

Lesson 3 赤外線コントロールカー



セクションのポイント

赤外線リモコンは、遠隔操作のために広く使用されている方法です。車には赤外線受信機が装備されているので、赤外線リモコンを使って制御することができます。

学習パート：

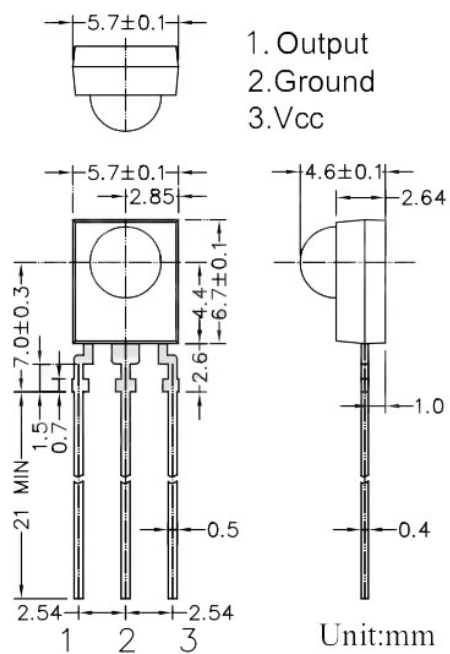
- ◆ 赤外線リモコンと受信機を理解する
- ◆ リモートコントロールの原理を理解する

準備：

- ◆ 車（バッテリー付き）
- ◆ USB ケーブル
- ◆ IR 受信モジュールと IR リモート

I. IR 受信モジュールと IR リモート

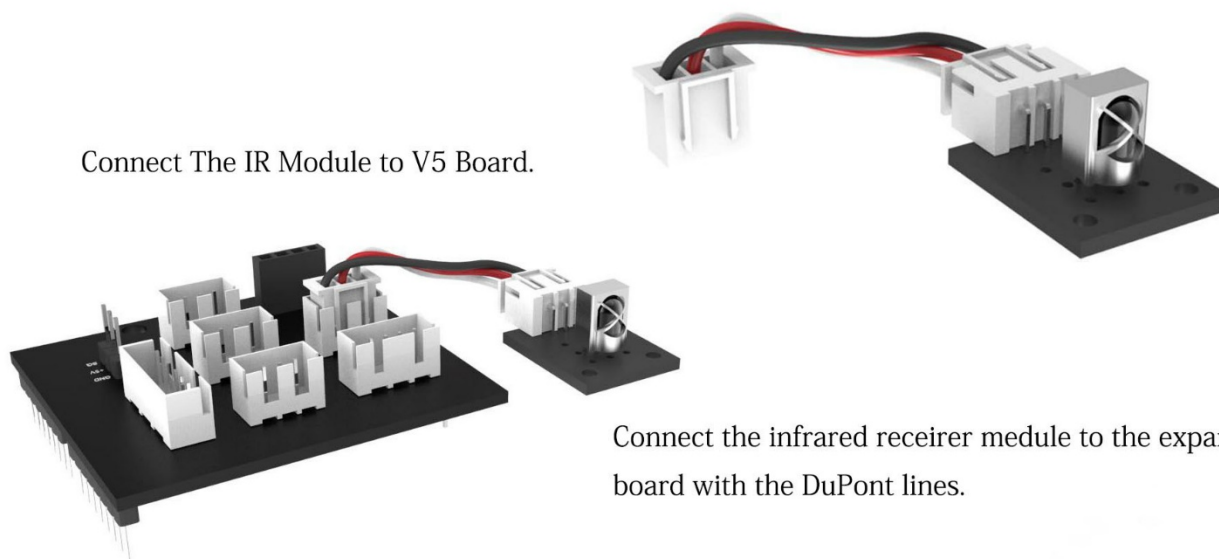
IR 受信機センサーのデータは以下の通りです:



受信モジュールの接続は以下の通りです:

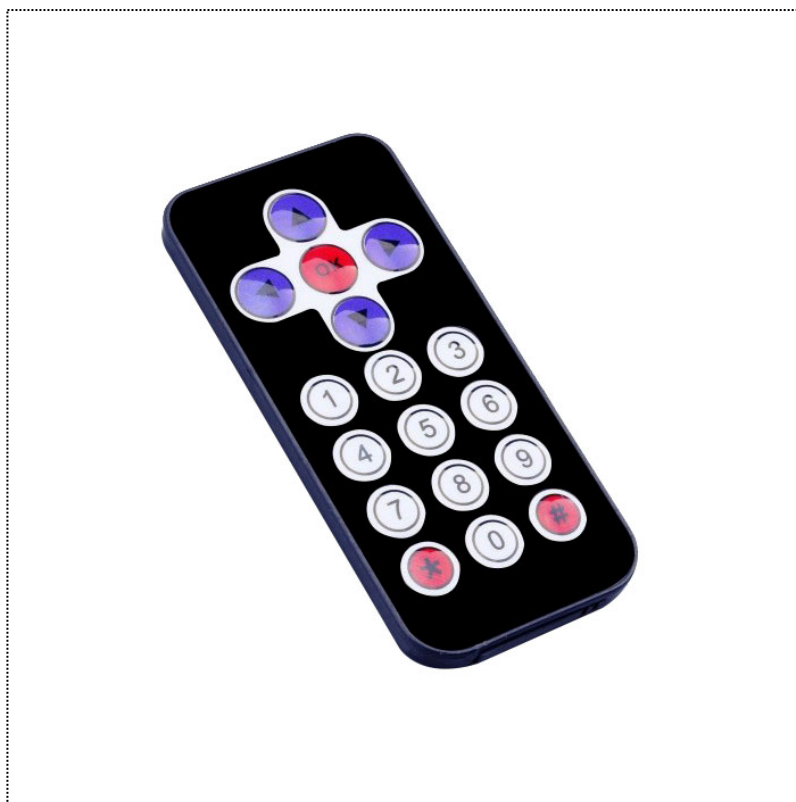


Connect The IR Module to V5 Board.



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

これが、IR リモートです：



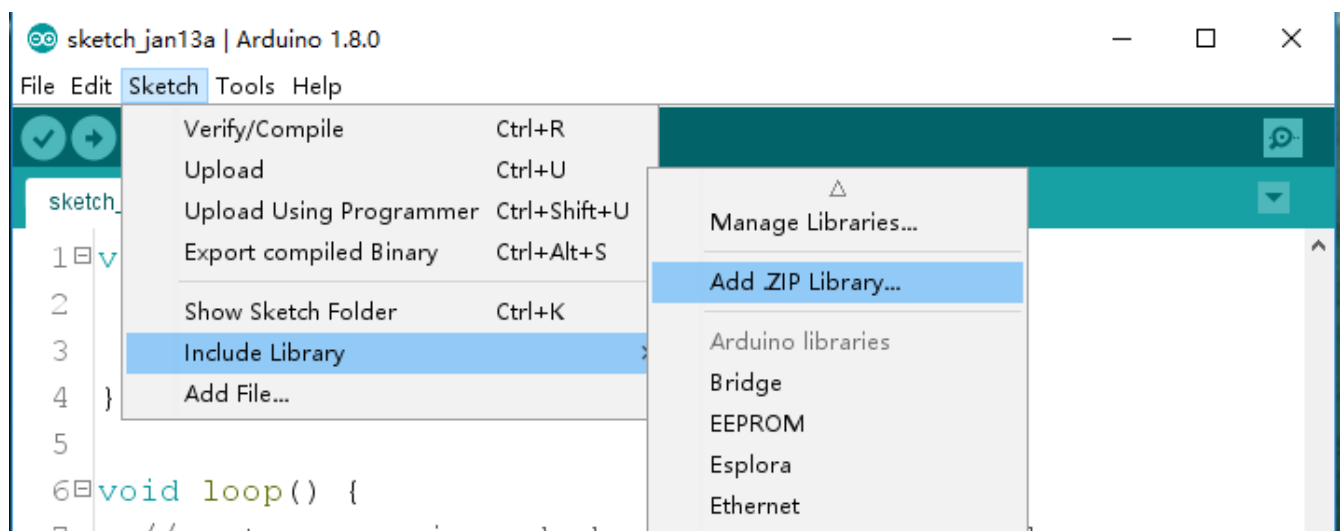
Ⅱ. テストプログラム

このプログラムでは、ライブラリを使用する必要があるため、最初にライブラリファイルを追加する必要があります。

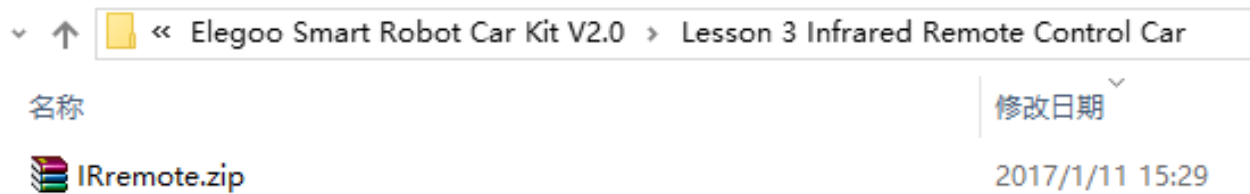
Arduino IDE を開く



Click Sketch—Include Library—Add .ZIP Library...—then select the library as blow.

