
BACHELORARBEIT

Herr
Yuwen Zhou

Rotierender Entfernungssensor zur Messung der Objektform

Mittweida, 2022

BACHELORARBEIT

Rotierender Entfernungssensor zur Messung der Objektform

Autor:

Herr Yuwen Zhou

Studiengang:

Elektro- und Informationstechnik

Seminargruppe:

EI18wA-BC

Erstprüfer:

Prof. Dr.-Ing. René Pleul

Zweitprüfer:

M.SC. Hongwei Xu

Einreichung:

Mittweida, 31.8.2022

Verteidigung/Bewertung:

Mittweida, 2022

BACHELOR THESIS

Measurement of object shape by rotating distance sensor

author:

Mr. Yuwen Zhou

course of studies:

Electronics and Information Technology

seminar group:

EI18wA-BC

first examiner:

Prof. Dr.-Ing. René Pleul

second examiner:

M.SC. Hongwei Xu

submission:

Mittweida, 31.8.2022

defence/ evaluation:

Mittweida, 2022

Bibliografische Beschreibung:

Zhou, Yuwen:

Rotierender Entfernungssensor zur Messung des Objektforms. - 2022. - III, III, IV S.

Mittweida, Hochschule Mittweida, Fakultät Elektro- und Informationstechnik, Bachelorarbeit, 2022

Referat:

In dieser Arbeit geht es um ein System zum Messen der Form eines Objekts, das aus einem Entfernungssensor, einem Servo, einem Tastenmodul und einem Arduino Mega 2560 besteht.

Inhalt

Inhalt	I
Abbildungsverzeichnis.....	III
Tabellenverzeichnis.....	V
1 Einleitung.....	1
2 Aufgabenstellung.....	2
3 Stand der Technik.....	3
3.1 Die optische 3D-Formmesstechnik	3
3.2 Prinzip von der Ultraschall-Entfernungsmessung	3
3.3 Prinzip von der Laser-Entfernungsmessung	4
4 Aufbau des Systems.....	5
4.1 Hardware.....	5
4.1.1 Arduino Mega 2560	5
4.1.2 Servomotor	6
4.1.3 Lasersensor.....	9
4.1.4 Ultraschallsensor	12
4.1.5 Tastenmodul	13
4.2 Software	14
4.2.1 Allgemeines Softwarekonzept	14
4.2.2 Servo-Programm	15
4.2.3 Lasersensor-Programm.....	16
4.2.4 Bildprogramm(Anzeigen von dem Form auf Computer).....	19
4.2.5Tastenmodul-Programm.....	25
5 Inbetriebnahme / Systemtest	27
5.1 Aufbau von Hard- und Software	27
5.2 Test.....	28
5.2.1 Messergebnis von dem Ultraschallsensor	28
5.2.2 Messergebnis von dem Lasersensor	29
6 Zusammenfassung	32

Literatur 33

Anlagen 34

Anlagen, Arduinoprogramm 35

Anlagen, Processingprogramm..... 37

Selbstständigkeitserklärung 39

Abbildungsverzeichnis

Abbildung 1: Arduino Mega 2560	5
Abbildung 2: Servomotor	6
Abbildung 3: Pulsweitenmodulation von Motor.....	8
Abbildung 4: Stecker Pin Belegung (URL: https://edistechlab.com/sg-90-servo-motor/?v=3a52f3c22ed6)	9
Abbildung 5: Ultraschallsensor HC-SR04	12
Abbildung 6: Tastenmodul	13
Abbildung 7: Interner Schaltplan von Tastenmodul	13
Abbildung 8: Allgemeines Softwarekonzept des System.....	错误!未定义书签。
Abbildung 9: Allgemeines Softwarekonzept	14
Abbildung 10: Servo-Programm	15
Abbildung 11: Lasersensor-Programm.....	17
Abbildung 12: Koordinatensystem in Processing	19
Abbildung 13: Zeigen von dem Scannhintergrund und der Scannlinien	23
Abbildung 14: Lasersensor des Aufbaus von Hardware.....	27
Abbildung 15: die Verarbeitungsbild-Programmiersoftware Processing	27

Abbildung 16:	Ergebnis von Ultraschallsensor	28
Abbildung 17:	Vergleich von verschiedene Strahlen	29
Abbildung 18:	Test von rechteckiges Objekt	30
Abbildung 19:	Test von gekrümmte Fläche	30
Abbildung 20:	Test von Welle	31
Abbildung 21:	Analyse von der Fehlmessung der Wellw.....	31

Tabellenverzeichnis

Tabelle 1 : Pin-Schnittstelle.....	10
Tabelle 2:Daten von VL55LXX-V2.....	11
Tabelle 3:Pin-Schnittstelle von HC-SR04.....	12
Tabelle 4:Daten von HC-SR04	12
Tabelle 5: Spezifikation von Tastenmodu	13

1 Einleitung

In den letzten Jahren haben sich Drohnen rasant entwickelt, und ein sehr wichtiger Teil von ihnen besteht darin, Objekte zu scannen, Entfernungen zu messen, in Innenräumen zu navigieren und Hindernissen durch Sensoren auszuweichen. Dies erfordert eine Bildkamera zur Entfernungsmessung, ein Lidar-System, das die Zeit erfasst, die das Licht benötigt, um von einem beliebigen umgebenden Objekt zurückzukehren, es mit Videodaten kombiniert und ein Echtzeit-3D-Bild erstellt. Es kann auch verwendet werden, um Gesichts- oder Handbewegungen zu verfolgen, Räume zu kartieren, Hintergründe aus Bildern zu entfernen und sogar 3D-Objekte in Bildern zu überlagern.

Daraus möchte ich ein Messsystem machen, das die Form durch den Entfernungssensor messen kann.

2 Aufgabenstellung

Dieses Messsystem kann eine geometrische Form innerhalb eines Halbkreises mit einem Radius von 1 Meter messen. Zunächst muss ein Entfernungssensor ausgewählt werden. Die erste Überlegung ist der Ultraschallsensor, aber wenn er nicht geeignet ist, gibt es eine zweite Option, den Lasersensor. Dann muss in diesem System der Sensor um 180 Grad gedreht werden, also wird ein Servomotor benötigt. Für den Servomotor gibt es auch ein Tastenmodul, um den Start und Stopp des Servomotors zu steuern. Das Wichtigste ist der Mikrocontroller, ich habe den Arduino Mega 2560 benutzt. Das obige ist die Hardware, ich muss auch den Mikrocontroller über die Arduino IDE-Software auf dem Computer programmieren und hochladen. Um schließlich den Form zu zeigen, werde ich die Bildbearbeitungssoftware Processing verwenden, um die Messergebnisse in Echtzeit anzuzeigen.

