

Lesson 3 Infrarotgesteuertes Auto

Die Punkte des Abschnitts

Infrarot- Fernbedienung ist eine weit verbreitete Methode für die Fernbedienungen. Das Auto wurde mit Infrarot- Empfänger ausgestattet und ermöglicht so die Steuerung über die Infrarot- Fernbedienung.

Lernteile:



Verstehen Sie die Infrarot- Fernbedienung und den Empfänger



Verstehen Sie die Fernsteuerungsprinzipien

Vorbereitungen:



Ein Auto (mit Batterie)



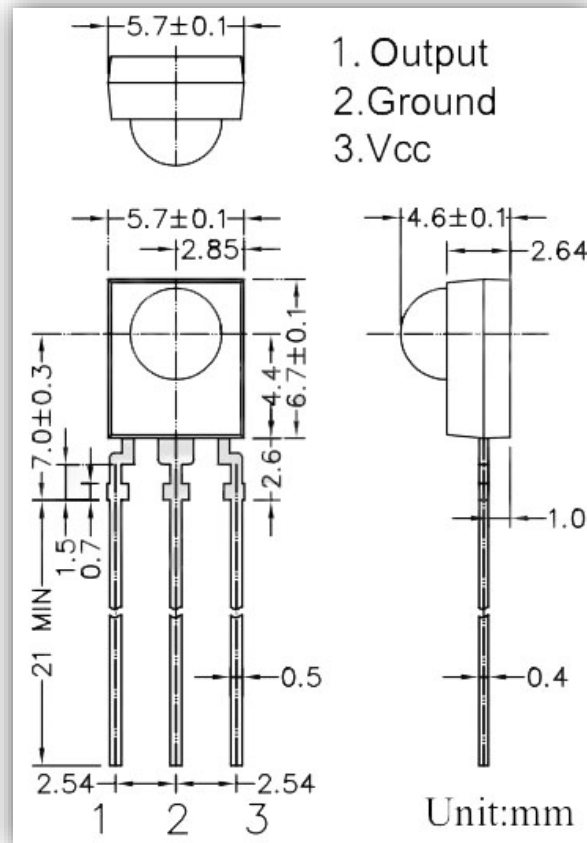
Ein USB Kabel



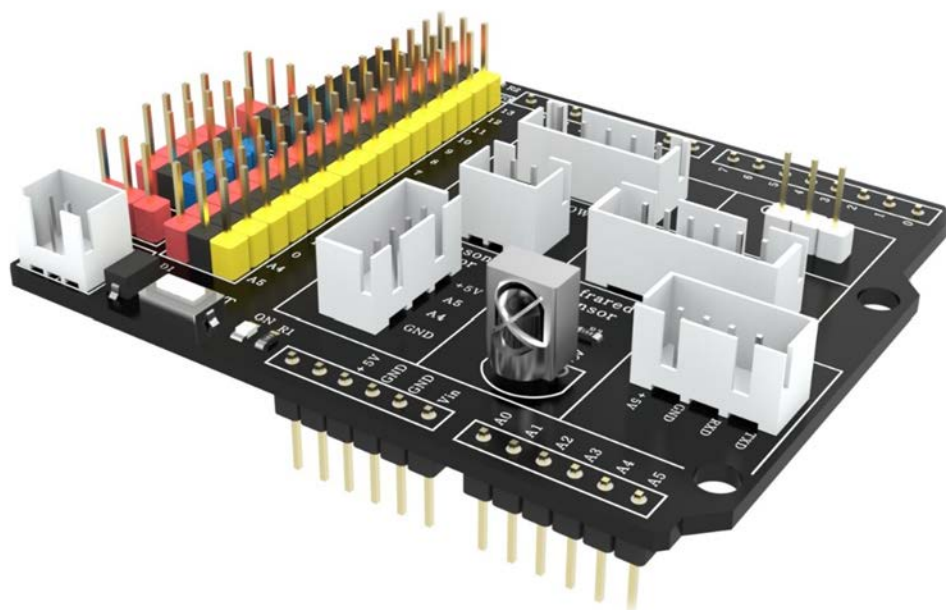
IR receiving module und IR Fernbedienung

