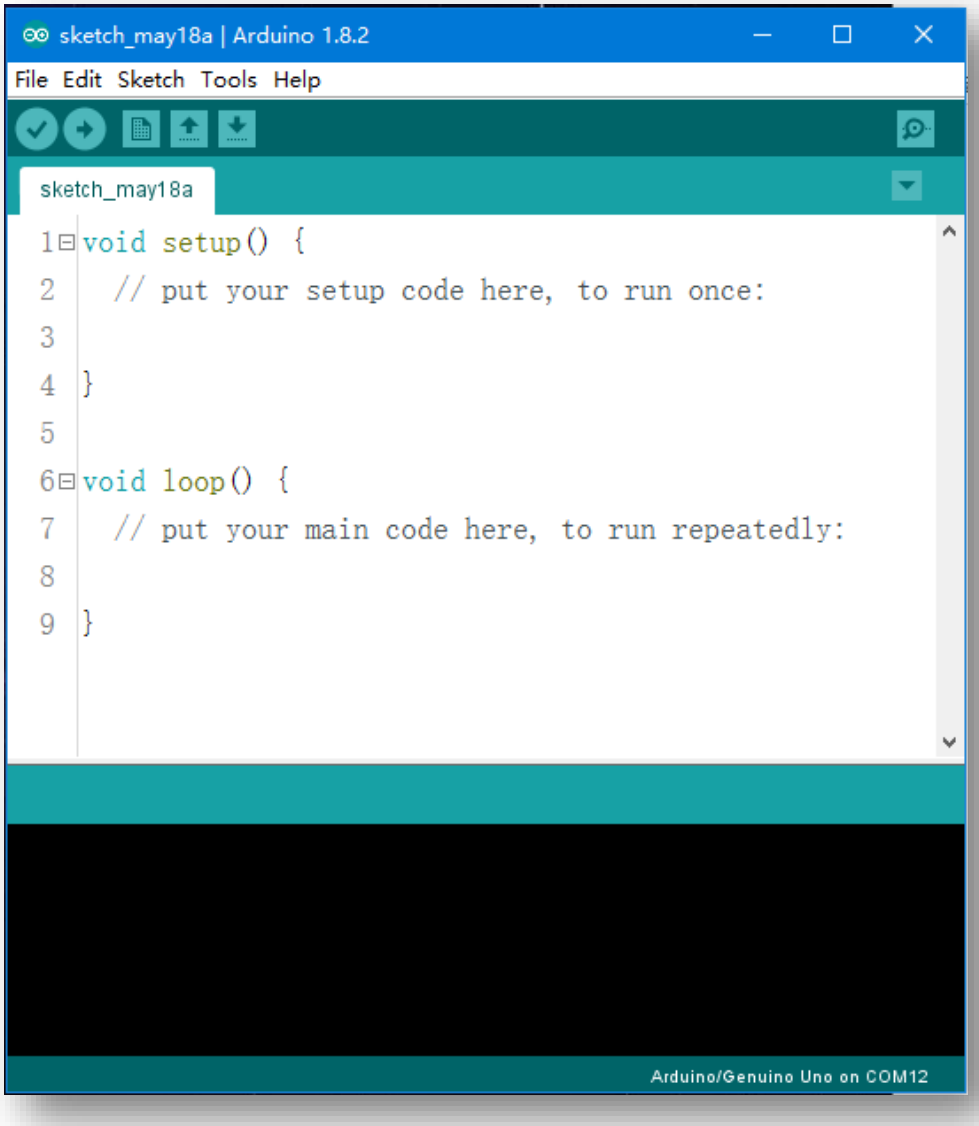


Questo è il telecomando IR:

