

Leçon 3 – Télécommande Infrarouge (Infrared Controlling Car)



Points clés de la leçon

Le contrôle par infrarouge est une méthode largement répandue. Le robot est équipé d'un récepteur infrarouge et permet donc un tel contrôle.

Sommaire:

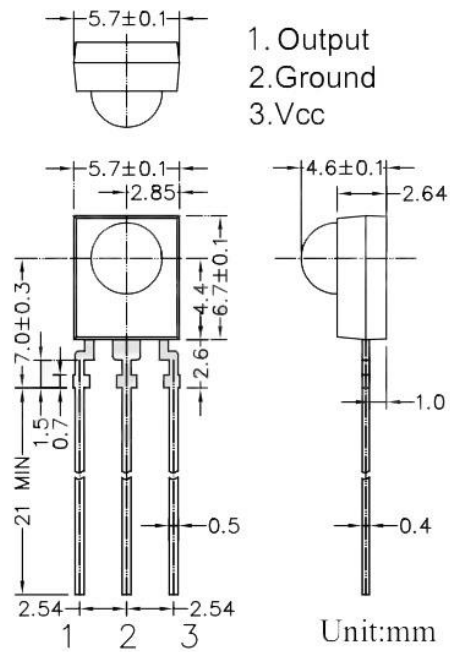
- Comprendre les télécommandes infrarouges et les récepteurs
- Comprendre le principe de fonctionnement

Préparatifs:

- Le robot
- Un câble USB (fourni)
- Une télécommande (fournie)

I . Télécommandes et récepteurs infrarouges

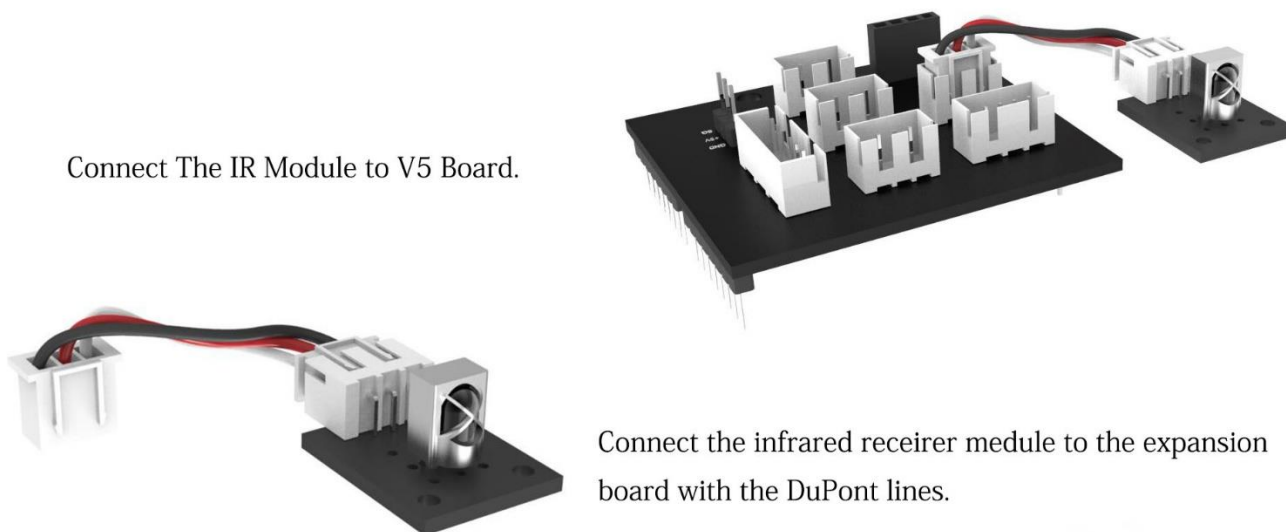
Schéma d'un récepteur infrarouge:



Connection du récepteur sur le robot:



Connect The IR Module to V5 Board.



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

Télécommande:



II. Tester le programme

Ce programme utilise une bibliothèque "IRremote.h" dont il convient de faire l'installation au préalable.



