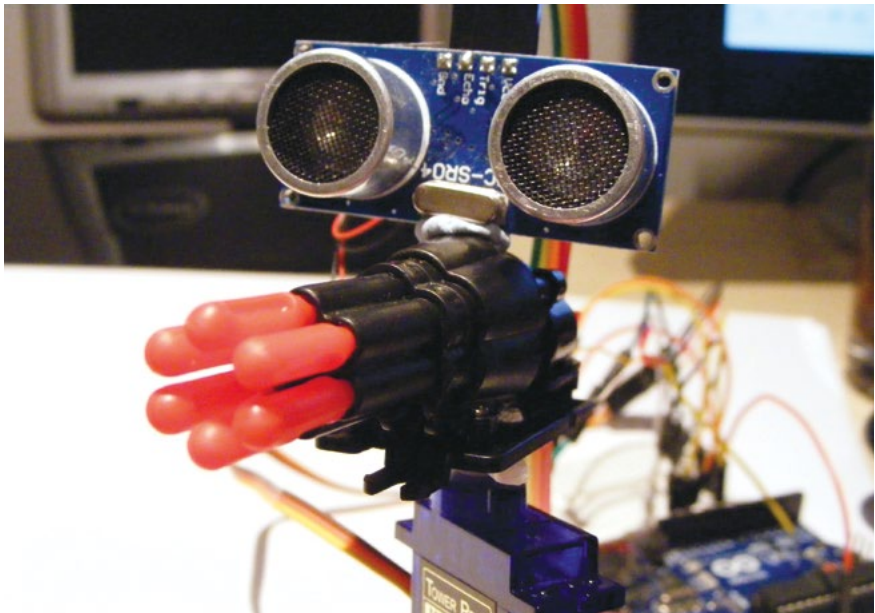
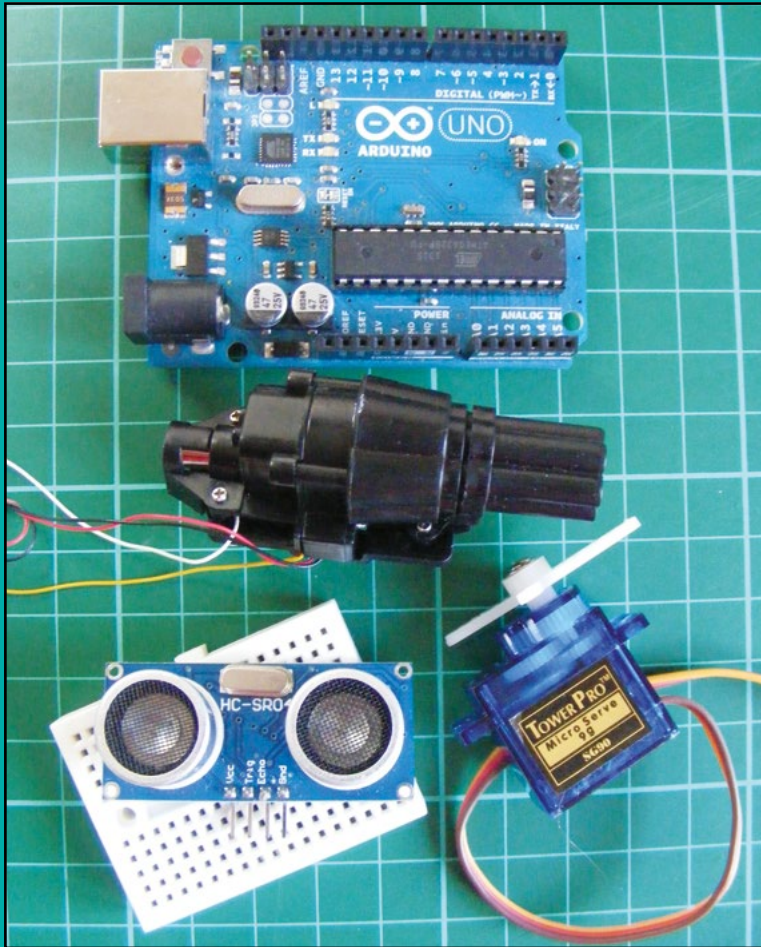


Projekt 20

Eine Roboter-Kanone

Eine Roboter-Kanone ist eine unbemannte Waffe, die Feinde mit einem Ultraschalldetektor automatisch erkennt und bekämpft. In diesem Projekt bauen wir eine Miniaturversion dieser Waffe.