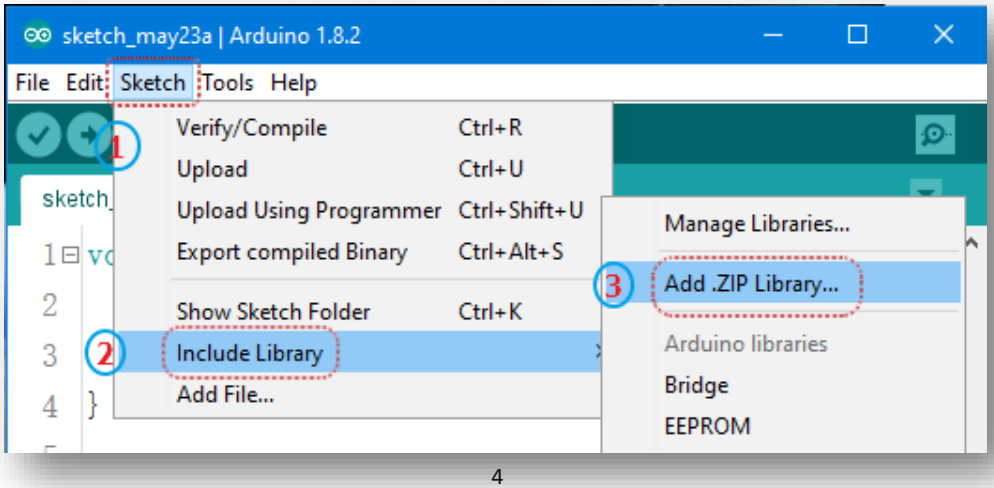
Ⅱ. Testare il programma

Per questo programma dobbiamo usare la libreria aggiungendo prima il file alla libreria

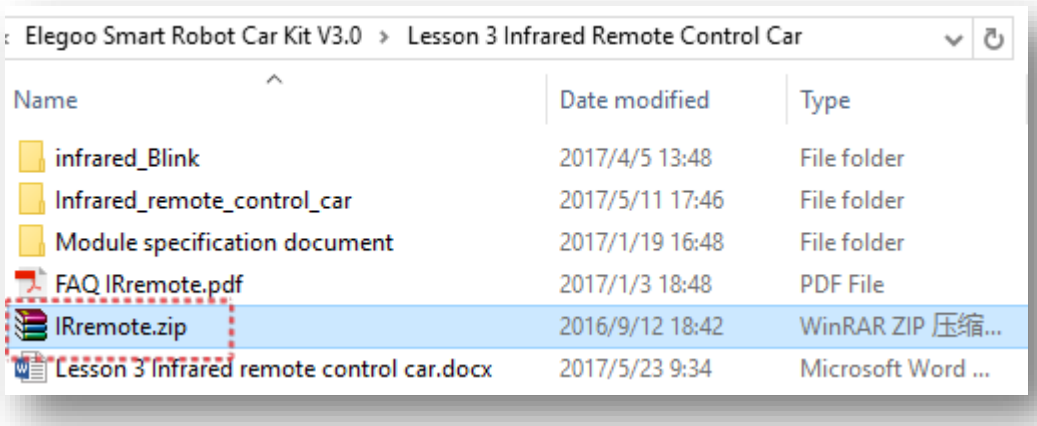
Collega l'UNO al computer e apri l'IDE Arduino.



Clicca Sketch --> Include Library --> Add .ZIP Library... --> seleziona la libreria