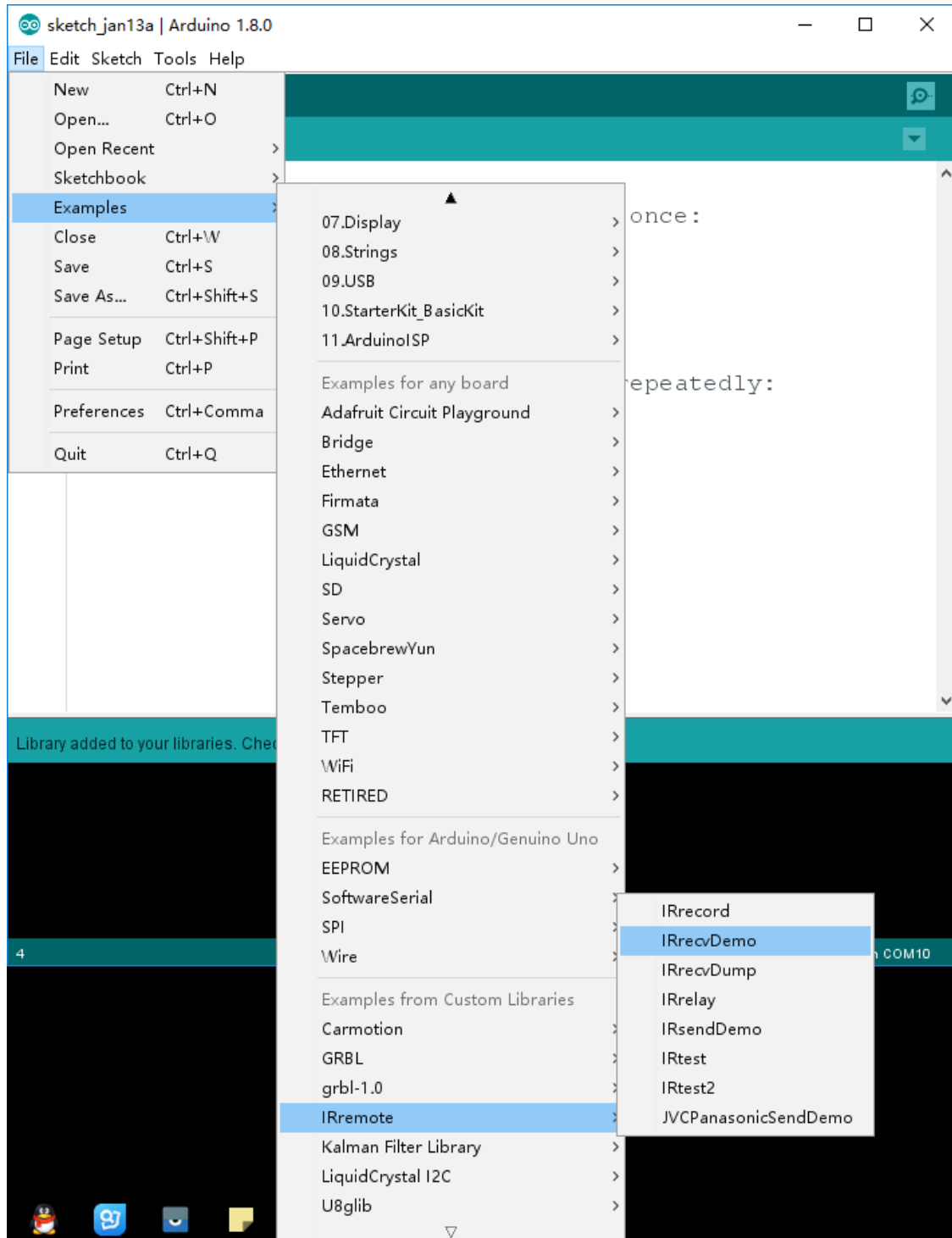


ZIP ライブラリのファイル名は IRremote.zip でなければなりません

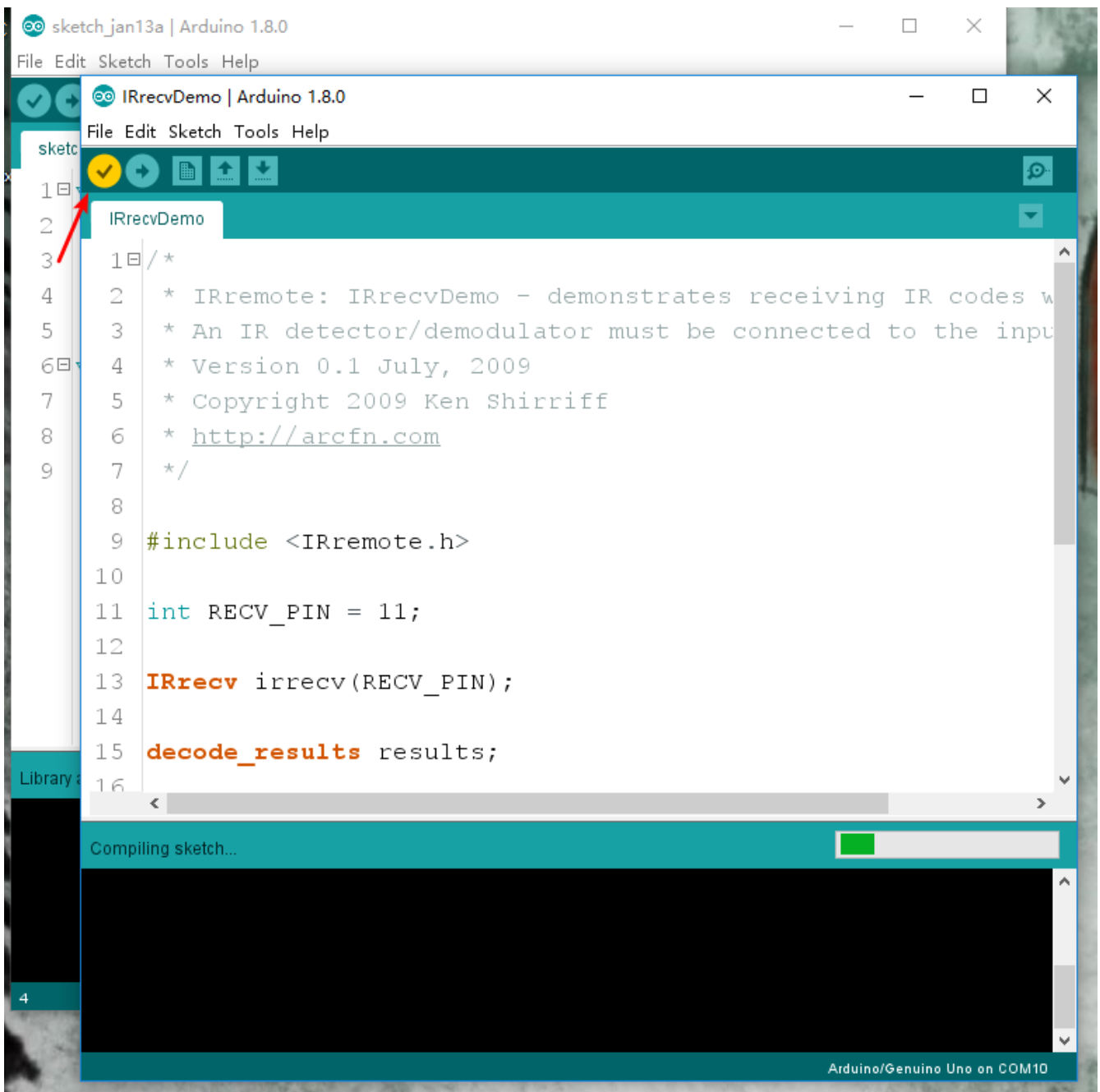


このライブラリファイルは、特別に変更されたライブラリファイルでコンパイルする必要があります。

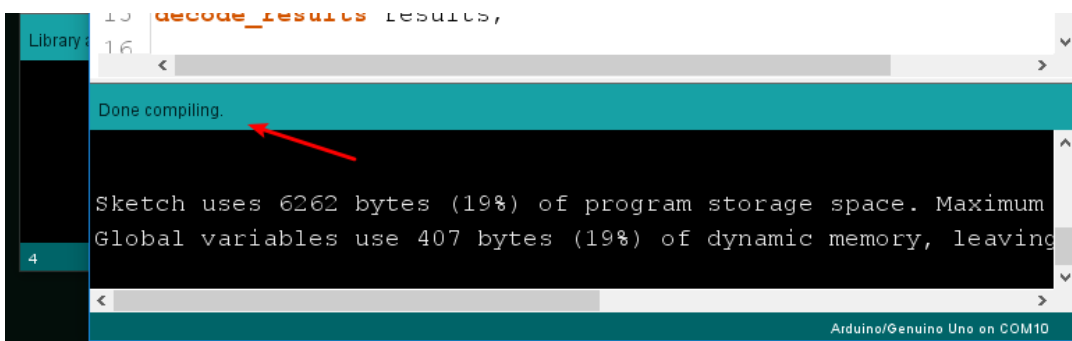
IRremote の例を選択する




[コンパイル]ボタンをクリックします



コンパイルを完了します。 そうでない場合は、IRremote ライブラリが正常にインストールされていないことを示します。 IRremote ライブラリを再度追加してください。



ファイルを開きます。infrared_Blink¥infrared_Blink. ino

