

Leçon 4 – Contrôle voiture Infrarouge

Point abordés dans cette section

La télécommande infrarouge est une méthode largement utilisée pour la commande à distance.

La voiture a été équipée d'un récepteur infrarouge et permet ainsi de le contrôler à l'aide de la télécommande infrarouge.

Apprentissage

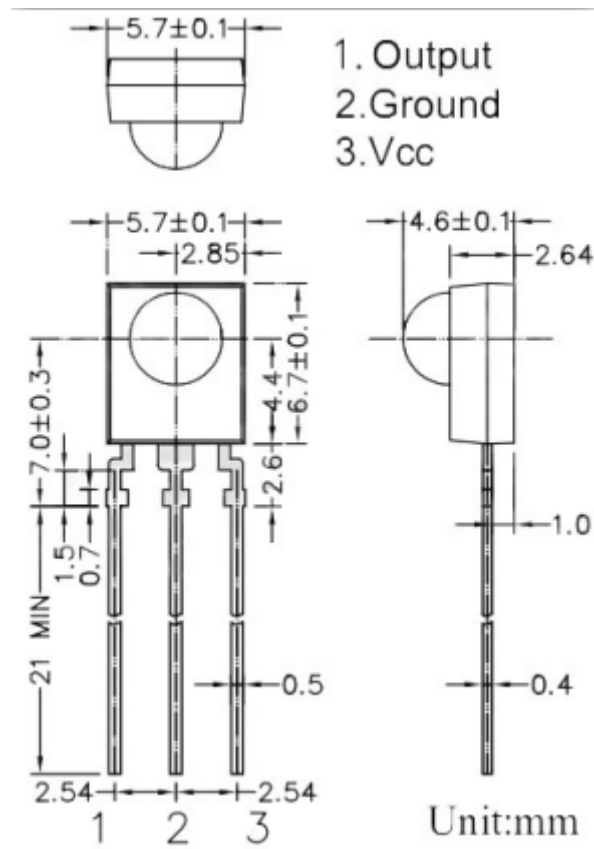
- *Comprendre la télécommande infrarouge et le récepteur*
- *Comprendre les principes de la télécommande*

Préparation

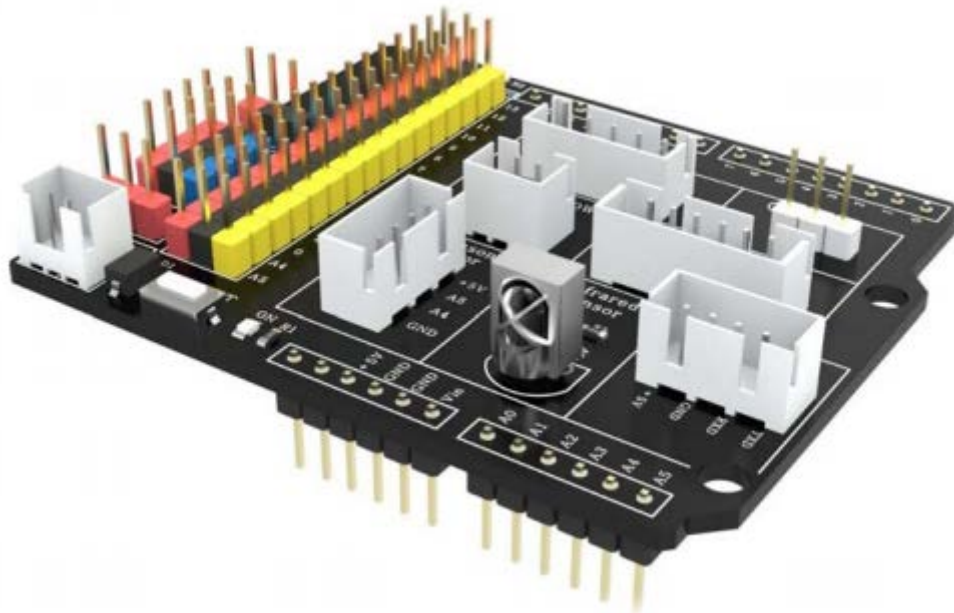
- *Un véhicule (équipé d'une batterie)*
- *Un câble USB*
- *Module de réception infrarouge et télécommande IR*