3 Stand der Technik

3.1 Die optische 3D-Formmesstechnik¹

Auf dem Gebiet der Formmessung von Objekten ist die bestehende Technologie die optische dreidimensionale Formmesstechnik. Optische 3D-Formmesstechniken können grob in zwei Kategorien eingeteilt werden: Scanning-Techniken und Non-Scanning-Techniken. Ein Beispiel für eine Scantechnologie ist Lidar. Bei dieser Technologie erkennt Lidar die Form eines Objekts, indem es seine Oberfläche mit einem Laserstrahl abtastet. Erstellen Sie Objektmodelle mit Licht, das von Objekten reflektiert wird. Scanverfahren wie Lidar sind im Allgemeinen zeitaufwändig, da sie ein ein- oder zweidimensionales Scannen erfordern, um die gesamte Oberfläche des Objekts abzudecken.

Nicht scannende Techniken zum Messen von 3D-Oberflächen sind im Allgemeinen schneller als scannende Techniken. Außerdem ist die Bildverarbeitung zum Abrufen von dreidimensionalen Konturinformationen relativ einfach und leicht verständlich. Ein Beispiel für eine nicht abtastende Technik ist das Streifenprojektionsverfahren. Das Interferenzstreifen-Projektionsverfahren ist ein Verfahren zur dreidimensionalen Formmessung durch Projizieren des Interferenzstreifenmusters auf das zu messende Objekt. Eine Kamera oder eine andere Bilderkennungs Vorrichtung erfasst ein Bild des modulierten Interferenzstreifenmusters, d. h. des projizierten Interferenzstreifenmusters von der Oberfläche des Objekts. Diese Bilder werden dann verarbeitet, um die 3D-Form des Objekts zu konstruieren

3.2 Prinzip von der Ultraschall-Entfernungsmessung²

Der Ultraschallsender sendet Ultraschallwellen in eine bestimmte Richtung aus und beginnt gleichzeitig mit der Emissionszeit mit der Zeitmessung. Die Ultraschallwelle breitet sich in der Luft aus und kehrt sofort zurück, wenn sie auf dem Weg auf ein Hindernis trifft. Der Ultraschallempfänger stoppt sofort die Zeitmessung wenn es die reflektierte Welle empfängt. Die Ausbreitungsgeschwindigkeit von Ultraschallwellen in der Luft beträgt 340 m/s. Anhand der vom Timer aufgezeichneten Zeit t kann die Entfernung (s) zwischen dem Startpunkt und dem Hindernis berechnet werden, nämlich: $s = 340 \cdot t/2$. Dies ist das sogenannte Zeitdifferenz-Ranging-Verfahren.

¹ <https://patents.google.com/patent/CN100520287C/zh>

² <https://baike.baidu.com/item/%E8%B6%85%E5%A3%B0%E6%B3%A2%E6%B5%8B%E8%B7%9D/8544144>

Das Prinzip der Ultraschall-Entfernungsmessung besteht darin, die Ausbreitungsgeschwindigkeit von Ultraschallwellen in der Luft zu kennen, die Zeit zu messen, zu der die Schallwellen nach der Übertragung von Hindernissen reflektiert werden, und die tatsächliche Entfernung vom Sendepunkt zum Hindernis entsprechend zu berechnen die Zeitdifferenz zwischen Senden und Empfangen. Es ist ersichtlich, dass das Prinzip der Ultraschall-Entfernungsmessung das gleiche wie das des Radars ist.

Die Formel für die Entfernung wird ausgedrückt als: $L=C \times T$

In der Formel ist L die gemessene Entfernungslänge, C die Ausbreitungsgeschwindigkeit von Ultraschallwellen in der Luft, T die Zeitdifferenz der gemessenen Entfernungsausbreitung (T ist die Hälfte des Zeitwerts von Senden bis Empfangen).

3.3 Prinzip von der Laser-Entfernungsmessung³⁴

TOF ist die Abkürzung für Time of Flight-Technologie, das heißt, der Sensor sendet modulierte Nahinfrarotlicht aus, das nach dem Auftreffen auf ein Objekt reflektiert wird. Tiefeninformationen, kombiniert mit herkömmlichen Kameraaufnahmen, können den dreidimensionalen Umriss des Objekts als ein darstellen Topografische Karte mit verschiedenen Farben, die unterschiedliche Entfernungen darstellen. Der Messbereich kann mehrere Meter erreichen, die Genauigkeit beträgt weniger als 2 cm und die Aktualisierungsfrequenz pro Sekunde kann 30 bis 60 fps erreichen.

Vorteile der TOF-Technologie:

- (1) Geringe Softwarekomplexität, einfaches Design und einfache Anwendung;
- (2) Es funktioniert gut in dunklen und stark beleuchteten Umgebungen;
- (3) Der Stromverbrauch ist nicht hoch;
- (4) Es gibt einen großen Erkennungsabstand;

³ <https://baike.baidu.com/item/TOF%E6%8A%80%E6%9C%AF/5198119?fr=aladdin>

⁴ <https://blog.csdn.net/limingmin2020/article/details/124141523>

4 Aufbau des Systems

4.1 Hardware

4.1.1 Arduino Mega 2560



Abbildung 1: Arduino Mega 2560

Arduino Mega 2560 ist ein Mikrocontroller-Board basierend auf ATmega2560, mit 54 digitalen Eingangs-/Ausgangsanschlüssen (15 davon können als PWM-Ausgang verwendet werden), 16 analogen Eingangsanschlüssen, 4 seriellen UART-Anschlüssen, 16-MHz-Quarzoszillator, USB-Anschluss, Batterieanschluss, ICSP-Header und Reset-Taste. Schließen Sie es einfach über USB an einen Computer an oder verwenden Sie einen AC/DC-Adapter.⁵

In diesem Projekt habe ich eine PWM-Ausgangsschnittstelle, eine 5-V-Stromschnittstelle, GND und eine SDA-, SCL-Schnittstelle verwendet.