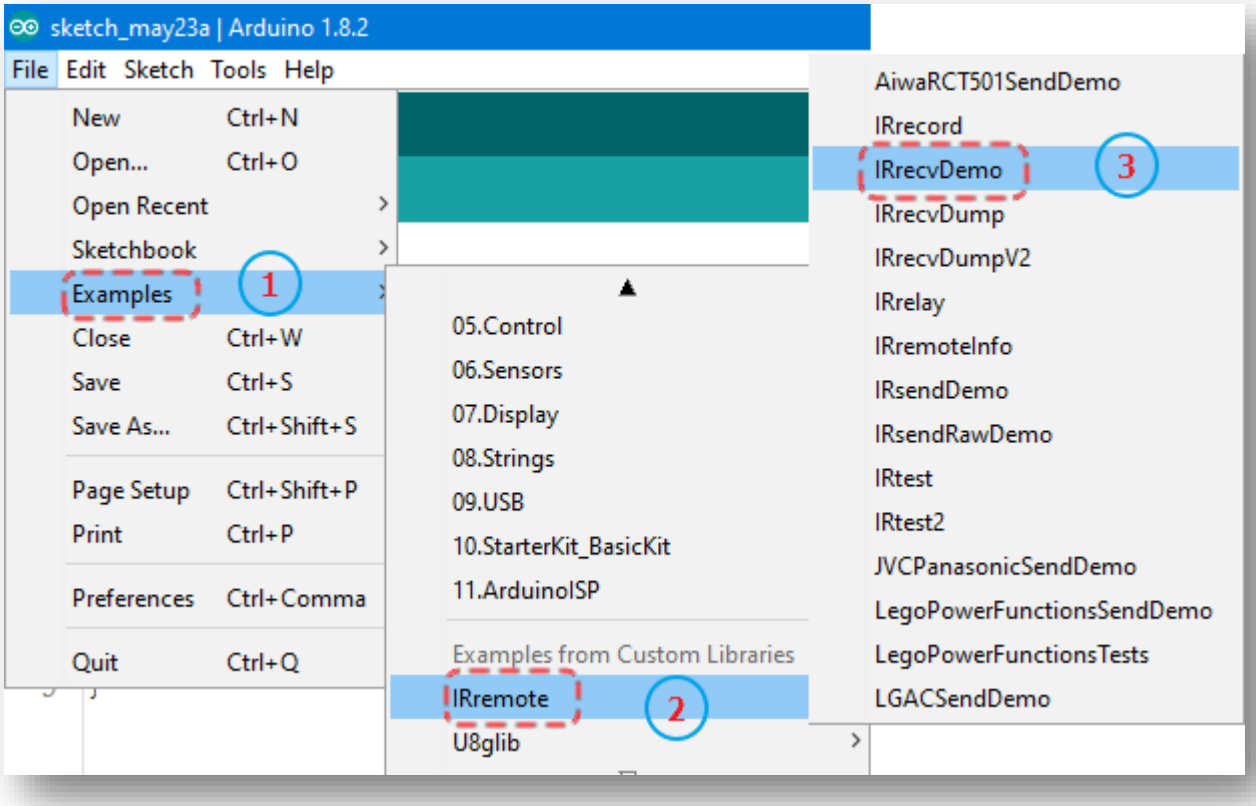


Il none del file della libreria ZIP deve essere IRremote.zip.

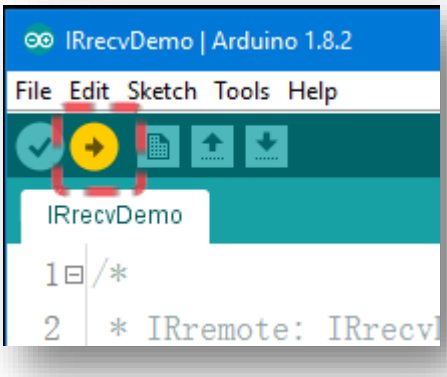


Deve essere compilato con questo file di libreria, che è un file di libreria appositamente modificato.

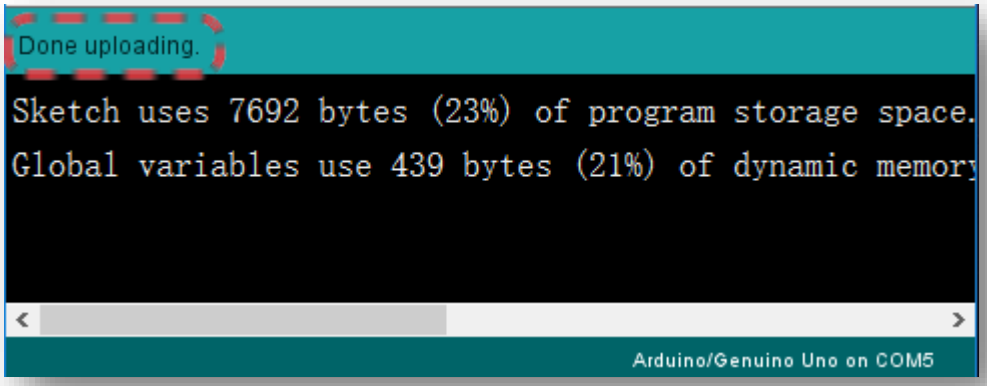
Seleziona IRremote da esempi



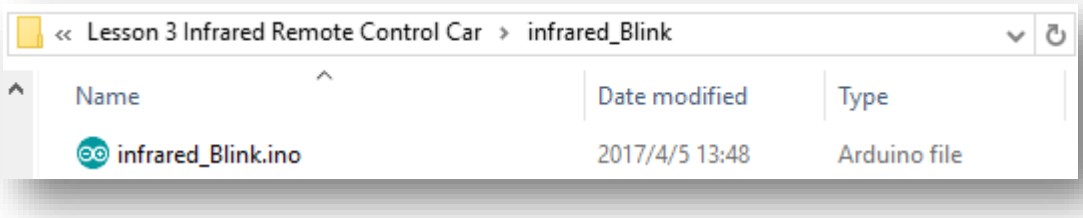
Click pulsante compila



Fallo compilare. In caso contrario, la libreria IRremote non è stata installata correttamente. Per favore vai ad aggiungere di nuovo la libreria IRremote.