I . R Empfangsmodul und IR- Fernbedienung

Die Daten des IR -Empfängersensors sind wie folgt:



Die Verbindung des Empfängermoduls ist wie folgt:



Das ist die IR- Fernbedienung:



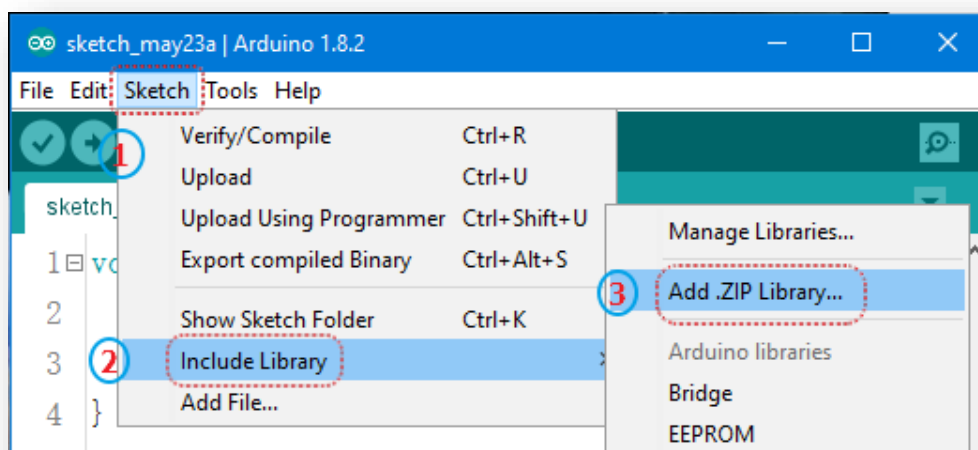
II. Test- Programm

Da wir in diesem Programm eine zusätzliche Bibliothek benötigen, laden wir diese zu allererst in unser Dateisystem.

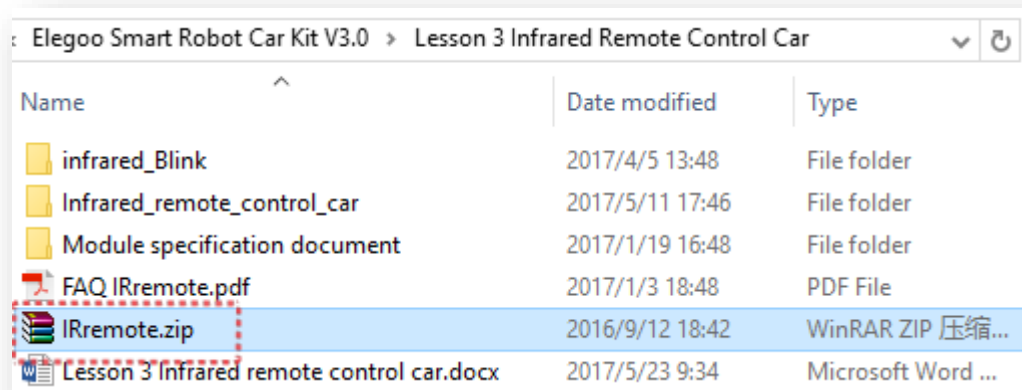
Öffnen Sie die Arduino IDE



Klicken Sie auf Sketch- Include Library- Add .ZIP- Bibliothek ... - wählen Sie die Bibliothek wie unten aus.