(1) PWM⁶

Das Servomotor wird durch PWM gesteuert, und die Steuerleitung wird verwendet, um die Impulse zu übertragen. PWM ist Pulsweitenmodulation. Es ist eine Technik zur

⁵ <https://ost.51cto.com/posts/1543>

⁶

https://blog.csdn.net/hypecycle/article/details/110674844?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_baidulandingword~default-1-110674844-blog-124407506.pc_relevant_aa_2&spm=1001.2101.3001.4242.2&utm_relevant_index=4

Erzielung analoger Ergebnisse mittels digitaler Steuerung. Digitale Steuerungen werden verwendet, um Rechteckwellen zu erzeugen, d. h. Signale, die zwischen hohen und niedrigen Pegeln umschalten. Dieser Modus simuliert alle Spannungen auf der Platine von Vcc (5 V) bis aus (GND), indem das Verhältnis zwischen der Zeit, in der sich das Signal im High-Zustand befindet, und der Zeit, in der es sich im Low-Zustand befindet, geändert wird. Die Dauer des High-Pegels wird als Impulsbreite bezeichnet. Ändern oder modulieren Sie die Impulsbreite, um einen sich ändernden Analogwert zu erhalten.

(2) SDA SCL⁶

SDA ist eine bidirektionale Datenleitung und SCL ist eine Taktleitung. Daten werden auf dem TWI-Bus (Two-Wire Serial Interface) übertragen, das höchste Bit wird zuerst gesendet, der Host sendet ein Startsignal, SDA springt während des hohen SCL-Pegels vom hohen Pegel auf den niedrigen Pegel, und dann sendet der Host a Byte Die Daten. Nachdem die Datenübertragung abgeschlossen ist, sendet der Host ein Stoppsignal und SDA springt während des hohen Pegels von SCL von niedrig auf hoch.

4.1.2 Servomotor

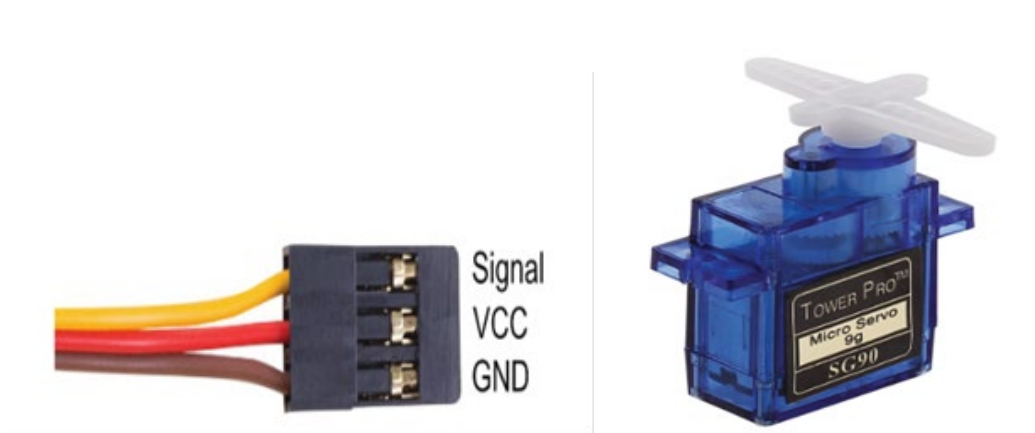


Abbildung 2: Servomotor

Der SG-90 Servomotor ist sehr weit verbreitet im Hobby Bereich und ist für ein paar Euro erhältlich. Man kann den Servo direkt am NodeMCU oder Arduino mit 3,3V – 5V Spannung versorgen, es ist aber empfohlen für einen Dauerbetrieb eine externe Spannungsversorgung bereitzustellen. Für unsere Tests und da wir nur einem Servo Motor betreiben, schliessen wir ihn direkt an. Der SG-90 Servomotor kann, wenn er mit 4,8 Volt betrieben wird ein maximales Drehmoment von 2,5 kg/cm aufbringen und hat eine

Geschwindigkeit von 0,1 Sekunden für eine 60° Drehung. Das Getriebe ist bei den billigeren Modellen aus Plastik, es gibt aber auch Modelle mit Metallgetriebe die viel robuster sind. Die Version mit Plastik Getriebe hat ein Gewicht von 9 Gramm und es sind verschiedene Adapter beigelegt. ⁷

(1) Servobetrieb

Die Drehung der Welle vom SG-90 Servomotor hängt von der Frequenz und dem Tastgrad des PWM-Signals ab. Für die meisten RC-Servomotoren ist eine PWM-Frequenz von 50 Hz erforderlich. Sie können sich zwischen 0° und 180° bei einem Tastgrad des PWM-Signals zwischen 1 und 2 Millisekunden drehen. Ein Tastgrad von einer 1 ms bei einer Frequenz von 50 Hz bewegt die Servowelle in einen Winkel von 0°. 1,5 ms bewegt die Servowelle auf 90° und 2 ms auf 180°. ⁸