Apri il codice dall'indirizzo"\Elegoo Smart Robot Car Kit V3.0\Lesson 4 Infrared Remote Control Car\infrared_Blink\infrared_Blink.ino" e carica il porgramma alla controller board.



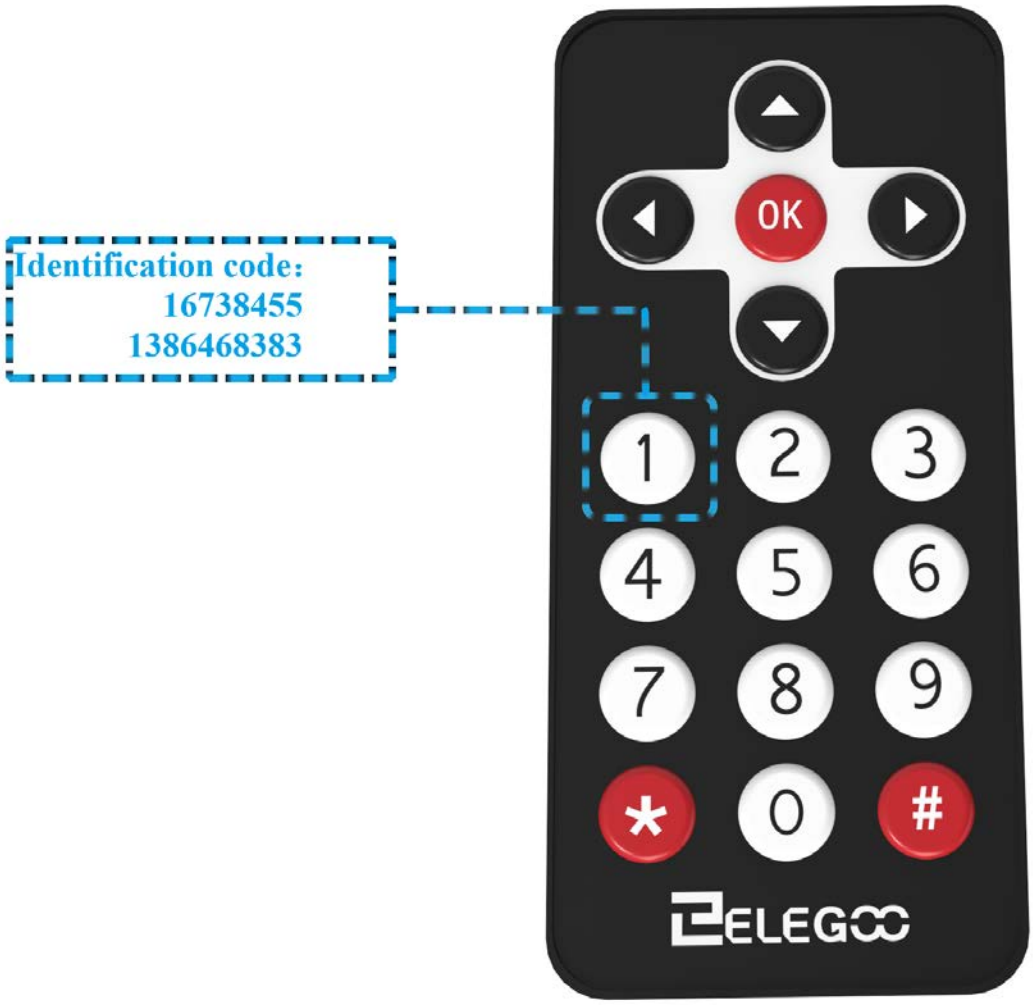
Code preview :

```
//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();
    }
  }
}
```

Dopo aver scollegato l'auto dal computer, è possibile accendere l'interruttore di alimentazione e mettere l'auto a terra.

Premi il pulsante "1" rivolto verso l'auto, osserva l'auto, e troverai il LED con l'etichetta "L" sulla scheda di estensione spegni.