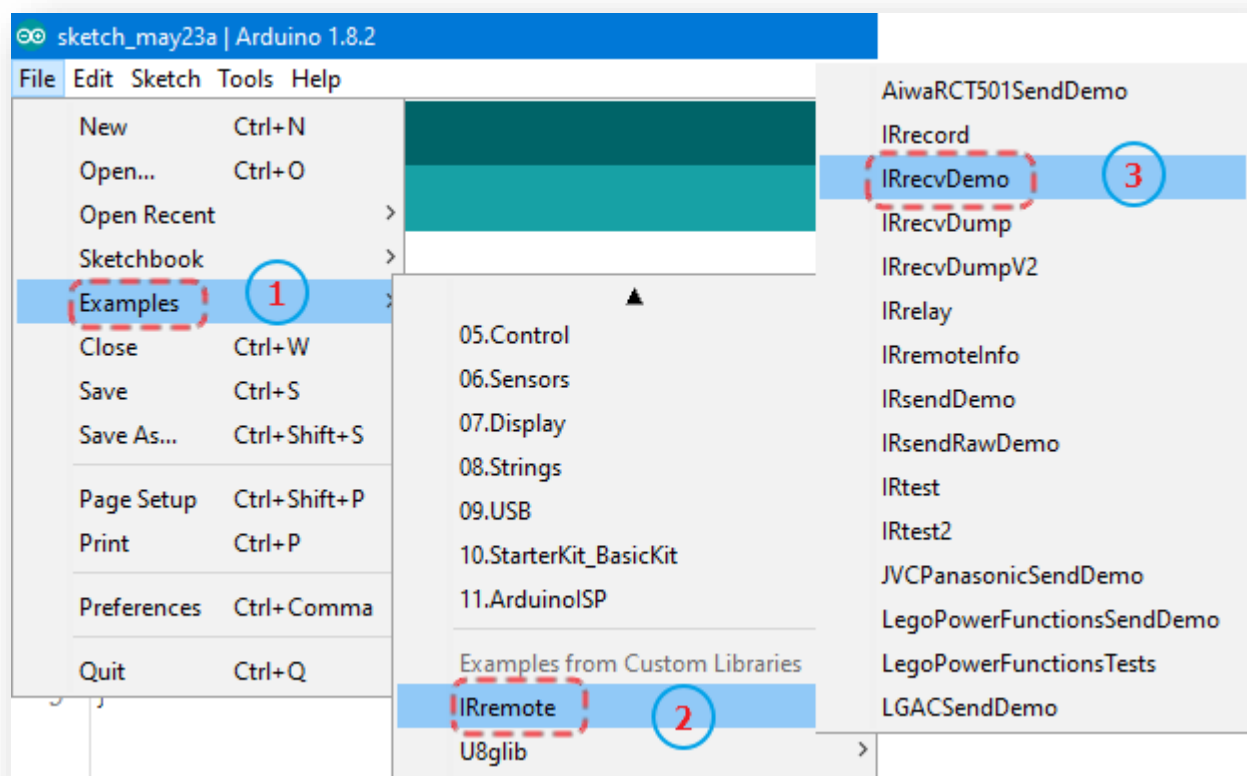


Der Dateiname der ZIP- Bibliothek muss IRremote.zip sein.

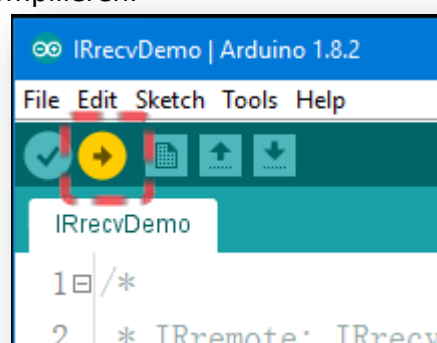


Das Programm muss mit dieser Bibliotheksdatei kompiliert werden, da es eine speziell modifizierte Bibliotheksdatei ist.