⁷ <https://edistechlab.com/sg-90-servo-motor/?v=3a52f3c22ed6>

⁸ <https://edistechlab.com/sg-90-servo-motor/?v=3a52f3c22ed6>

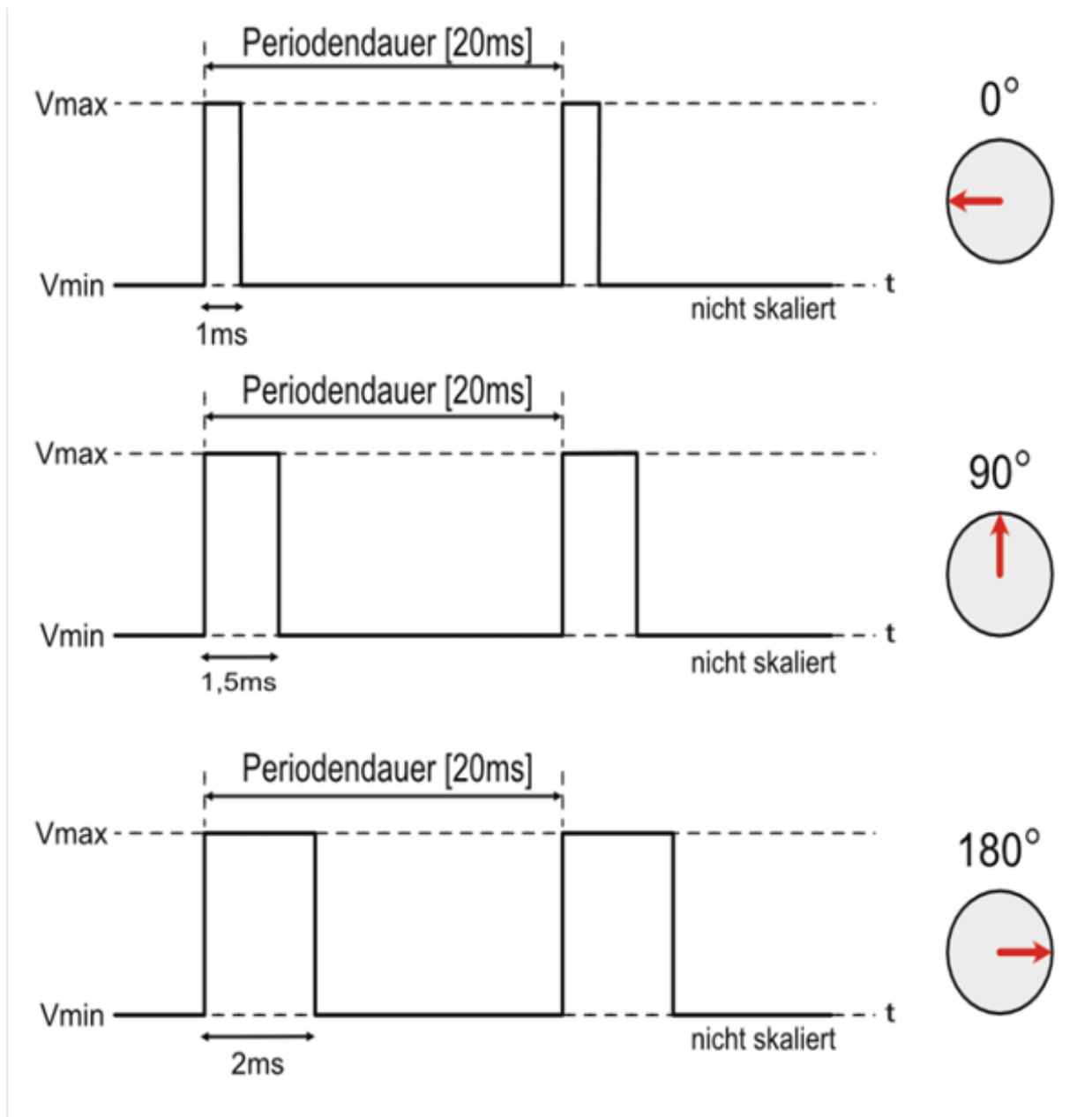


Abbildung 3: Pulsweitenmodulation von Motor⁹

10

(2) Der Anschluss von SG90 Servomotor

Der SG-90 Servomotor hat drei Anschlüsse. VCC ist immer in der Mitte um einen falschen Anschluss zu verhindern und wird an +5 Volt gelegt, die Farbe ist meistens rot. GND mit den Farben schwarz oder braun wird am Mikrocontroller auf GND gelegt und als letztes

⁹ <https://edistechlab.com/sg-90-servo-motor/?v=3a52f3c22ed6>

haben wir noch den Signalanschluss. Dieser hat die Farbe Gelb oder weiß und wird auf den Pin 9 für das PWM Signal angeschlossen.¹¹

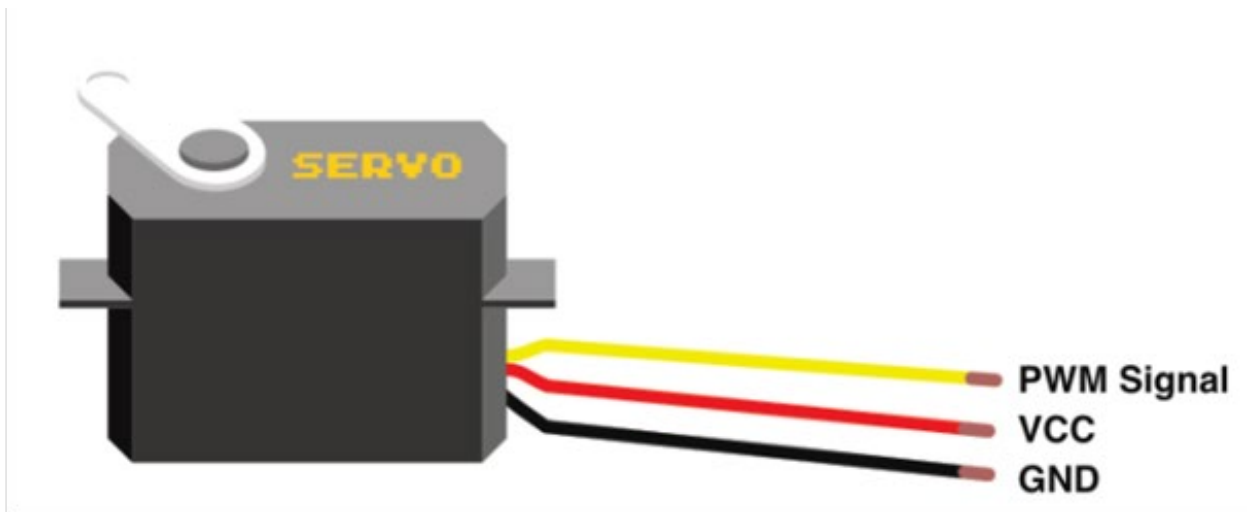


Abbildung 4: Stecker Pin Belegung

(URL: <https://edistechlab.com/sg-90-servo-motor/?v=3a52f3c22ed6>)

4.1.3 Lasersensor

In diesem System verwende ich den Lasersensor VL53LXX-V2. Dieser Sensor kann eine Distanz von bis zu 2m messen und arbeitet dabei mit einer Abtastgeschwindigkeit von 50Hz.

Dies ist ein Time-of-Flight (ToF)-Entfernungsmodul basierend auf dem VL53Lxx-Design mit einer genauen Entfernungsreichweite bis zu 4 Metern, I2C-Schnittstellenkommunikation und geringem Stromverbrauch. Der VL53LXX ist die von ST eingeführte FlightSense-Technologie der zweiten Generation. Anders als der herkömmliche Entfernungssensor verwendet der VL53LXX die ToF-Technologie von ST, die die Entfernung unabhängig von der Farbe und dem Reflexionsvermögen des Ziels messen kann und über eine stärkere Anti-Interferenz-Fähigkeit verfügt.