Menu "Croquis/Inclure une bibliothèque/Ajouter la bibliothèque .ZIP..."

IRremote	2016/6/11 23:00	WinRAR ZIP 压缩...	33 KB
----------	-----------------	------------------	-------

Ouvrez le fichier "infrared_Blink\infrared_Blink.ino"

U深度U盘 (H:) > 工作项目 > 4WD car tutorial and code > code > 3 , Infrared remote control car > infrared_Blink				
名称	修改日期	类型	大小	
infrared_Blink.ino	2016/1/29 6:16	Arduino Source ...	1 KB	

Le code est le suivant:

```

#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();
    }
  }
}
```

Déversez le code dans la carte Elegoo UNO, débranchez le câble USB, posez le robot au sol. Allumez-le, vous pouvez maintenant contrôler les mouvements du robot avec la télécommande.



III. Principes

1. Principes de fonctionnement

Le contrôle universel par infrarouge se divise en deux parties : l'émission et la réception d'un signal.

Les signaux qui sont émis par la télécommande sont une série d'impulsions codées en binaire. Le signal est modulé en fréquence pour ne pas interférer avec les signaux d'autres systèmes infrarouges.

Le récepteur décode le signal pour en extraire le codage binaire.

La faiblesse du signal, le gain, l'amplification, le filtrage ... sont gérés par des systèmes intégrés aux émetteurs et récepteurs et ce de manière automatique, sans besoin de réglage par l'utilisateur.

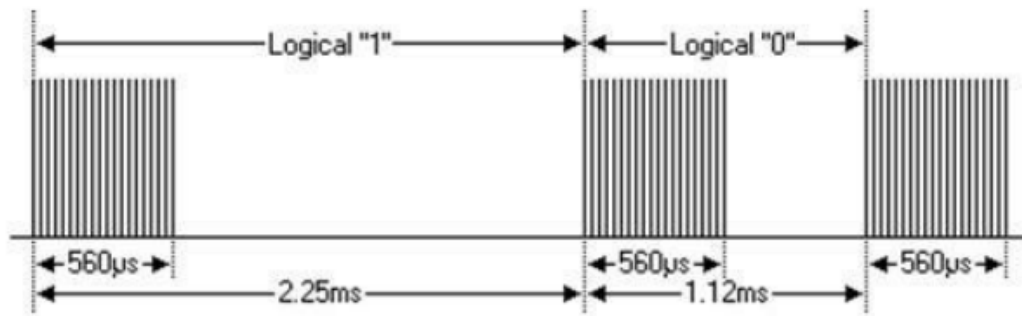
2. Protocole d'échange de données

L'échange de données suit un protocole nommé NEC.

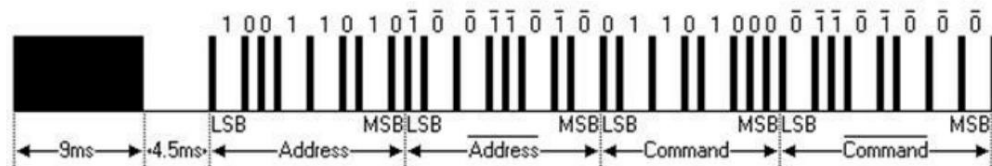
Caractéristiques:

- (1) 8 bits pour l'adresse, 8 bits pour la commande
- (2) Bits d'adresses et bits de commandes sont envoyés deux fois afin de garantir la fiabilité de l'échange.
- (3) Modulation de la position de la pulsation
- (4) Fréquence de la porteuse 38kHz
- (5) Temporisation des bits 1.125ms or 2.25ms

Définition des 0 et 1 logiques :



Protocole:



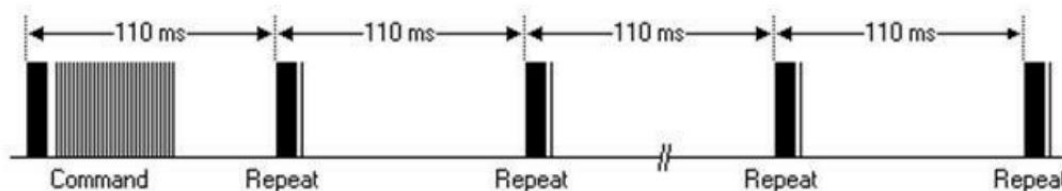
Le protocole utilisé est dit "protocole par envoi de bits non significatifs" en premier : protocole LSB (least-significant bit) .