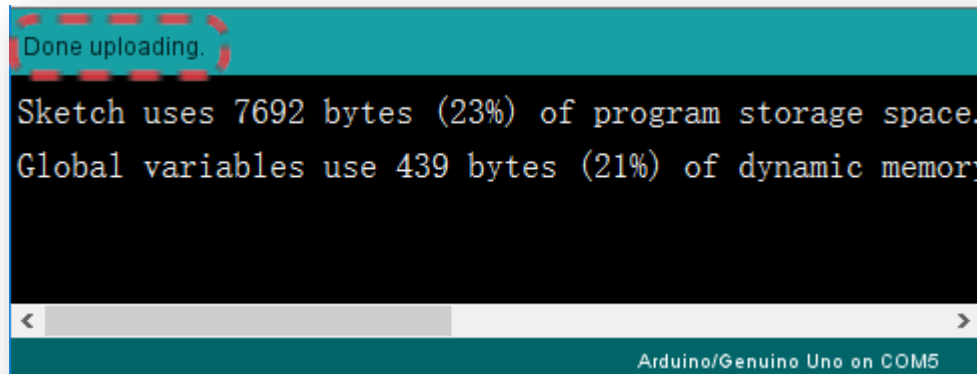
Wähle ein IRremote- Beispiel



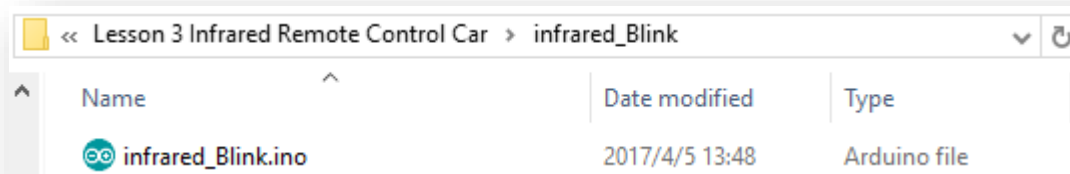
Klicken Sie auf die Schaltfläche Kompilieren.



Fertig kompiliert. Wenn nicht, wurde die IRremote- Bibliothek nicht erfolgreich installiert. Bitte füge die IRremote- Bibliothek noch einmal hinzu.



Öffnen Sie die Datei infrared_Blink \ infrared_Blink.ino



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;

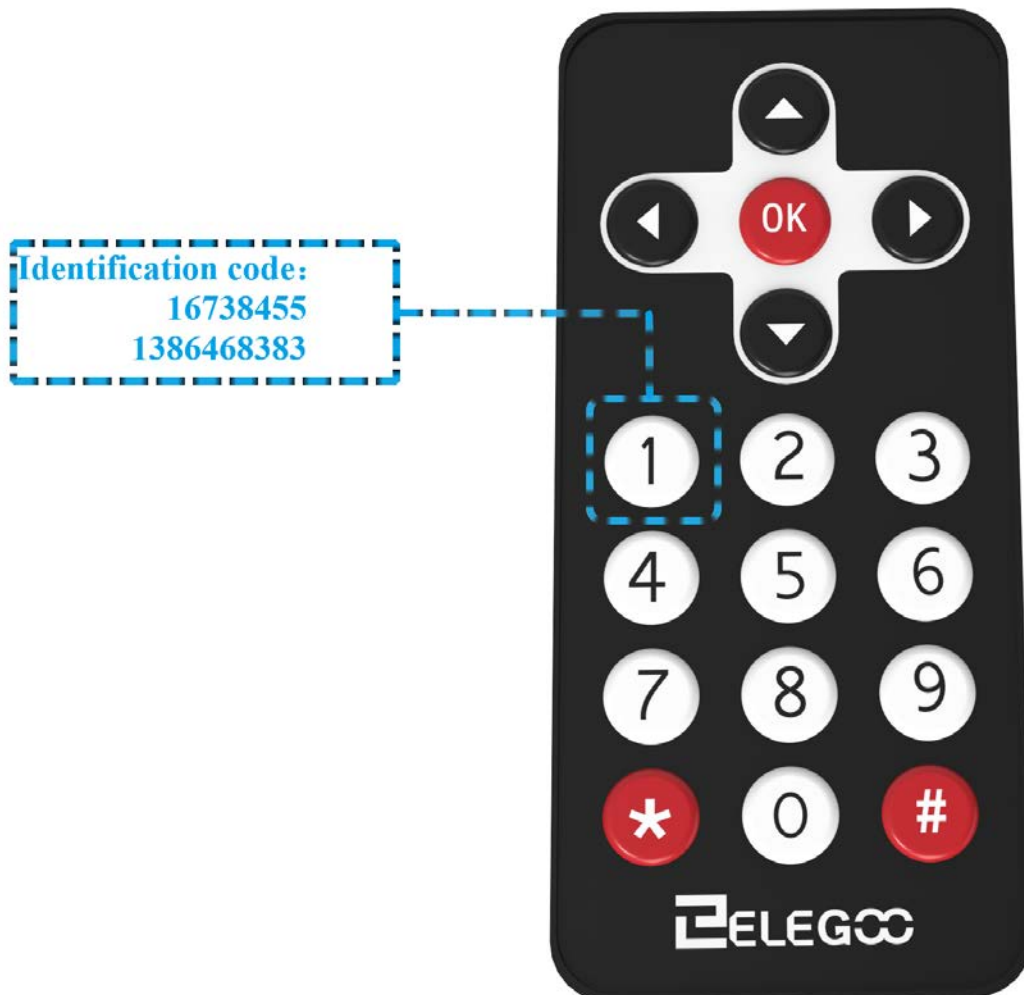
IRrecv irrecv(RECV_PIN); //initialization
decode_results results;  //Define structure type