¹¹ <https://edistechlab.com/sg-90-servo-motor/?v=3a52f3c22ed6>

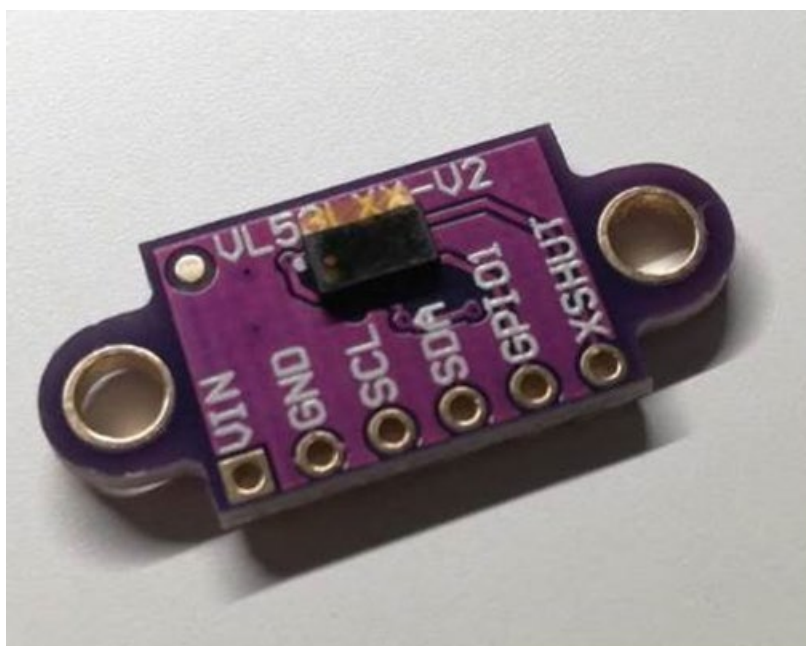


Abbildung 1: Lasersensor VL53L xx-V2

(1) Pin-Schnittstelle

Der Sensor verfügt über 6 Pins welche wie folgt an den Arduino Mega2560 angeschlossen werden.

VL53LXX-V2	Arduino Mega 2560
VIN	5V
GND	GND
SCL	21
SDA	22
GPI01	Keine Verbindung
XSHUT	Kene Verbindung

Tabelle 1 : Pin-Schnittstelle

2)Entfernungseigenschaft

Einzelmessung: In diesem Modus wird nur eine Ranging-Messung ausgelöst. Der VL53L0X-Sensor kehrt in den Standby-Modus zurück und wartet auf den nächsten Trigger.

Kontinuierliche Messung: In diesem Modus werden Entfernungsmessungen kontinuierlich durchgeführt. Sobald die Messung beendet ist, beginnt die nächste Messung sofort, der

Benutzer muss die Entfernungsmessung beenden, um in den Standby-Modus zurückzukehren, und die letzte Messung wird vor dem Stoppen durchgeführt.

Zeitmessung: In diesem Modus werden Entfernungsmessungen kontinuierlich durchgeführt. Nach Ende der Messung startet die nächste Messung erst nach einer benutzerdefinierten Verzögerungszeit. Der Benutzer muss die Entfernungsmessung beenden, um in den Standby-Modus zurückzukehren, und die letzte Messung wird vor dem Herunterfahren durchgeführt.

(3) Datenblatt:

Betriebsspannung	3.3V~5V
Messbereich	40mm~4m
Wellenlänge	940nm
Betriebtemperatur	-20°C~80°C

Tabelle 2:Daten von VL55LXX-V2

4.1.4 Ultraschallsensor



Abbildung 5:Ultraschallsensor HC-SR04¹²

(1) Pin-Schnittstelle

HC-SR04	Arduino Mega 2560
VCC	5V
GND	GND
Trig	7
Echo	8

Tabelle 3:Pin-Schnittstelle von HC-SR04

(2) Datenblatt

Betriebsspannung	DC 5V
Betriebsstrom	15mA
Betriebsfrequenz	40Hz
Messbereich	2cm~4m
Messwinkel	15 grad

Tabelle 4:Daten von HC-SR04

¹² <https://www.sparkfun.com/products/15569>

4.1.5 Tastenmodul



Abbildung 6:Tastenmodul

Dieses Tastenmodul wird verwendet, um den Start und Stopp des Servomotor zu steuern und besteht aus einem Taster in Reihe mit einem Widerstand.

Interner Schaltplan:

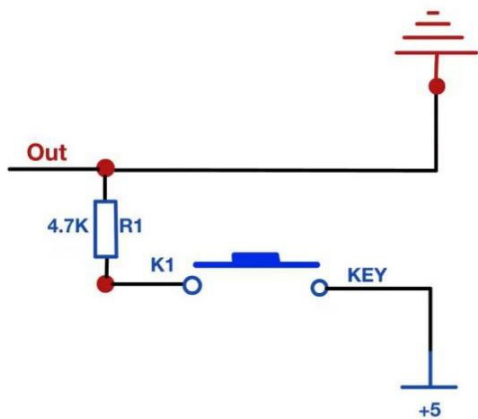


Abbildung 7: Interner Schaltplan von Tastenmodul

(1) Spezifikation

Produkt Name	Tastenmodul
Größe	11*22mm
Spannung	3.5-5V
Ausgang	Digital (drücken Hoch, loslassen tief)
Interface	3P Interface S V G
Platform	MCU;ARM;Raspberrz Cake

Tabelle 5: Spezifikation von Tastenmodu

4.2 Software

4.2.1 Allgemeines Softwarekonzept

Softwareprogrammierung ist in diesem Projekt sehr wichtig. Zuerst müssen wir die Sensorprogrammierung vervollständigen, um die Entfernung zu messen, und dann das