Benötigte Bauteile

- Arduino-Platine
- Mini-Steckbrett
- Jumperkabel
- Jumperkabel (männlich/männlich)
- Ultraschallsensor HC-SR04 mit vier Pins
- Raketenwerfer RC V959 von WLToys
- Servomotor Tower Pro SG90 9g

Benötigte Bibliotheken

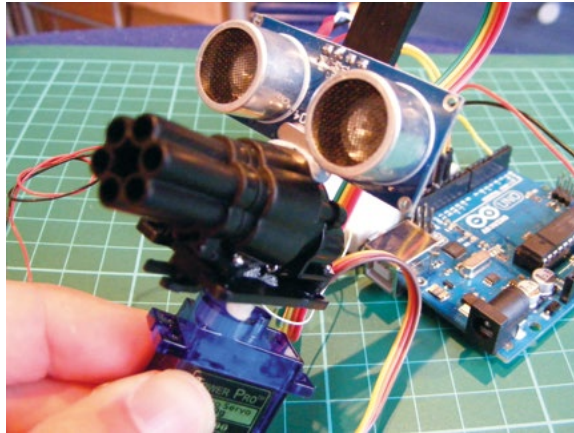
- Servo
- NewPing

So funktioniert es

Wir verbinden den kleinen Raketenwerfer und den Ultraschallsensor mit dem Servo-Arm (siehe Abbildung 20-1), sodass der Servomotor Kanone und Sensor um 180 Grad drehen kann, was dem Ultraschallsensor einen großen Erfassungsbereich ermöglicht. Wird ein Feind erkannt, löst der Arduino die Kanone aus und schießt die Raketen ab. Nähere Informationen zum Ultraschallsensor finden Sie in Projekt 18.

Abbildung 20-1:

Die Spielzeugkanone und der Ultraschallsensor werden am Servo-Arm befestigt und haben einen weiten Erfassungs- und Drehbereich.



Die wichtigste Komponente dieses Projekts ist der Raketenwerfer WLToys RC V959, auch als Walkera-RC-V959-19-Abschussgerät für ferngesteuerte Hubschrauber bekannt (Abbildung 20-2).

Abbildung 20-2:

Die Kanone Walkera RC V959-19 im Detail



Dieses tolle Bauteil ist recht günstig (zwischen 5 und 10 €) und online gut zu bekommen. Im Abschussgerät befindet sich ein Mini-Servo, der sich dreht, um die Raketen abzuschießen. Die Anschlussleitungen dieses Servos sind weiß (Masse) und gelb (+5V). Es gibt auch schwarze und rote Anschlussleitungen für Einzelschüsse, wir verwenden jedoch nur die gelben und weißen Kabel für das Dauerfeuer.

Der Aufbau

1. Zuerst bereiten wir die Kanone vor. Entfernen Sie die vier Kabel vorsichtig aus ihrem kleinen Plastiksockel. Sie sollten sich leicht lösen lassen. Sie können auch ein Jumperkabel mit Stecker verwenden, um den Plastikclip herunterzudrücken.
2. Der Kern der Leitung ist verdreht und recht dünn. Isolieren Sie die Kabel also ein Stück ab und löten Sie Leitungen an, die sich besser in den Arduino stecken lassen, wie in Abbildung 20-3 zu sehen. Kappen Sie die schwarzen und roten Kabel oder befestigen Sie sie so, dass sie nicht im Weg sind.