L'adresse de l'instruction ci-dessus est 0x59

L'ordre de l'instruction ci-dessus est 0x16

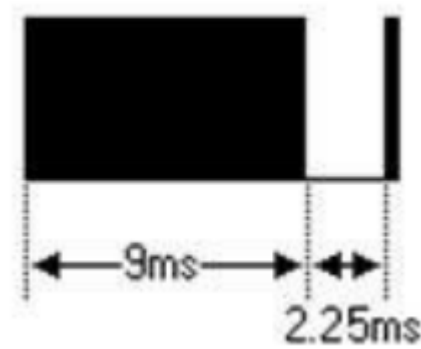
Une instruction débute par un état haut de 9ms puis un état bas de 4,5ms. Suivent l'adresse et l'ordre. L'adresse et l'ordre sont répétés 2 fois. La deuxième fois, les bits sont inversés et peuvent servir à confirmer la bonne réception de l'adresse et de l'ordre. Le temps total de l'impulsion reste fixé. Il est possible de ne pas faire de répétition et/ou d'inversion et donc de passer à 16bits.

Une commande est envoyée à la fois et répétée à interval régulier:



Si le bouton de la télécommande est toujours pressé, l'impulsion est dupliquée toutes les 110ms.

Impulsion répétée: état haut pendant 9ms, état bas pendant 2,25ms.



Note: après l'impulsion, la vague d'onde entre dans le récepteur infrarouge pour y être décodée. Lorsqu'il n'y a pas d'onde reçue, le récepteur renvoi un signal haut à la carte Elegoo UNO et un signal bas lorsqu'une réception est en cours. Le signal reçu est donc opposé à celui émis.

Il est possible d'observer ces signaux sur des oscilloscopes ou des programmes informatiques dédiés.

3. Programmer un robot contrôlé par infrarouge

Comme exposé dans le chapitre précédent, le protocole NEC expose comment coder un ordre en 4 parties :

- pulsion de tête (9ms + 4,5ms)
- code d'adresse (8bits code d'adresse / 8bits code de récupération)
16 bits code d'adresse (8bits code d'adresse / 8bits code de récupération)
- 16 bits code d'ordre (8bits ordre / 8 bits récupération)

Le code est répété en impulsion 9ms + 2,25ms + 560µs

Le temps est exploité pour tester l'état haut (1 logique), l'état bas (0 logique), la pulsion de tête, la pulsion répétée et savoir si le signal est correct ou incorrect.

Pour notre expérimentation, nous devons juste permettre au robot d'aller en avant, en arrière, à droite, à gauche, s'arrêter: soit 5 commandes.

Télécommande	Valeur émise
Bouton OK	16712445
Triangle supérieur	16736925
Triangle inférieur	16754775
Triangle gauche	16720605
Triangle droite	16761405



IV. Un robot commandé par infrarouge

Ouvrez le fichier "Infrared_remote_control_car\ Infrared_remote_control_car.ino" et déversez le code sur la carte Elegoo UNO:

```
#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

#define E 5316027

#define F 2747854299

#define Y 3622325019

#define G 1386468383

#define H 553536955
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);
irrecv.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==E)
```

```
{
```

```
    _mForward();
```

```
}
```

```
else if(RED==F)
```

```
{
```

```
    _mBack();
```

```
}
```

```
else if(RED==G)
```

```
{
```

```
    _mleft();
```

```
}
```

```
else if(RED==H)
```

```
{
```

```
    _mright();
```

```
}
```

```
else if(RED==X)
```

```
{
```

```
        _mStop();  
    }  
    else if (RED == Y)  
    {  
        _mStop();  
    }  
  
}  
}
```

Débranchez le câble USB, posez le robot sur le sol, mettez-le en marche. Vous commandez ses déplacements via la télécommande infrarouge.