I – Module de réception infrarouge et télécommande infrarouge

Les données du capteur du récepteur IR sont les suivantes:



La connexion du module récepteur est comme suit:



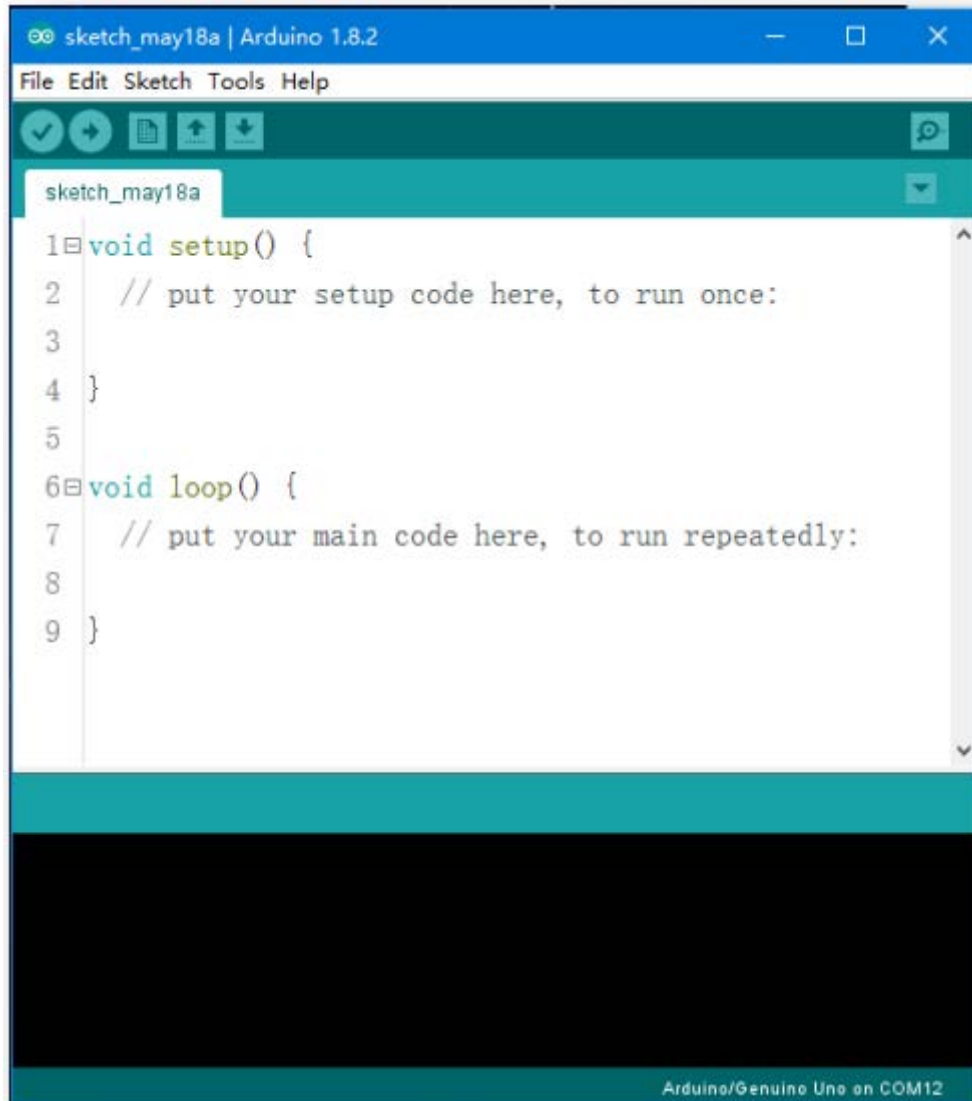
C'est une télécommande infrarouge:



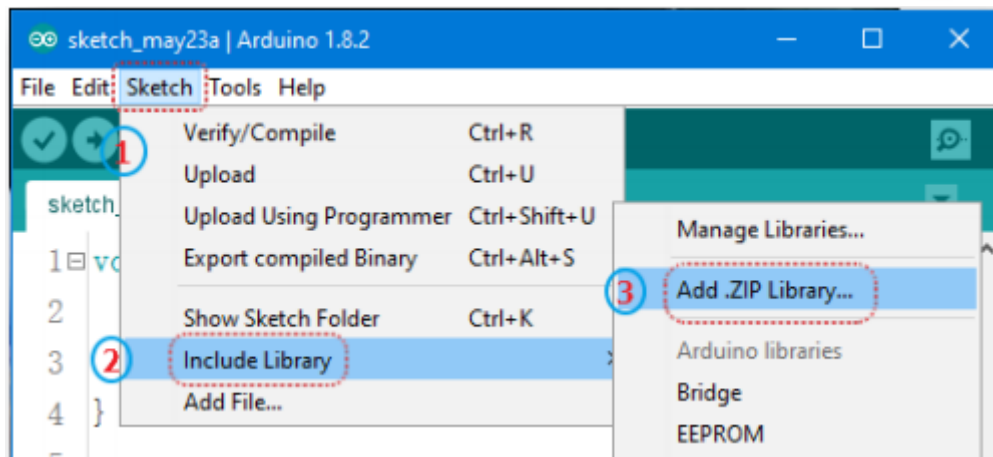
I – Programme de test

Parce que dans ce programme, nous devons utiliser la bibliothèque pour le recpeteur infrarouge. Nous allons avoir besoin d'ajouter le fichier de bibliothèque au début.

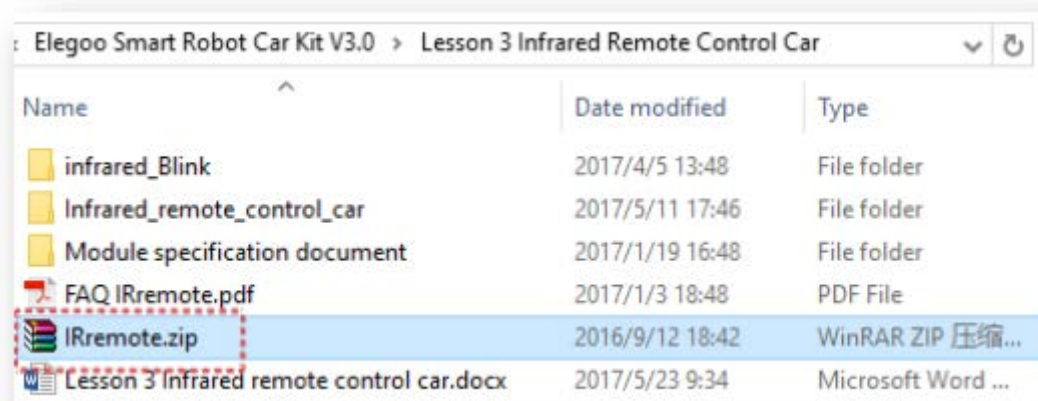
Connectez l'UNO à l'ordinateur et ouvrez l'IDE Arduino.



Cliquez sur Sketch -> Include Library -> Ajouter .ZIP Library ... -> puis sélectionnez la bibliothèque comme indiqué ce dessous.