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();  
    }  
  }  
}
```

Nach dem Trennen des Autos vom Computer können Sie den Netzschalter einschalten und das Fahrzeug auf den Boden stellen.

Drücken Sie die Taste "1" in Richtung des Autos, beobachten Sie das Auto, und Sie sehen, dass die LED mit einem Label "L" auf der Erweiterungskarte, sich ausschaltet.



III. Einführung des Grundprinzips

1. Arbeitsprinzip

Das Universal- Infrarot- Fernsteuerungssystem besteht aus zwei Teilen: Senden und Empfangen, das sendende Teil besteht aus einer IR-Fernbedienung, das empfangende Teil besteht aus einem Infrarot-Empfangsrohr. Die Signale, die durch IR- Fernsteuerung gesendet werden, sind ein serieller binärer Puls- Code. Um von der Ablenkung anderer Infrarotsignale während des drahtlosen Transports frei zu sein, ist es allgemein üblich, sie bei der gegebenen Trägerfrequenz zu modulieren und sie dann durch einen infrarot ausgestrahlten Phototransistor zu senden. Infrarot- Empfangsschlauch filtert andere Rauschwellen aus, empfängt nur Signale mit gegebener Frequenz und stellt dann den Binärpuls-code mit Hilfe der Demodulation wieder her. Der Eingebaute Empfangsschlauch empfängt und verwandelt Lichtsignale, die von Infrarot-Leuchtdioden mit schwachen elektrischen Signalen gesendet werden. Die Signale werden durch Verstärker im IC verstärkt und durch automatische Verstärkungssteuerung, Bandpassfilterung, Demodulation, Wellenformung und so weiter wiederhergestellt. Die Codierung, die per Fernbedienung gesendet wird, erkennt die Schaltung durch die Codierung, die in das elektrische Gerät über den Signalausgangspin des Infrarot- Empfangsmoduls eingegeben wird.