Ⅲ. Introduzione ai principi

1 Funzionamento principale

Il sistema di controllo remoto a infrarossi universale è composto da due parti: l'invio e la ricezione, la parte di invio è costituita da un telecomando IR, la parte ricevente è costituita da un canale di ricezione a infrarossi. I segnali inviati dal telecomando a infrarossi sono una serie di codici di impulsi binari. Per essere liberi dall'interferenza di altri segnali a infrarossi durante il trasporto wireless, è generalmente necessario modularlo a una determinata frequenza portante, quindi lanciarlo attraverso un fototransistor emesso all'infrarosso. Il canale di ricezione a infrarossi filtra le altre onde di rumore, riceve solo i segnali di una determinata frequenza e li ripristina al codice di impulsi binari che è la demodulazione. Il canale di ricezione incorporato trasforma i segnali luminosi inviati dal diodo emettitore di luce infrarossa a segnali elettrici deboli, i segnali vengono ingranditi attraverso l'amplificatore all'interno di IC e attraverso il controllo automatico del guadagno, filtraggio passa-banda, demodulazione, modellatura dell'onda e ripristino all'originale codifica inviata tramite il telecomando, riconoscere il circuito tramite la codifica che viene immessa nell'elettrodomestico tramite il pin di uscita del segnale del modulo di ricezione a infrarossi.

2. Protocollo del telecomando a infrarossi

Lo schema di codifica del controllo remoto IR abbinato è: protocollo NEC.

Successivamente, impariamo quale sia il protocollo NEC.

Caratteristiche:

(1) 8 bit di indirizzo, 8 bit di ordine

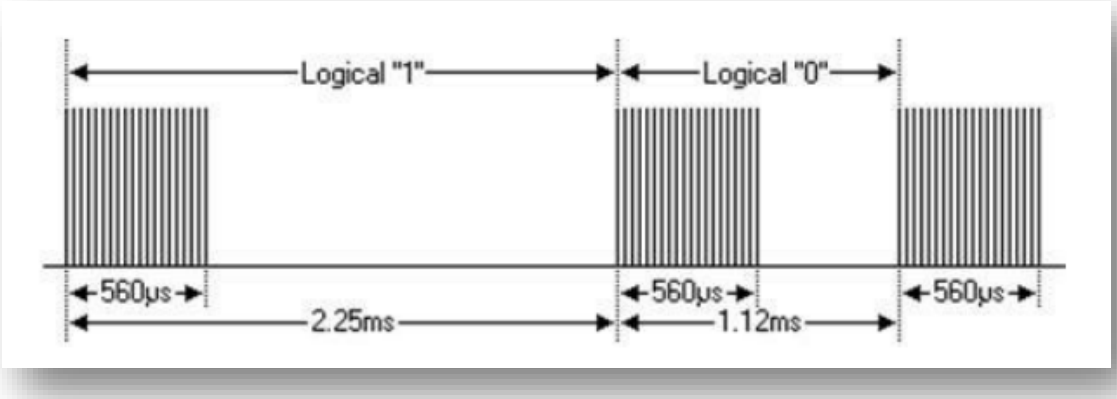
(2) Il bit di indirizzo e il bit di ordine vengono trasmessi due volte per garantire l'affidabilità

(3) Modulazione della posizione dell'impulso

(4) La frequenza portante è 38kHz

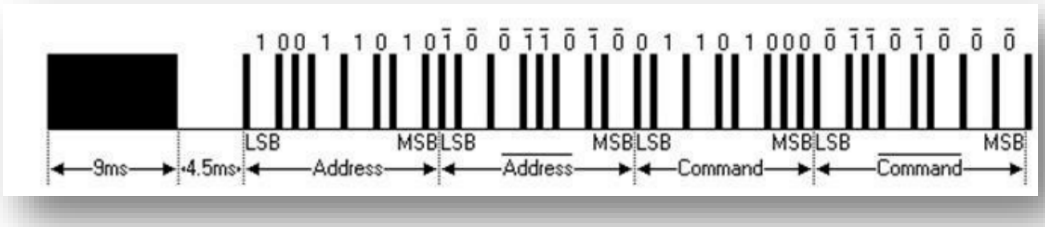
(5) Il tempo di ogni bit è 1.125ms o 2.25ms

Di seguito la definizione di logica 0 e 1



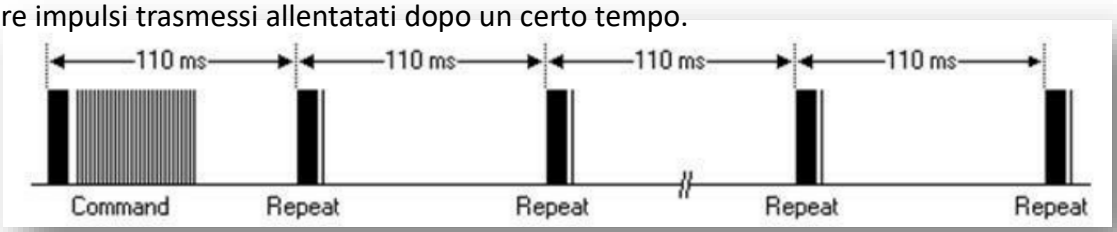
Il protocollo è il seguente

Premere istantaneamente allenta l'impulso di trasmissione:



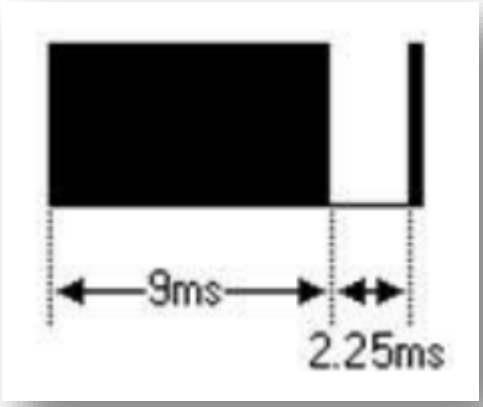
Nota: questo è il protocollo di invio LSB (bit meno significativo) in primo luogo. L'indirizzo di trasporto dell'impulso precedente è 0x59, l'ordine è 0x16. Un messaggio inizia da un alto livello di 9 ms, il livello seguente è un basso livello di 4,5 ms (codice di guida per le forme a due livelli) e tramite il codice di indirizzo e il codice di ordine. Indirizzo e ordine vengono trasmessi due volte. Nella seconda volta, tutti i bit sono invertiti l'opposto, può essere utilizzato per confermare i messaggi di ricezione da utilizzare. Il tempo di invio totale è fisso, se non ti interessa, puoi ignorare l'affidabilità dell'invert, e puoi espandere l'indirizzo e l'ordine a 16 bit! Perché il fatto che la lunghezza si ripete se ogni bit è opposto.

Premere impulsi trasmessi allentatati dopo un certo tempo.



Una volta inviato un comando, anche se viene premuto il pulsante di controllo remoto. Quando il pulsante è ancora premuto, l'impulso dei primi 110 ms è diverso da quello precedente, il codice duplicato viene trasmesso ogni 110 ms. Il codice duplicato è costituito da un impulso di alto livello di 9 ms e un livello basso di 2,25 e un livello elevato di 560 µs.

Impulsi ripetuti:



Nota: dopo che la forma d'onda dell'impulso entra nell'integrazione del sensore, a causa del fatto che l'integrazione del sensore deve essere decodificata, il segnale ingrandito e la plastica, è necessario notare il momento in cui non ci sono segnali a infrarossi, il suo terminale di uscita è di alto livello, è di basso livello quando ci sono segnali Quindi il livello del segnale di uscita è opposto al terminale di trasmissione. Tutti possono vedere il polso del ricevitore attraverso l'oscilloscopio, capire il programma con la forma d'onda visiva

3. The idea of programming remote control car

Secondo la caratteristica del codice NEC e dell'onda di ricezione-fine, questo esperimento divide l'onda di ricezione-fine in quattro parti: codice principale (Pulse of 9ms e 4.5ms), codice di indirizzo (incluso il codice di indirizzo a 8 bit e 8 bit indirizzo di recupero),

Codice indirizzo a 16 bit (incluso codice indirizzo a 8 bit e recupero indirizzo a 8 bit), codice ordine a 16 bit (incluso codice ordine a 8 bit e recupero ordine a 8 bit), codice ripetuto (essere composto da un impulso di 9 ms , 2.25ms, 560us).

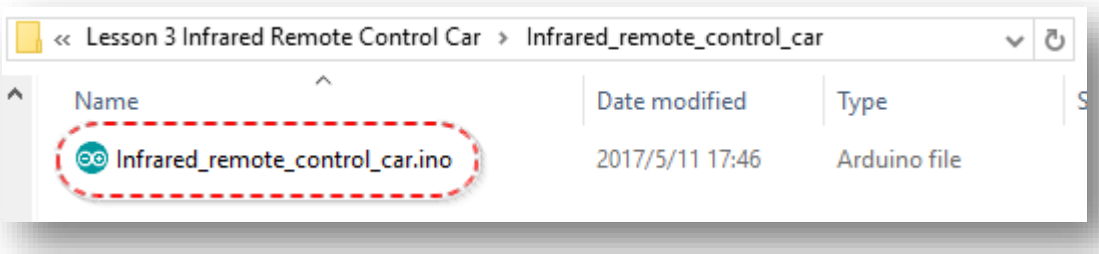
Sfruttare il timer per testare l'alto livello e il basso livello di onda ricevuto, distinguendosi in base al tempo testato: logico "01", logico "1", impulso iniziale, impulso ripetuto. Il codice principale e il codice indirizzo sono giudicati corretti, non archiviati, poiché il codice d'ordine di ciascuna chiave è diverso, l'azione viene eseguita per codice d'ordine.