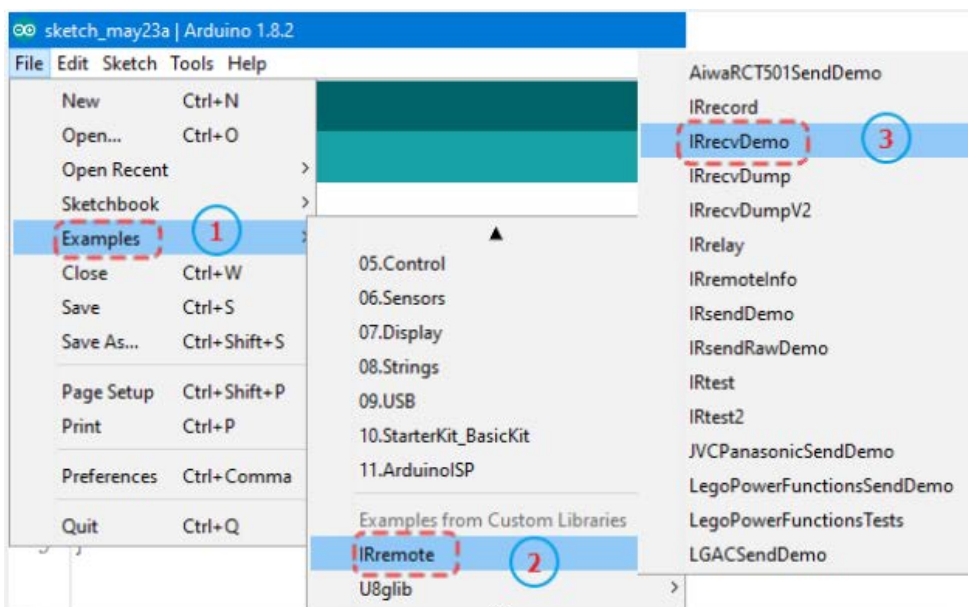


Le nom de fichier de la bibliothèque ZIP doit être IRremote.zip.

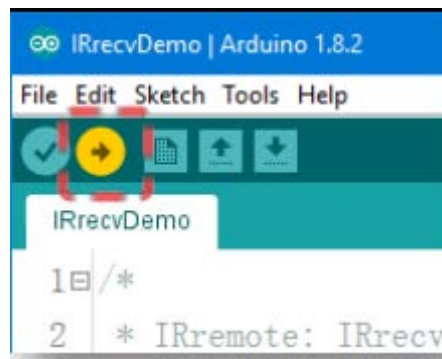


Doit être compilé avec ce fichier de bibliothèque, qui est un fichier de bibliothèque spécialement modifié.

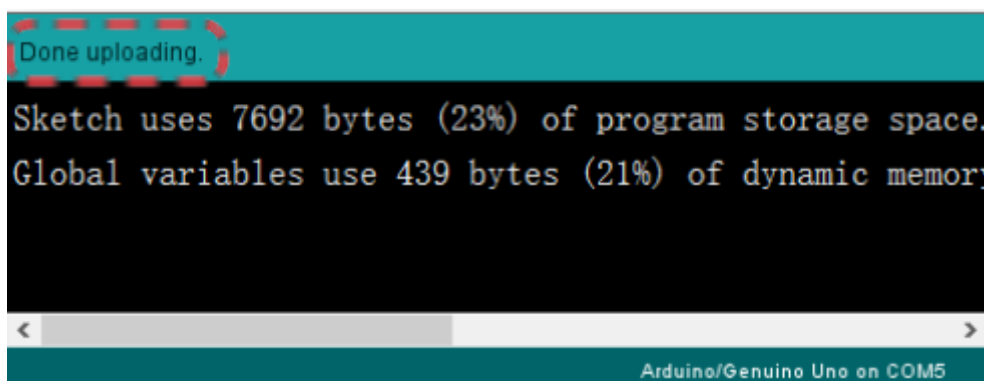
Sélectionnez un exemple d'IRremote



Cliquer sur le bouton compilation



Terminé la compilation. Dans le cas contraire, la bibliothèque IRremote n'a pas été installée avec succès. Veuillez ajouter à nouveau la bibliothèque IRremote.



Ouvrez le fichier de code dans le chemin "\\ Elegoo Smart Robot Car Kit V3.0 \\ Leçon 4 Contrôle voiture infrarouge \\ infrared_Blink \\ infrared_Blink.ino" et téléchargez le programme sur la carte contrôleur.