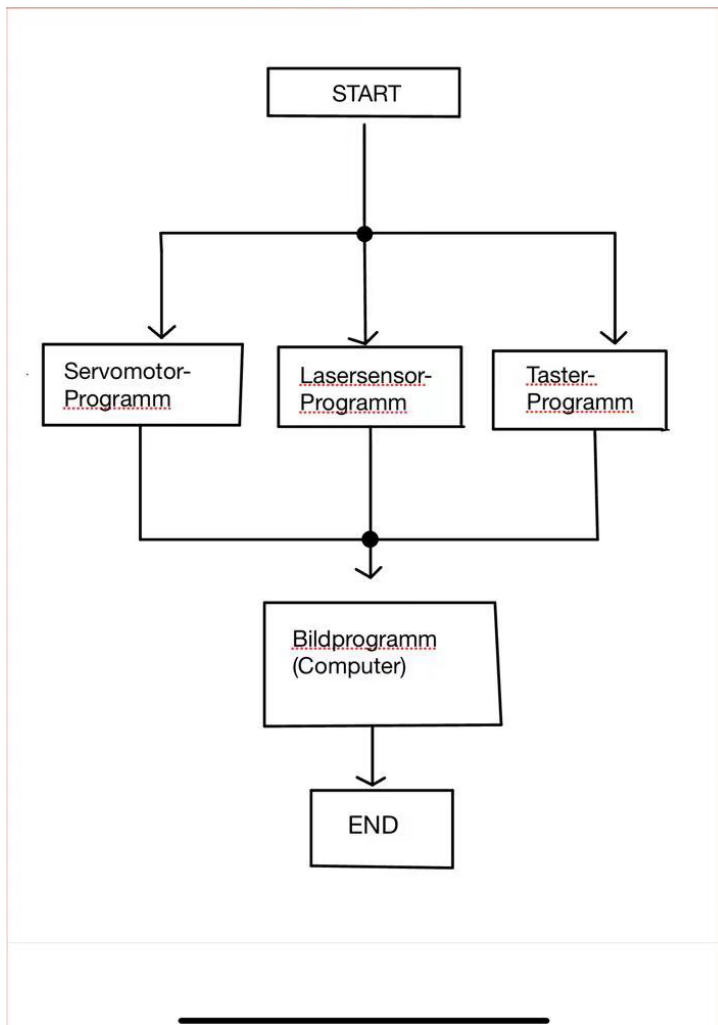


Abbildung 8: Allgemeines Softwarekonzept

Servoprogramm hinzufügen, um den Sensor rotieren zu lassen. Außerdem brauchen wir ein Tastenprogramm, um den Start und Stopp zu steuern, und schließlich diese Programme in den importieren Arduino Mega 2560.

Um die Form des gescannten Objekts anzuzeigen, müssen wir ein Programm auf dem Computer erstellen, indem wir eine Bildverarbeitungssoftware Processing verarbeiten. Die Sensor- und Servodaten werden in Echtzeit über die serielle Schnittstelle an die Verarbeitung übertragen, um die Form zu beschreiben.

4.2.2 Servo-Programm¹³

```
#include <Servo.h>
Servo myservo;
void setup() {

    myservo.attach(9);
}
void loop()
{
    delay(10);
    if (angle == 180)
        v = -1;
    else if (angle == 0)
        v = 1;
    angle += v;
    myservo.write(angle);
}
```

Abbildung 9: Servo-Programm

(1) Erläuterung der Codefunktion

```
#include <Servo.h>
Servo myservo;
```

Diese beiden Codezeilen laden zuerst die Servobibliothek und erstellen dann ein mehrstufiges Objekt myservo.

```
void setup() {

    myservo.attach(9);
}
```

¹³ Ich programmiere selbst durch den gesuchten Basiscode <http://www.xjxhd.cn/?id=678006>

Dieser Code dient zum Initialisieren der Servostifte. Dieser Pin 9 hängt davon ab, an welchem Pin des Arduino Mega2560 PWM das gelbe Kabel des Servos angeschlossen ist.

```
void loop()
{
    delay(10);
    if (angle == 180)
        v = -1;
    else if (angle == 0)
        v = 1;
    angle += v;
    myservo.write(angle);
}
```

Dieser Code dient zur Steuerung der Drehung des Servos. Wenn der Winkel das Maximum erreicht, beginnt der Winkel des Lenkgetriebes allmählich abzunehmen, und wenn der Winkel das Minimum erreicht, nimmt der Winkel allmählich zu, sodass sich das Lenkgetriebe langsam innerhalb von 180 Grad drehen kann. Schließlich werden die Winkeldaten durch die Write() funktion erhalten.

4.2.3 Lasersensor-Programm¹⁴

¹⁴Ich programmiere selbst durch den gesuchten Basiscode
<https://jingyan.baidu.com/article/5d6edee2ce952299eadeec9d.html>

```

#include <Wire.h> //加载I2C通讯库
#include <DFRobot_VL53L0X.h>

DFRobotVL53L0X VL53L0X;

void setup()
{
    Serial.begin(9600);
    Wire.begin();
    VL53L0X.begin(0x50);
    VL53L0X.setMode(Continuous, Low);
    VL53L0X.start();
}

void loop()
{

    int distance = int(VL53L0X.getDistance());

    Serial.println("#"+String(angle)+", "+String(distance));
}

```

Abbildung 10: Lasersensor-Programm

(1) Erläuterung der Codefunktion

```

#include <Wire.h>
#include <DFRobot_VL53L0X.h>

DFRobotVL53L0X VL53L0X;

```

Dieser Code lädt die I2C-Kommunikationsbibliothek, die Servobibliothek und die Entfernungssensorbibliothek.

```

void setup()
{
  Serial.begin(9600);
  Wire.begin();
  VL53L0X.begin(0x50);
  VL53L0X.setMode(Continuous, Low);
  VL53L0X.start();
}

```

Dieser Code stellt zuerst die Baudrate ein, normalerweise 9600, initialisiert dann I2C, stellt dann die Kommunikationsadresse, den Messmodus und die Genauigkeit des Entfernungssensors ein und startet schließlich den Entfernungssensor

`Serial.begin(9600)`: Die `Serial.begin`-Funktion wird verwendet, um die Baudrate der seriellen Schnittstelle einzustellen.

`Wire.begin()`: Es initialisiert Wire und fügt zu I2C hinzu.

`VL53L0X.begin(0x50)`: Es stellt die Kommunikationsadresse des Entfernungssensors ein.

`VL53L0X.setMode(Continuous, Low)`: Es stellt den Messmodus und die Genauigkeit ein.

`VL53L0X.start()`: Es aktiviert den Entfernungssensor.

```

void loop()
{

  int distance = int(VL53L0X.getDistance());

  Serial.println("#"+String(angle)+", "+String(distance));
}

```

Dieser Code erhält zuerst den Abstand in mm und sendet dann die Daten über die serielle Schnittstelle

`VL53L0X.getDistance()`: Es erhält die gemessene Entfernung.

`Serial.println("#" + String(angle) + "," + String(distance))`: Es gibt serielle Daten aus.

4.2.4 Bildprogramm(Anzeigen von dem Form auf Computer)¹⁵

Um die Umrisse des Scans zu zeigen, habe ich die Verarbeitungsbild-Programmiersoftware Processing verwendet, um ein Bildgebungsprogramm zu machen.

In der Processing Software erstelle ich ein Koordinatensystem und dann einen Punktsatz mit 181 gemessenen Punkten, stelle dann die Koordinaten jedes Punkts im Koordinatensystem durch Vektoren dar und zeichne dann den Form durch die Software.