名称	修改日期	类型
 infrared_Blink.ino	2017/1/5 11:57	Ardui

コードは以下の通りです:

```
#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);

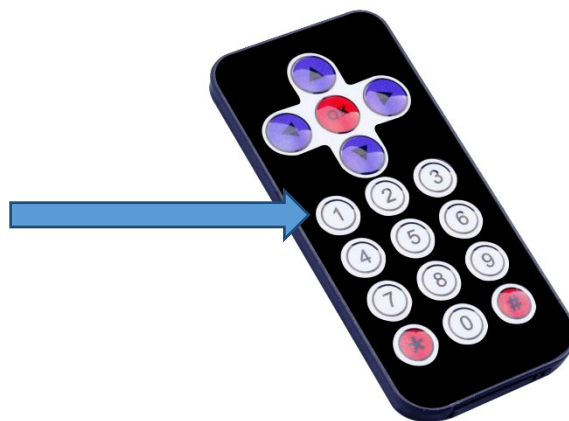
    if(RED==L)
```



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

ROMEO コントローラボードにプログラムをアップロードします。車をコンピュータから切断した後、電源スイッチをオンにして車を地上に置きます。

車に向かってボタン「1」を押すことで、車のLEDが消灯します。



Ⅲ. 原理の導入

1. 動作原理

ユニバーサル赤外線遠隔制御システムは、送信と受信の2つの部分から成り、送信部は赤外線リモコンで構成され、受信部は赤外線受信管で構成されています。赤外線リモコンで送信される信号は、バイナリパルスコードのシリアルです。無線輸送中に他の赤外線信号が散逸しないようにするには、特定の搬送波周波数で変調し、赤外線放射フォトランジスタを通じて発射するのが一般的です。赤外線受信管は、他のノイズ波をフィルタリングし、所与の周波数の信号のみを受信し、それらを復調しバイナリパルスコードに復元する。内蔵された受信管は、赤外発光ダイオードから送られた光信号を弱い電気信号に変換し、IC内部のアンプで増幅し、自動利得制御、バンドパスフィルタリング、復調、波形整形することでリモートコントローラから送られた信号を復元する。その後、赤外線受信モジュールの信号出力ピンを介してデバイスに入力されるコードによって回路に認識させる。