//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
```

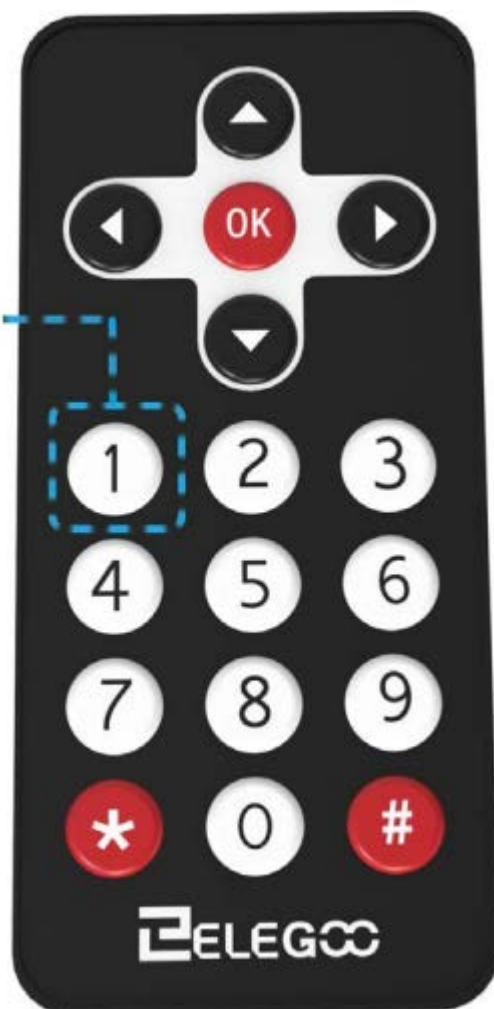
```
decode_results results; //Define structure type
```

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

Après avoir débranché la voiture sur l'ordinateur, vous pouvez allumer l'interrupteur d'alimentation et mettre la voiture sur le sol.

Appuyez sur le bouton "1" en face de la voiture, observez la voiture et vous trouverez la LED avec une étiquette "L" sur la carte d'extension s'éteindre.

Identification code:
16738455
1386468383



III – Introduction du principe

1 - Principe de fonctionnement

Le système de contrôle à distance infrarouge universel se compose de deux parties: l'envoi et la réception, la partie d'envoi se compose d'une télécommande IR, la partie de réception se compose d'un tube récepteur infrarouge. Les signaux envoyés par télécommande infrarouge sont une série de code d'impulsions binaires. Afin de ne pas être perturbé par d'autres signaux infrarouges pendant le transport sans fil, il est en général modulé à une fréquence de données porteuse, puis lancé à travers un phototransistor à émission infrarouge. Le tube de réception infrarouge filtre les autres ondes de bruit, ne reçoit que des signaux de fréquence donnée et les restaure au code d'impulsion binaire qui est la démodulation. Les signaux lumineux transformée par le tube de réception intégrés sont envoyés à la diode électroluminescente infrarouge en signaux électriques faibles. Les signaux sont agrandis par l'amplificateur à l'intérieur de l'IC, et par le contrôle automatique du gain, le