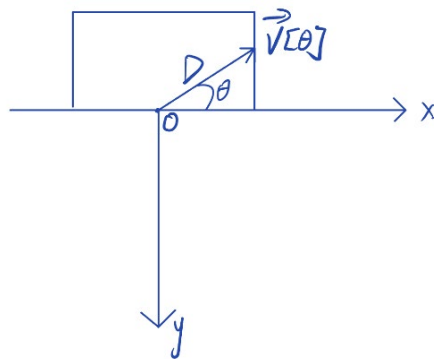


Abbildung 11: Koordinatensystem in Processing

die mathematische Formel: $\vec{v}[\theta] = (D \cdot \cos\theta, -D \cdot \sin\theta)$

Der vollständige Code lautet wie folgt:

¹⁵Ich programmiere selbst durch den gesuchten Basiscode. https://blog.csdn.net/fangjin_kl/article/details/40428655 und https://blog.csdn.net/qj_38249231/article/details/103773813

```

import processing.serial.*;

Serial myPort;
int angle, distance;
PVector[] points = new PVector[181];

void setup()
{
    size(800, 600);

    myPort = new Serial(this, "COM3", 9600);
    myPort.bufferUntil('\n');

    for(int i=0; i<181; i++)
    {
        points[i] = new PVector(width/2, height);
    }
}

```

Abbildung 2: Bildprogramm(1)

```

void draw()
{
    background(0);
    push();
    translate(width/2, height);
    noFill();
    stroke(255);
    strokeWeight(2);
    beginShape();
    for (int i=0; i<181; i++)
    {
        vertex(points[i].x, points[i].y);
        //strokeWeight(dist(points[i].x, points[i].y, 0, 0)/60);
        //point(points[i].x, points[i].y);
    }
    endShape();

    rotate(radians(-angle));
    stroke(255, 0, 0);
    strokeWeight(3);
    line(0, 0, distance, 0);
    pop();
}

```

Abbildung 3: Bildprogramm(2)

```

void serialEvent(Serial p)
{
    if (p.read()=='#')
    {
        String inString = trim(p.readString());
        String inData[] = split(inString, ',');
        angle = parseInt(inData[0]);
        distance = constrain(parseInt(inData[1]), 0, width/2);

        points[angle].x = distance*cos(radians(angle));
        points[angle].y = - distance*sin(radians(angle));
        print(inString);
    } else
    {
        while (p.available()>0)
        {
            p.read();
        }
    }
}

```

Abbildung 4: Bildprogramm(3)

(1) Erläuterung der Codefunktion

```

import processing.serial.*;
Serial myPort;
int angle, distance;
PVector[] points = new PVector[181];

```

Dieser Code lädt zuerst die serielle Portbibliothek, erstellt dann das serielle Portobjekt "myport", definiert dann den Winkel, die Entfernung und den Vektorpunktsatz.

Serial myPort: Es erstellt ein serielles Port-Objekt.

PVector[] points = new PVector[181]: Es definiert eine Vektorpunktmenge.

```
void setup()
{
  size(800,600);
  myPort = new Serial(this, "COM3" , 9600);
  myPort.bufferUntil('\n');
  for(int i=0;i<181;i++)
  {
    points[i] = new PVector(width/2,height);
  }
}
```

Dieser Code stellt zuerst die Vollbildanzeige ein, initialisiert dann den seriellen Anschluss, liest die Daten des seriellen Anschlusses und stellt die Baudrate des seriellen Anschlusses und die Stoppmarkierung für die Daten des seriellen Anschlusses ein. Anschließend wird der Vektorpunktsatz initialisiert.

`size(800,600)`: Es stellt die Fenstergröße ein.

`myPort = new Serial(this, "COM3" , 9600)`: Es stellt die serielle Schnittstelle und die Baudrate ein.

`myPort.bufferUntil('\n')`: Es speichert alle Nachrichten von der seriellen Schnittstelle, bis „/n“ empfangen wird.


```

void draw()
{
  background(0);
  push();
  translate(width/2, height);
  noFill();
  stroke(255);
  strokeWeight(2);
  beginShape();
  for (int i=0; i<181; i++)
  {
    vertex(points[i].x, points[i].y);
    //strokeWeight(dist(points[i].x, points[i].y,0,0)/60);
    //point(points[i].x, points[i].y);
  }
  endShape();

  rotate(radians(-angle));
  stroke(255, 0, 0);
  strokeWeight(3);
  line(0, 0, distance, 0);
  pop();
}

```

Dieser Code legt den Hintergrund fest und zeichnet dann den Scannhintergrund und die Scanlinien.

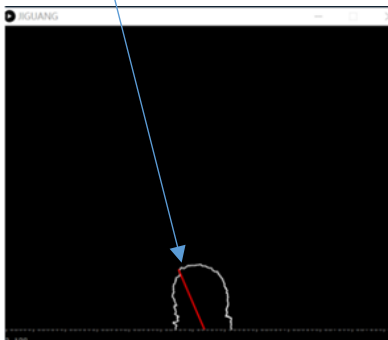


Abbildung 12: Zeigen von dem Scannhintergrund und der Scannlinien

Außerdem habe ich eine For-Schleife verwendet, um die Vertex()-Funktion mehrmals aufzurufen, um die geometrischen Scheitelpunkte zur Beschreibung der Form zu bestimmen.

draw(): Die Funktion draw() wird einmal pro Frame ausgeführt. Die Standardbildrate beträgt 60 fps. Das Programm ruft diese Funktion kontinuierlich auf, um den Bildschirm zu aktualisieren und eine Animation zu erstellen.

background(): Mit dieser Funktion wird die Hintergrundfarbe eingestellt.

push(): Diese Funktion speichert die transformierten Ursprungscoordinate.

pop(): Diese Funktion kann zu den gespeicherten Ursprungskoordinaten zurückkehren.

`translate(width/2, height)`: Diese Funktion übersetzt den Ursprung des Koordinatensystems in `(width/2, height)`.

`noFill()`: Diese Funktion deaktiviert die innere Farbe des Diagramms.

`stroke()`: Diese Funktion legt die Rahmenfarbe fest.

`strokeWeight()`: Diese Funktion legt die Dicke des Randes fest.

`beginShape()`: Diese Funktion dient dazu, mit dem Zeichnen zu beginnen.

`endShape()`: Diese Funktion beendet das Zeichnen.

`vertex()`: Diese Funktion kann mehrfach aufgerufen werden, um die Scheitelkoordinaten der Zeichnungsgrafik zu bestimmen.

`radians()`: Diese Funktion wandelt Grad in Bogenmaß um.

`rotate(n)`: Diese Funktion dreht das Koordinatensystem um `n` Bogenmaß um den Ursprung.