Durante l'esperimento in auto, abbiamo solo bisogno di controllare l'auto per andare avanti e indietro, girare a sinistra e destra, e fermarsi, il che significa che avremmo bisogno di 5 chiavi e il loro valore è il seguente:

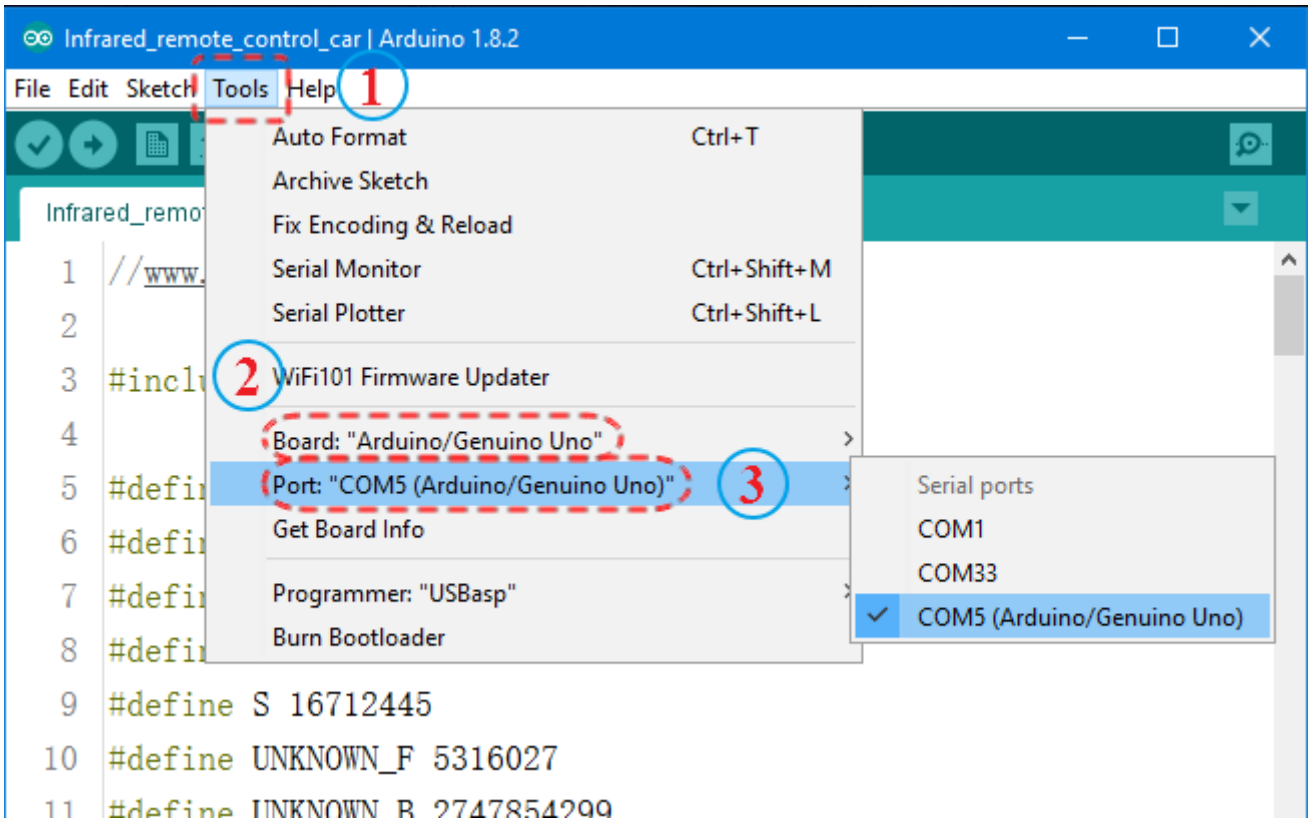
Remote control character	key value
Pulsante rosso al centro	16712445, 3622325019
Freccia su	16736925, 5316027
Freccia giù	16754775, 2747854299
Freccia sinistra	16720605, 1386468383
Freccia destra	16761405, 553536955