filtrage passe-bande, la démodulation, la mise en forme de l'onde permettent de restaurer le signal original envoyé par la télécommande, est reconnu puis envoyé à la carte ARDUINO par l'intermédiaire d'une broche de sortie de signal du module de réception infrarouge.

2 - Protocole de contrôle à distance infrarouge

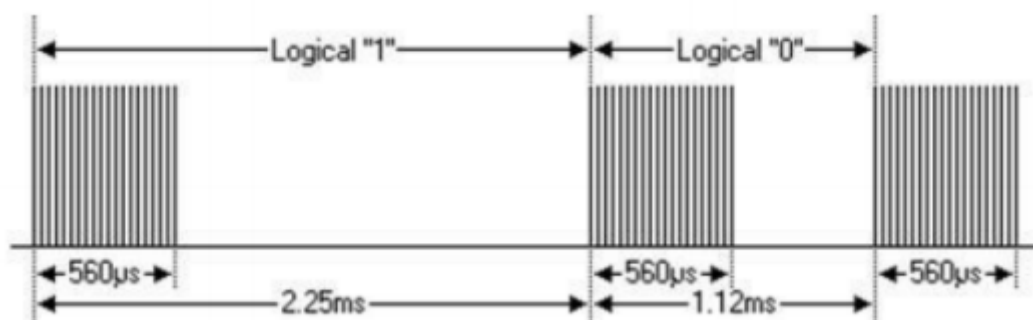
Le schéma de codage de la télécommande infrarouge assortie est: protocole NEC.

Ensuite, apprenons ce qu'est le protocole NEC.

Caractéristiques:

- (1) 8 bit d'adresse, 8 bit de commande
- (2) Le bit d'adresse et le bit de commande sont transmis deux fois afin de garantir la fiabilité
- (3) Modulation de position d'impulsion
- (4) La fréquence du transporteur est de 38kHz
- (5) L'heure de chaque bit est de 1.125ms ou 2.25ms

Les définitions des logiques 0 et 1 sont les suivantes:



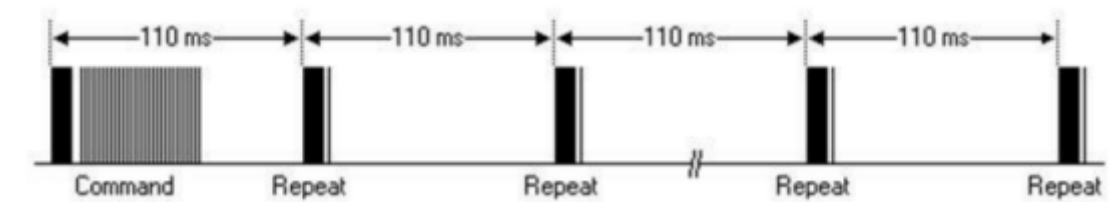
Le protocole est comme suit:

Appuyez sur le déclenchement instantané de la impulsion de transmission:



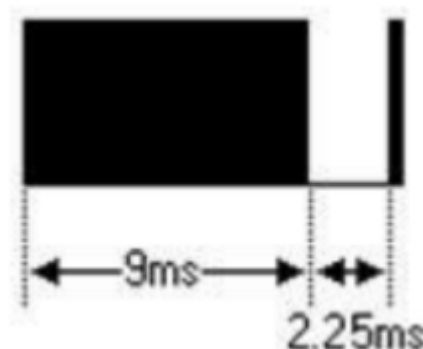
Remarque: Ceci est le protocole d'envoi de LSB (bit d'importance minimale). L'adresse de transport de l'impulsion ci-dessus est 0x59, l'ordre est 0x16. Un message commence à partir d'un niveau élevé de 9ms, le suivi est un faible niveau de 4.5ms, (code d'orientation des deux niveaux) et par le code d'adresse et le code de commande. L'adresse et l'ordre sont transmis deux fois. La deuxième fois, tous les bits sont inversés et peuvent être utilisés pour confirmer les messages de réception à utiliser. Le temps d'envoi total est fixé, si vous n'êtes pas intéressé, vous pouvez ignorer la fiabilité de l'inversion et vous pouvez donc étendre l'adresse et commander à 16 bits!