2. Protokoll der Infrarotfernsteuerung

Das Codierungsschema der angepassten IR-Fernsteuerung ist: NEC-Protokoll.

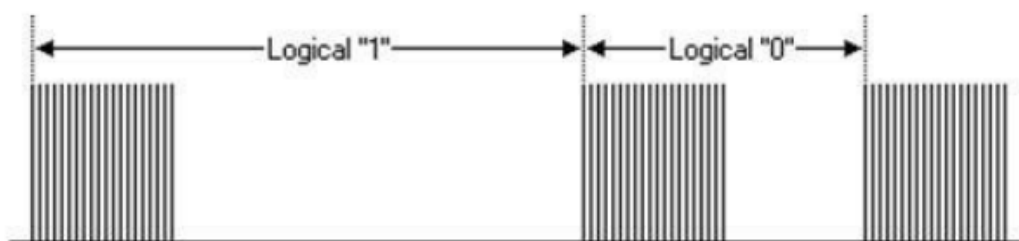
Als nächstes wollen wir lernen, was NEC-Protokoll ist.

Eigenschaften:

- (1) 8 Adreßbit, 8 Ordnung Bit
- (2) Adressbit und Bestellbit werden zweimal gesendet, um die Zuverlässigkeit zu gewährleisten
- (3) Pulspositionsmodulation
- (4) Trägerfrequenz ist 38kHz
- (5) Die Zeit eines jeden Bits beträgt 1.125ms oder 2.25ms

Definitionen von logischen 0 und 1 sind wie folgt:

Definitions of logical 0 and 1 are as below:



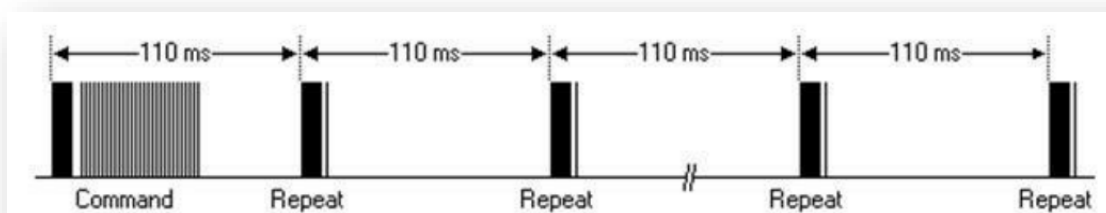
Das Protokoll ist wie folgt:

Drücken Sie sofort den Übertragungsimpuls:



Hinweis: Dies ist das Protokoll des Sendens von LSB ((least-significant bit)am wenigsten signifikantes Bit). Transportadresse des obigen Impulses ist 0x59, Auftrag ist 0x16. Eine Nachricht beginnt von einem hohen Niveau von 9ms, die Folge ist ein niedriger Grad von 4,5ms, (zwei Ebenen Formular Leitfaden) und durch Adresscode und Bestellcode. Adresse und Bestellung werden zweimal übermittelt. Beim zweiten Mal können alle Bits umgekehrt umgekehrt werden, um die zu verwendenden Empfangsmeldungen zu bestätigen. Die Gesamtsendungszeit ist fest, wenn Sie nicht daran interessiert sind, können Sie die Zuverlässigkeit der Invertierung ignorieren und können Adresse und Ordnung um 16 bit erweitern! Weil die Tatsache, dass Länge wiederholen, wenn jedes Bit ist entgegengesetzt.