IV. Creare un'auto telecomandata

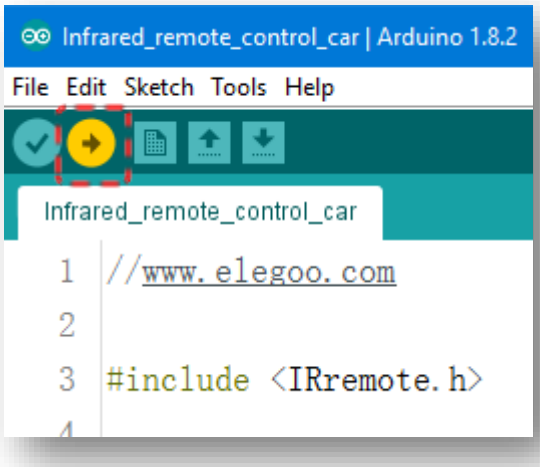
Aprire Il codice dal percorso"\Elegoo Smart Robot Car Kit V3.0\Infrared_remote_control_car\ Infrared_remote_control_car.ino" e caricarlo sull'auto come qui sotto



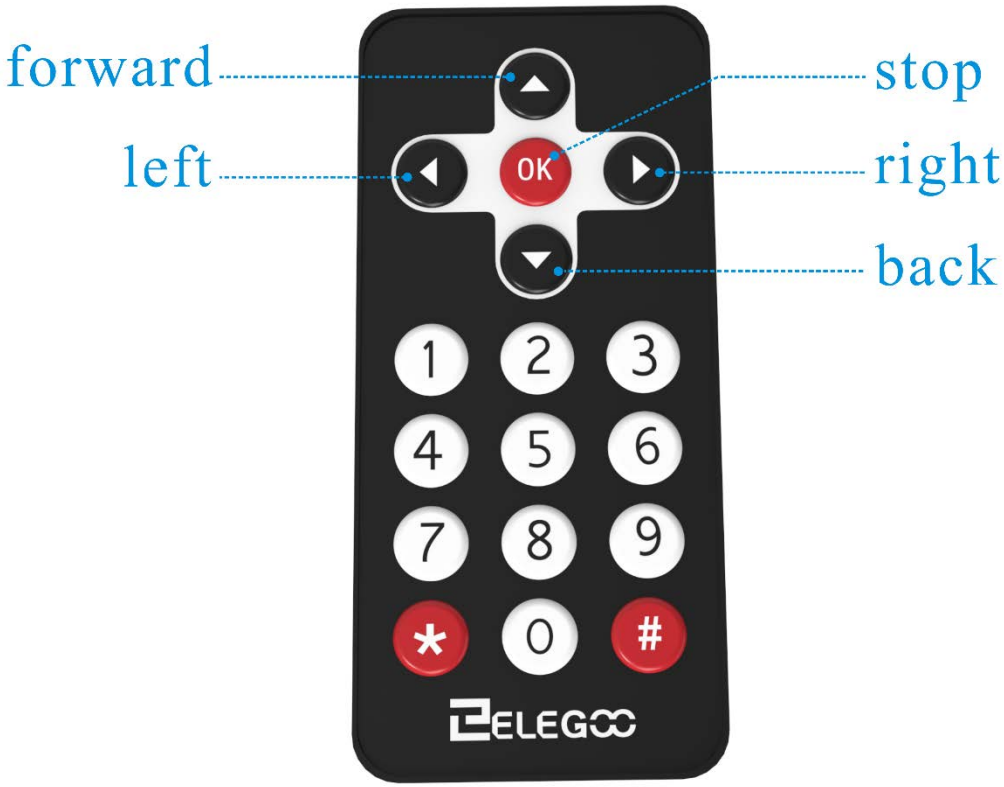
Seleziona Arduino Uno Board e Serial port.



Premi il pulsante di upload



Al termine del caricamento, scollegare l'automobile dal computer. Quindi accendere l'interruttore di alimentazione e mettere l'auto a terra. Premi il pulsante sul telecomando e puoi vedere l'auto spostarsi in relazione ai comandi.



Voilà, ora puoi giocare felicemente con il telecomando IR

Code preview:

```
//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();
    }
  }
}
```