Appuyez sur l'impulsion et après un certain temps, relachez.



Une fois qu'une commande a été envoyée, si vous pouvez continuer à appuyer sur le bouton de la télécommande. Dans ce cas, lorsque le bouton est encore pressé, l'impulsion des 110 premières ms est différente de celle ci-dessus, le code dupliqué est transmis toutes les 110 ms. Le code en double est constitué d'une impulsion de niveau élevé de 9 ms et d'un faible niveau de 2,25 et d'un niveau élevé de 560 μ s.

Pulsion répétée:



Remarque: Une fois que la forme d'onde impulsionnelle entre dans l'intégration du capteur, en raison du fait que l'intégration du capteur doit être décodé, signal amplifié, vous devez noter le moment où il n'y a pas de signaux infrarouges, sa borne de sortie est de haut niveau, et est de niveau bas lorsqu'il y a des signaux. Le niveau du signal de sortie est donc opposé au terminal de transmission. Tout le monde peut voir le récepteur pulser par oscilloscope, comprendre le programme avec une forme d'onde.

3 – Principe de programmation de la voiture de télécommande

Selon la caractéristique du code NEC et de l'onde de réception, cette expérience divise l'onde de réception en quatre parties:

- code principal (impulsion de 9ms et 4.5ms)
- code d'adresse (code d'adresse 8 bits et 8 bits) adressage d'adresse de 8 bits (code d'adresse 8 bits et récupération d'adresse 8 bits)
- code de commande 16 bits (code de commande de 8 bits et extraction de commande de 8 bits)