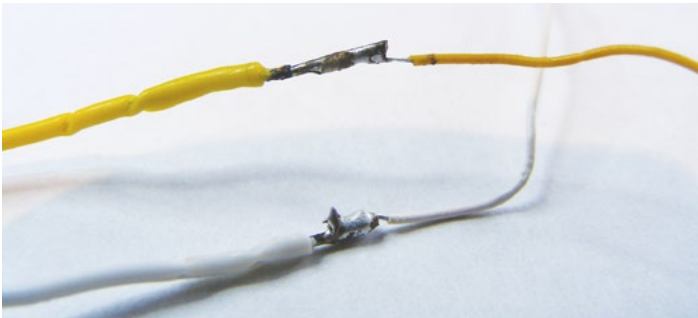


Abbildung 20-3:

Die Kabel der Kanone abisolieren und verlängern

3. Kleben Sie den Arm des Servomotors an die Basis der Kanone (siehe Abbildung 20-4).



Abbildung 20-4:

Den Arm des Servomotors festkleben

4. Befestigen Sie den Ultraschallsensor oben an der Kanone, wie in Abbildung 20-5 gezeigt. Für eine feste Verbindung verwenden Sie eine Heißklebepistole oder nehmen Sie einfaches Klebeband, wenn Sie später die Verbindung wieder lösen wollen.

Abbildung 20-5:
Den Ultraschallsensor
befestigen



5. Verbinden Sie den Ultraschallsensor über Jumperkabel mit dem Arduino: Schließen Sie Trig direkt an Pin 13 des Arduino an und Echo direkt an Pin 12. Wir verwenden ein Mini-Steckbrett, um die verschiedenen Verbindungen zum Arduino herzustellen.

Ultraschallsensor	Arduino
Versorgungsspannung	+5V
Trig	Pin 13
Echo	Pin 12
Masse	GND

6. Verbinden Sie das braune Kabel des Servomotors über das Mini-Steckbrett mit GND des Arduino, das rote Kabel mit +5V und das gelb/weiße direkt mit Pin 9 des Arduino.

Servo	Arduino
Braunes Kabel	GND
Rotes Kabel	+5V
Gelbes Kabel	Pin 9

7. Verbinden Sie das weiße Kabel der Kanone mit der Masseleiste des Mini-Steckbretts und das gelbe Kabel direkt mit Pin 3 des Arduino.

Kanone	Arduino
Weißes Kabel	GND
Gelbes Kabel	Pin 3

8. Ihre Roboter-Kanone sollte jetzt so aussehen wie in Abbildung 20-6. Laden Sie jetzt die Geschosse.

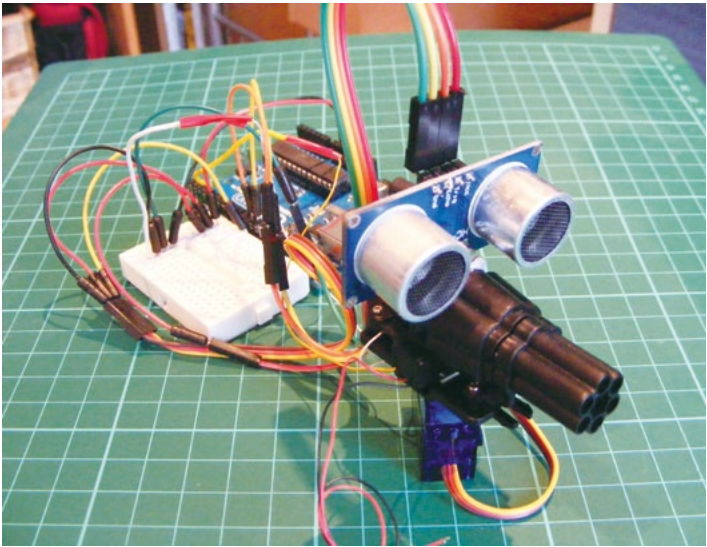
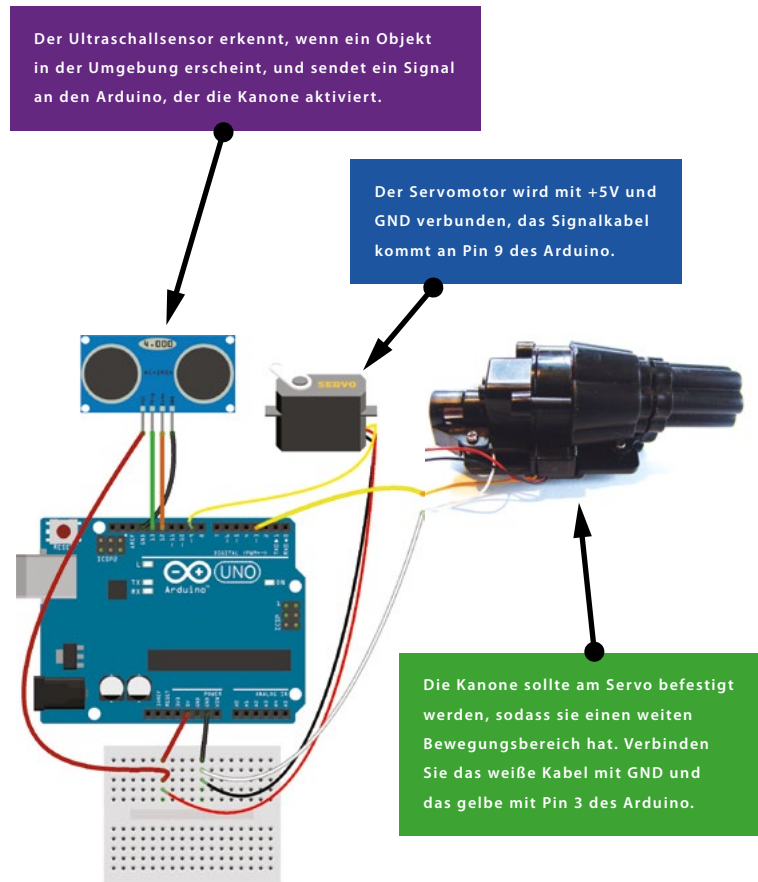