2. 赤外線リモコンのプロトコル

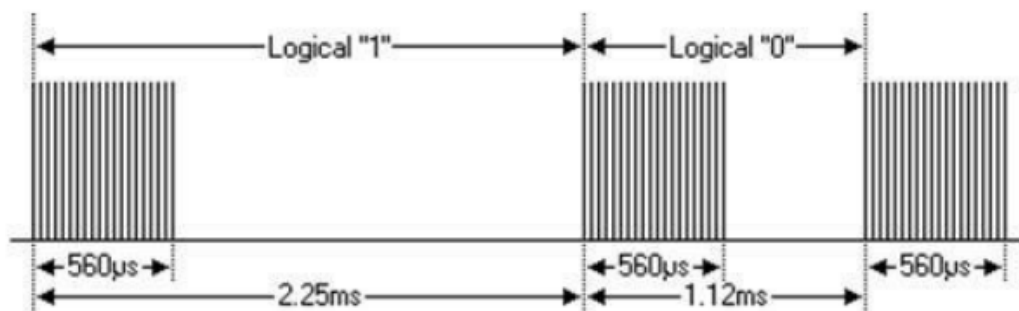
一致する IR リモート制御のコード体系は、NEC プロトコルです。

次に、NEC プロトコルが何であることを学びましょう。

特徴：

- (1) 8 アドレスビット、8 ビット
- (2) 信頼性を保証するため、アドレスビットとオーダービットを2回送信する
- (3) パルス位置変調
- (4) キャリア周波数は 38kHz
- (5) 各ビットの時間は 1.125ms または 2.25ms である

論理 0 と論理 1 の定義は次のとおりです：



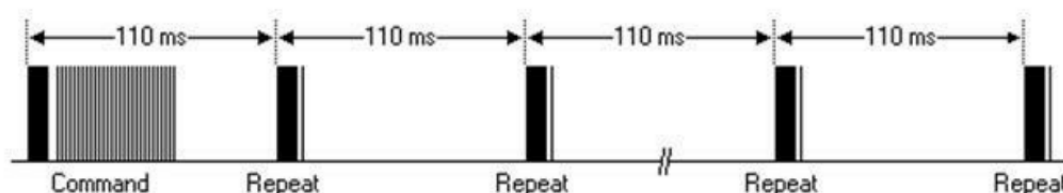
プロトコルは以下の通りです:

ボタンを押した瞬間に送信パルスを開放します:



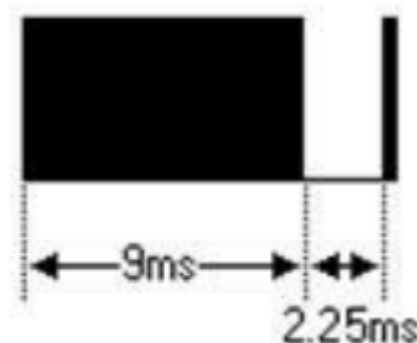
注: これはLSB (最下位ビット) を最初に送信するプロトコルです。上記パルスの転送アドレスは 0x59、次は 0x16 です。1つのメッセージは 9ms の高レベルから始まり、その後は 4.5ms の低レベル (2 レベルのガイダンスコード) とアドレスコードと命令コードによって開始されます。アドレスと命令は 2 回送信されます。2 回目は、すべてのビットが反転され、使用する受信メッセージを確認するために使用できます。合計の送信時間は固定ですが、信頼性の必要がない場合、反転の送信を無視することができ、16 ビットでアドレスと命令を展開することができます。なぜなら、すべてのビットが反対であれば、同じ長さの命令が繰り返されるからです。

押した瞬間の波形は、時間がたつと開放されます。



リモコンのボタンが押されていても、コマンドが送信された。ボタンを押したままにすると、最初の 110ms のパルスは上と異なり、重複したコードは 110ms ごとに送信されます。重複コードは、9ms の高レベルパルスと 2.25 の低レベルと 560 μ s の高レベルで構成されています。

繰り返しパルス:



注: インパルス波形がセンサーに入った後、センサーは信号を増幅し柔軟に復調しなければならないため、

赤外線信号がない時、信号があり出力端子が High 又は Low である時を考慮しなければならない。したがって、出力信号のレベルは送信端子とは反対である。オシロスコープを通して受信パルスを見ることでプログラムを理解できます。

3. リモコンカーのプログラミングのアイデア

NEC コードの特性と受信側の波に基づいて、受信側の波をリーディングコード（パルス 9ms と 4.5ms）、アドレスコード（8 ビットのアドレスコードと 8 ビットアドレスフェッチ）、

16 ビット・アドレス・コード（8 ビット・アドレス・コードと 8 ビット・アドレス・フェッチを含む）、16 ビット・オーダ・コード（8 ビット・オーダ・コードと 8 ビット・オーダ・フェッチを含む）、リピート・コード（9ms のパルス、2.25ms、560us）。

論理 “01”、論理 “1”、リーディングパルス、リピートパルス、テストされた時間に応じて区別されて受信された波のハイレベルとローレベルをテストするタイマを利用する。各キーのオーダーコードに異なる動作を割り当てることで、先行コードとアドレスコードが正しく保管されているかどうか判断されます。


車の実験中には、前後に進んで左右に回して停止するように車を制御する必要があります。つまり、5つのキーが必要です。その値は次のとおりです。:

リモコン文字	キーの値
中赤ボタン	16712445
上三角	16736925
下三角	16754775
左三角	16720605
右三角	16761405

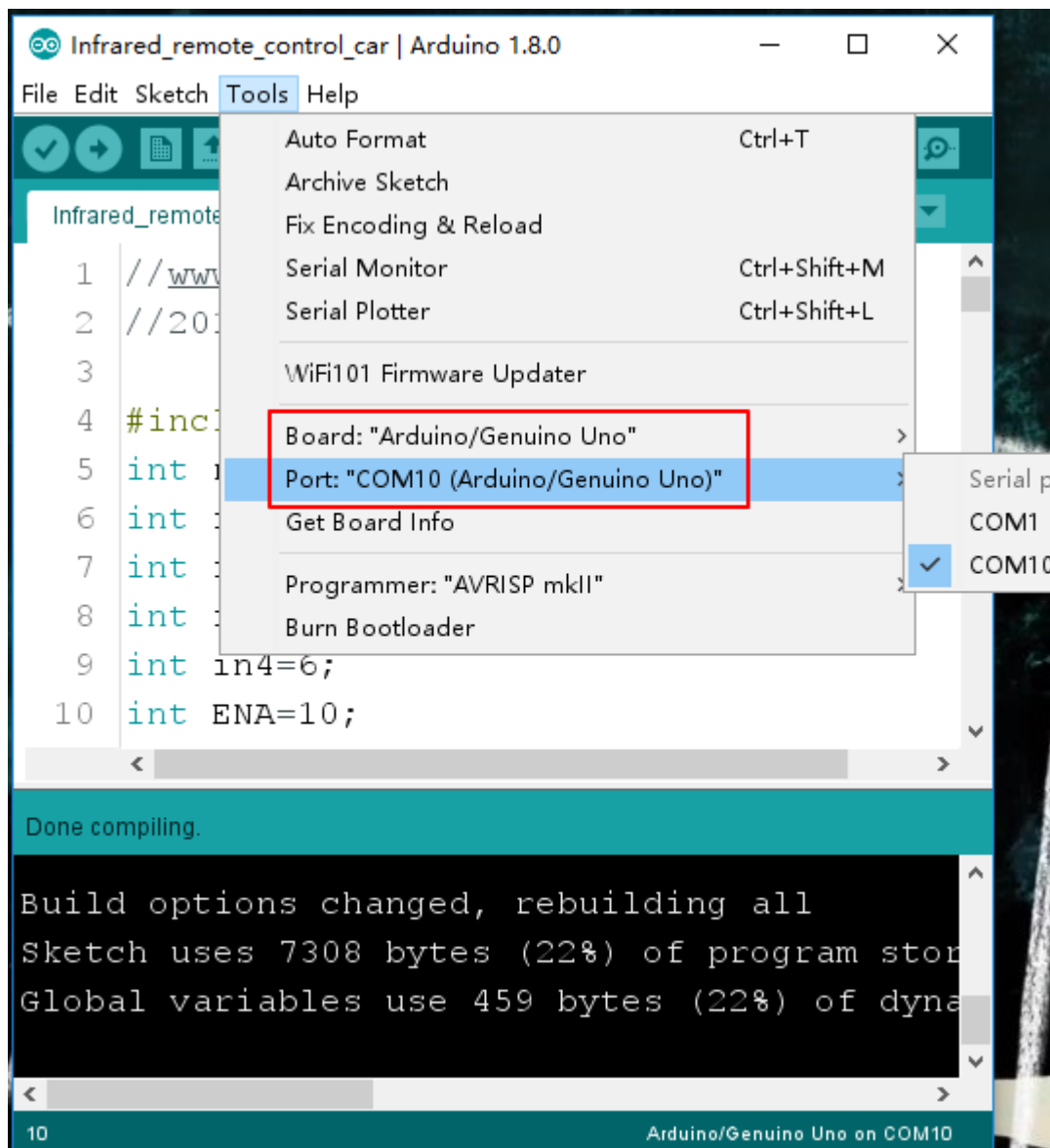


IV. 遠隔制御車を作る

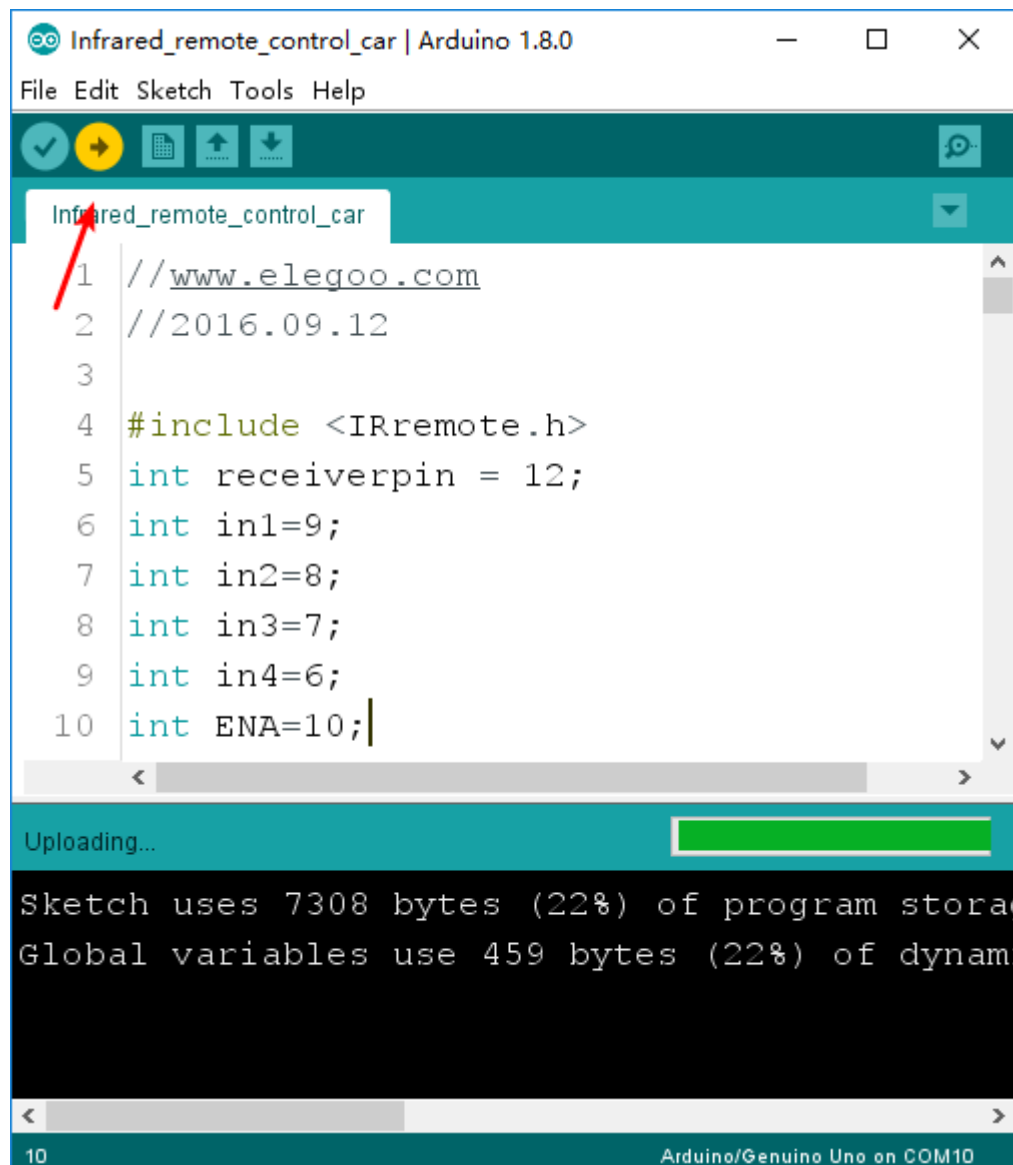
以下のようにプログラムを車にアップロードし、Infrared_remote_control_car ¥ Infrared_remote_control_car.ino ファイルを開きます。

<< Lesson 3 Infrared Remote Control Car > Infrared_remote_control_car			
名称	修改日期	类型	
 Infrared_remote_control_car.ino	2016/9/12 21:03	Arduino file	

.Arduino Uno Board とシリアルポートを選択します。



アップロードボタンを押す



アップロードが完了したら、車をコンピュータに切断します。その後、電源スイッチをオンにして、車を地上に置きます。リモコンのボタンを押すと、指示通りに車が移動するのがわかります。



これで、IR 制御車で楽しく遊ぶことができます。

コードは以下の通りです：

```
#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);
        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==X)
{
    _mStop();
}

}
```