- code de répétition (être composé) d'impulsion de 9ms, 2.25ms, 560us).

Exploitez la minuterie pour tester le niveau élevé et le faible niveau d'onde reçue, qui se distinguant selon le temps testé: logique "01", logique "1", impulsion avant, répétition de l'impulsion. Le code principal et le code d'adresse sont jugés corrects, non stockés, du fait que le code de commande de chaque clé est différent, les actions sont effectuées par le code de commande.

Pendant l'utilisation de la voiture, il suffit de la contrôler pour aller en avant et en arrière, tourner à gauche et à droite, et arrêter, ce qui signifie que nous aurions besoin de 5 touches et la valeur de celles-ci sont les suivantes:

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

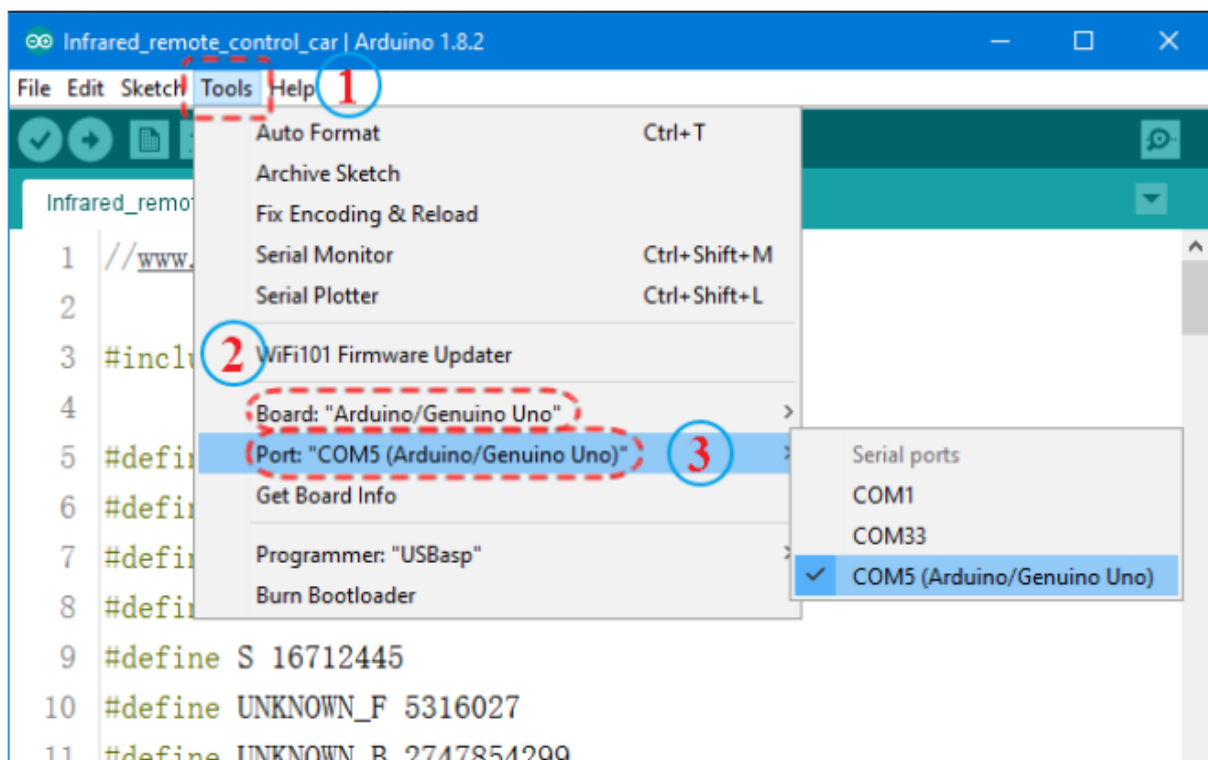


IV – Créer une voiture contrôlée par télécommande

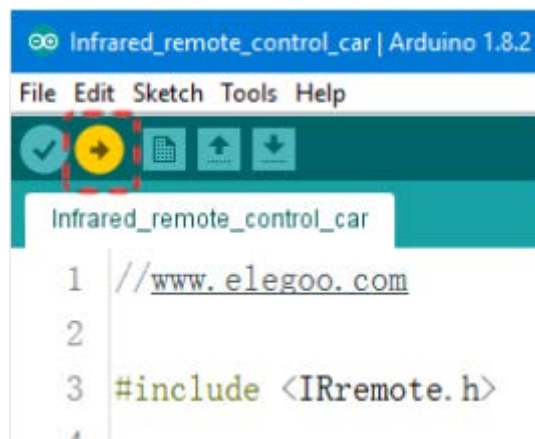
Ouvrez le fichier de code dans le chemin "\\ Elegoo Smart Robot Car Kit V3.0 \ Infrared_remote_control_car \ Infrared_remote_control_car.ino", puis téléchargez le programme comme ci-dessous à la voiture.



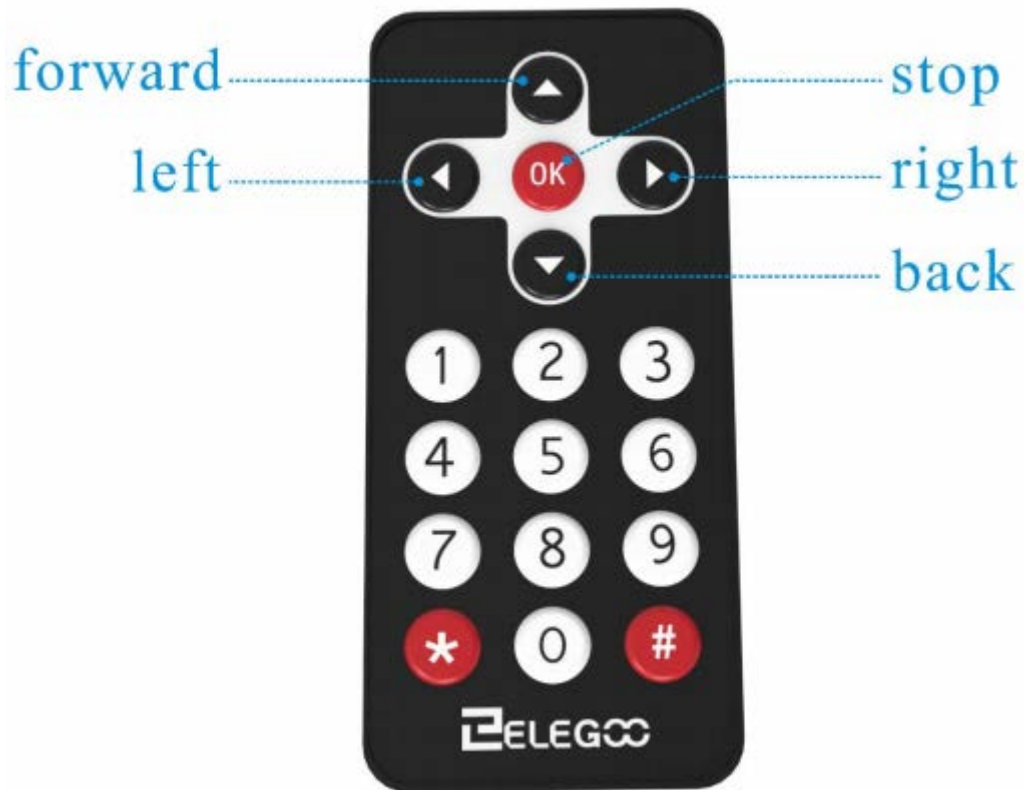
Sélectionnez l'Arduino Uno Board et le port Serial.



Appuyez sur le bouton de téléchargement



Lorsque vous avez téléchargé, débranchez la voiture sur l'ordinateur. Ensuite, allumez l'interrupteur et mettez la voiture au sol. Appuyez sur le bouton sur la télécommande et vous pouvez voir la voiture se déplacer en fonction des touches que vous pressez.



Voilà, vous pouvez maintenant jouer avec la voiture de contrôle IR.

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