Abbildung 20-6:
Ihre Roboter-Kanone ist
bereit zum Feuern!

9. Prüfen Sie, ob Ihre Schaltung so aussieht wie in Abbildung 20-7. Laden Sie den Sketch von Seite 176 auf den Arduino.

Abbildung 20-7:
Der Schaltplan für
die Roboter-Kanone



Der Sketch

Zuerst ruft der Sketch die Bibliotheken NewPing und Servo auf, mit deren Funktionen der Servomotor und der Ultraschallsensor gesteuert werden. (Die Bibliothek NewPing müssen Sie von <http://www.dpunkt.de/arduino-projekt> herunterladen und in Ihrem Arduino-Ordner speichern.) Der Servomotor bewegt sich um 180 Grad hin und zurück. Der Sensor sendet sein Ultraschallsignal aus, und wenn es auf ein Objekt trifft, kommt ein zeitverzögertes Echo zurück. Der Arduino rechnet diesen Wert in den Abstand zwischen Sensor und Objekt um. Ist der Abstand geringer als 15 cm, hält der Servo an und die Kanone wird aktiviert, sodass sie ihre Geschosse auf das Objekt abschießt. Sie können die Abschussdistanz (die in Zentimetern angegeben wird) an **1** ändern.

```

-----
#include <NewPing.h> // Ruft die NewPing-Bibliothek auf
#include <Servo.h>    // Ruft die Servo-Bibliothek auf
#define trigPin 12   // Trig-Pin für Ultraschallsensor
#define echoPin 13   // Echo-Pin für Ultraschallsensor
#define MAX_DISTANCE 500

NewPing sonar(trigPin, echoPin, MAX_DISTANCE);

int blaster = 3; // Pin für Kanone

int angle = 0; // Legt Servoposition in Grad fest

Servo servo;

void setup() {
    Serial.begin(115200);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(blaster, OUTPUT);
    servo.attach(9); // Pin für Servo
}

void loop() {
    int duration, distance, pos = 0, i;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH); // trigPin sendet Ping
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH); // echoPin empfängt Ping
    distance = (duration / 2) / 29.1;
    Serial.print(distance);
    Serial.println(" cm");
    if (distance <= 15) { ❶ // Wenn Abstand kleiner als 15 cm
        digitalWrite(blaster, HIGH); // Kanone schießt
        servo.write(90);
    }
    else {
        digitalWrite(blaster, LOW); // Anderenfalls wird die Kanone
                                     // nicht eingeschaltet
        for (angle = 0; angle < 180; angle++) { // Schwenkt den Servo
            servo.write(angle);
            delay(15);
        }
        for (angle = 180; angle > 0; angle--) {
            servo.write(angle);
        }
        delay(450);
    }
}
}
-----

```