Drücken Sie den gesendeten Puls nach einiger Zeit.



Ein Befehl wird gesendet, auch wenn die Taste der Fernbedienung immernoch gedrückt wird. Wenn die Taste noch gedrückt wird, ist der Puls der ersten 110ms von oben verschieden, der doppelte Code wird nach allen 110ms übertragen. Der doppelte Code besteht aus einem Hochpegelimpuls von 9ms und einem niedrigen Niveau von 2,25 und einem hohen Niveau von 560µs.

Wiederholen Sie den Puls:



Anmerkung: Nachdem die Impulswellenform in die Integration des Sensors eintritt, aufgrund der Tatsache, dass die Integration des Sensors dekodiert, signalvergrößert und plastisch sein sollte, sollten Sie die Zeit beachten, in der keine Infrarotsignale vorhanden sind, deren Ausgangsklemme auf hohem Niveau ist, wenn es Signale gibt. So ist das Ausgangssignal dem Sendeterminal entgegengesetzt. Jeder kann sehen, Empfänger Puls durch Oszilloskop, verstehen Programm mit Wellenform gesehen.

3. Die Idee der Programmierung eines solchen Ferngesteuerten Autos

Entsprechend der Charakteristik des NEC-Codes und der Welle des Empfangs-Endes teilt dieses Experiment die Welle des Empfangs-Endes in vier Teile: Leitcode (Puls von 9ms und 4,5ms), Adresscode (einschließlich 8-Bit-Adresscode und 8-Bit) Adresse abrufen), 16-Bit-Adresscode (inkl. 8-Bit-Adresscode und 8-Bit-Adressabruf), 16-Bit-Bestellcode (inkl. 8-Bit-Bestellcode und 8-Bit-Bestellabruf), Wiederholungscode (bestehend aus Puls von 9ms) , 2,25 ms, 560us).

Nutzen Sie den Timer, um den hohen Pegel und den niedrigen Wellenpegel zu testen, der nach der getesteten Zeit unterschieden wird: logisch "01", logisch "1", führender Puls, Wiederholimpuls. Führender Code und Adresscode werden beurteilt ob sie korrekt sind, werden nicht gespeichert, aufgrund der Tatsache, dass der Bestellcode jeder Taste unterschiedlich ist, wird die Aktion durch den Bestellcode ausgeführt.

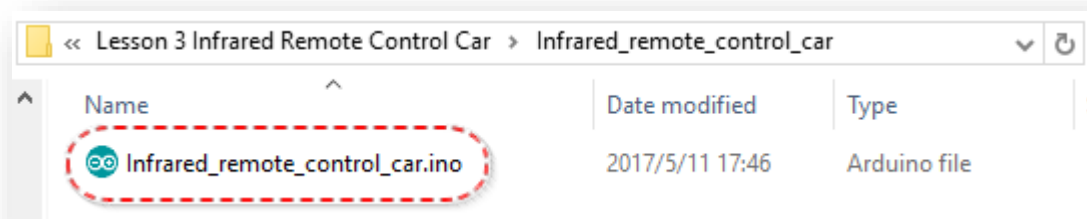
Während des Auto-Experiments müssen wir nur das Auto steuern, um vorwärts und rückwärts zu fahren, nach links und rechts zu drehen und zu stoppen, was bedeutet, dass wir 5 Schlüssel benötigen und der Wert von ihnen ist wie unten zu sehen:

Fernsteuerungszeichen	Schlüssel- Variable
Mittlerer roter Knopf	16712445
Oberes Dreieck	16736925
Unteres Dreieck	16754775
Linkes Dreieck	16720605
Rechtes Dreieck	16761405

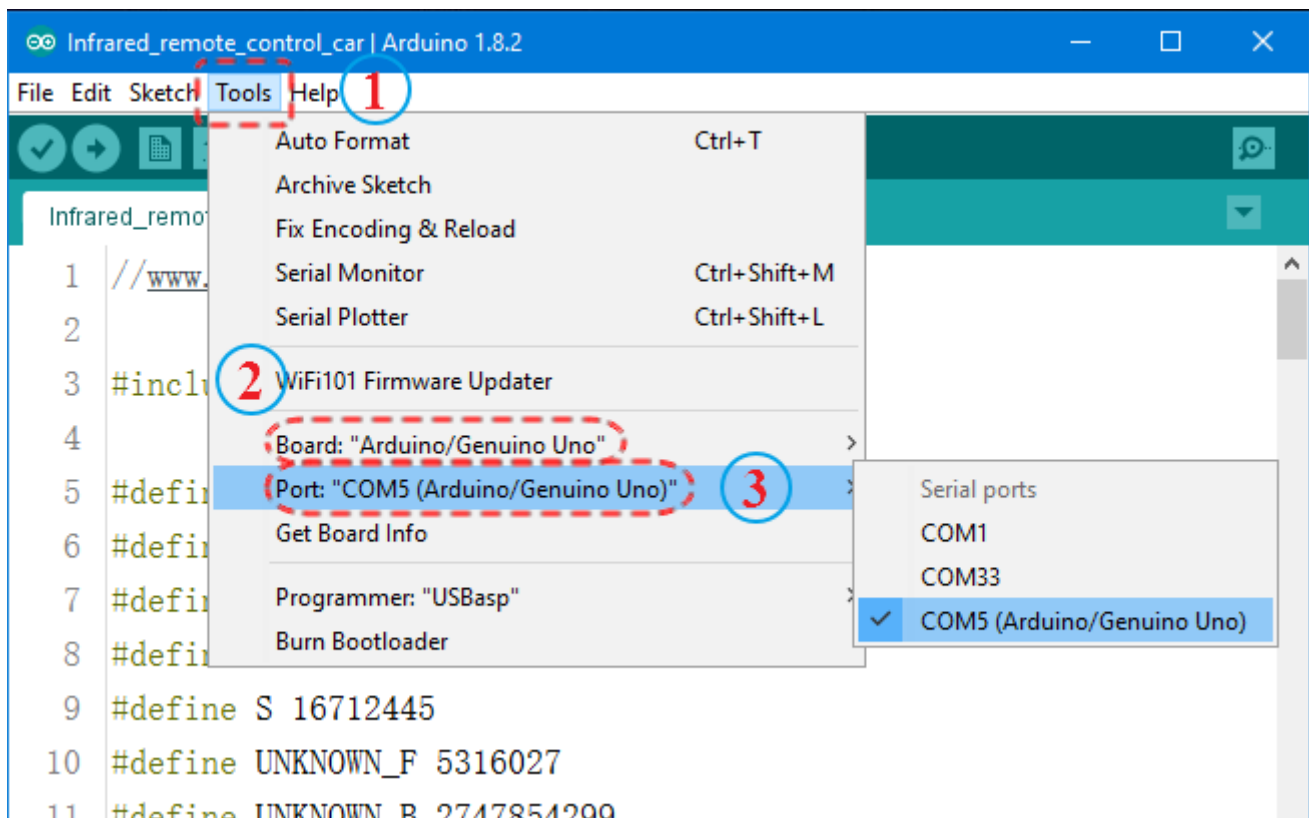


IV. Bau ein Ferngesteuertes Auto

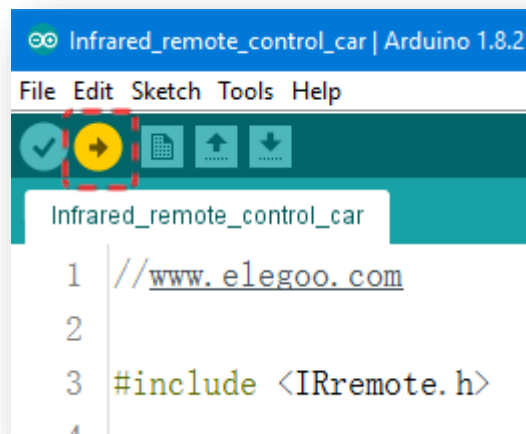
Wir laden das Programm wie unten auf das Auto, öffnen Sie die Datei Infrared_remote_control_car \ Infrared_remote_control_car.ino



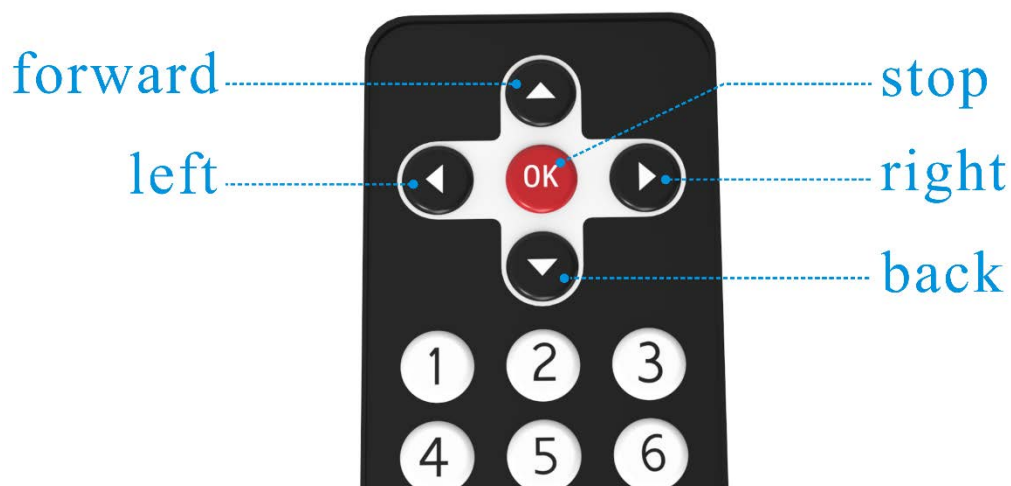
Wählen Sie die Arduino Uno Board und Serial Port.



Drücken Sie die Upload-Taste



Nach dem Hochladen, trennen Sie das Auto von den Computer. Dann schalten Sie den Netzschalter ein und stellen den Wagen auf den Boden. Drücken Sie die Taste auf der Fernbedienung und Sie können sehen, das Auto bewegen sich entsprechend, wie Sie befehlen.



Voila, jetzt kannst du mit dem IR-Steuerwagen glücklich spielen.

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();
  }
}
}
```