`line(0, 0, distance, 0)`: Diese Funktion ist eine Zeichenlinie, die ersten beiden Zahlen sind die Startkoordinaten und die letzten beiden Zahlen sind die Endkoordinaten.

```
void serialEvent(Serial p)
{
  if (p.read()=='#')
  {
    String inString = trim(p.readString());
    String inData[] = split(inString, ',');
    angle = parseInt(inData[0]);
    distance = constrain(parseInt(inData[1]), 0, width/2);

    points[angle].x = distance*cos(radians(angle));
    points[angle].y = - distance*sin(radians(angle));
  } else
  {
    while (p.available()>0)
    {
      p.read();
    }
  }
  //print(p.readString());
}
```

Dieser Code ist ein Ereignis der seriellen Schnittstelle. Verwenden Sie die if-Anweisung, um festzustellen, ob der Datenrahmen-Header erkannt wird. Wenn ja, lesen Sie die Daten, analysieren Sie die Daten, erhalten Sie den Winkel und die Entfernung und konvertieren Sie dann den Winkel und die Entfernung in X, Y Koordinaten und speichern sie im Vektorpunktsatz.

trim(): Diese Funktion entfernt die Leerzeichen vor und nach der Zeichenfolge.

p.readString(): Diese Funktion liest die Daten der seriellen Schnittstelle.

split(): Diese Funktion teilt die Zeichenfolge gemäß dem angegebenen Trennzeichen.

parseInt(): Diese Funktion analysiert einen String und gibt eine ganze Zahl zurück.

constrain(): constrain(value,min,max), value ist der angegebene Variablenwert(min~max).

4.2.5 Tastenmodul-Programm¹⁶

(1) Erläuterung der Codefunktion

```
volatile bool start_flag = false;
volatile uint32_t last_time;
```

zunächst definiert es das Start-Flag und die letzte Interrupt-Auslösezeit.

```
void setup() {
  pinMode(key_pin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(key_pin), key_zhong_duan, FALLING);
}
```

¹⁶Ich programmiere selbst durch den gesuchten Basiscode. <https://www.csdn.net/tags/MtzaEg0sODY2OS1ibG9n.html>

Es initialisiert Tastenstift und einstellt Interrupt-Trigger-Programm, Trigger mit fallender Flanke.

Volatile: Wenn die Variable mit volatile modifiziert wird, garantiert der Compiler, dass die

Lese- und Schreiboperationen auf der Variable nicht optimiert werden, wodurch sichergestellt

wird, dass die Variable i im Originalprogramm rechtzeitig erfasst werden kann, nachdem die

Variable i durch die geändert wurde externes Programm.

pinMode(): Mit der Funktion pinMode() können Sie die Pins des Arduino auf die folgenden drei Modi konfigurieren. (Output, Input, Input_PULLUP).

attachInterrupt(digitalPinToInterrupt(key_pin), ISR, mode): Die Funktion attachmentInterrupt() wird verwendet, um die ISR (Interrupt Service Routine) für das Arduino-Entwicklungsboard zu setzen und auszuführen.

```
void key_zhong_duan()
{
    if (millis() - last_time > 300)
    {
        last_time = millis();
        start_flag = !start_flag;
    }
}
```

Dieser Code ist die Interrupt-Service-Programm(ISR). „mills() - last_time >300“ ist zu entprellen. Mills() kann aktuelle Zeit erhalten. Last_time ist die letzte Interruptzeit. Wenn „mills() - last_time >300“ erfüllt ist, ist Interrupt-Service-Programm funktioniert.

Mills(): Die Funktion kann die aktuelle Zeit erhalten.

5 Inbetriebnahme / Systemtest

5.1 Aufbau von Hard- und Software

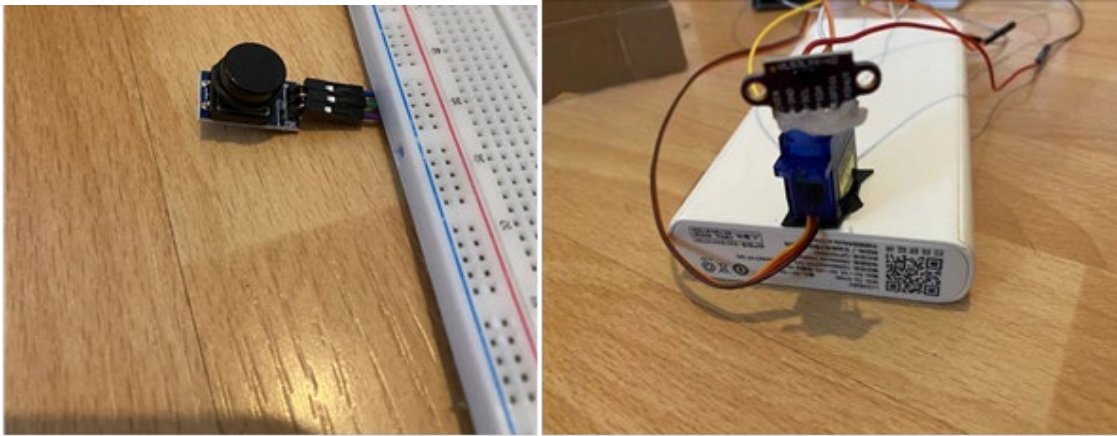


Abbildung 13: Lasersensor des Aufbaus von Hardware

In diesem Projekt werden die Ultraschallsensor oder Lasersensor, Servomotor und Tastenmodule über Dupont-Kabel mit dem Arduino Meag2560 verbunden, und dann wird das Arduino-Board über USB mit dem Computer verbunden und über die Arduino IDE-Software programmiert. Dann fixiere ich den Sensor mit Schmelzkleber auf dem Servo und klebe ich Servo auf einen Sockel. In der Option Tools/Boards muss der richtige Boardmanager (Arduino Mega 2560) ausgewählt werden (serieller Port aus: COM3).

Um die Umrisse des gescannten Objekts anzuzeigen, musste ich die Verarbeitungsbild-Programmiersoftware verwenden, um die Daten vom Arduino über die serielle Schnittstelle zur Verarbeitung zu übertragen und dann zu verarbeiten.



Abbildung 14: die Verarbeitungsbild-Programmiersoftware Processing

5.2 Test

Zuerst schlieÙe ich die Sensor, Servomotor und Tasten an das Arduino-Board an und lade dann das programmierte Programm auf das Arduino-Board in der Arduino IDE hoch. Ich öffne das Programm in Processing, drücke die Taste, startet das System den Scanvorgang, drücke erneut, um zu stoppen, und ich beobachte dann die Form bei Processing.

5.2.1 Messergebnis von dem Ultraschallsensor



Abbildung 15:Ergebnis von Ultraschallsensor

Wie aus dem obigen Bild ersichtlich ist, kann der Ultraschallsensor den Abstand jedes Punkts auf dem Objekt nicht genau messen, was zu einem großen Fehler im Messergebnis führt. Deshalb glaube ich, Ultraschall nicht geeignet für das Projekt zu sein.

Die Gründe werden wie folgt analysiert:

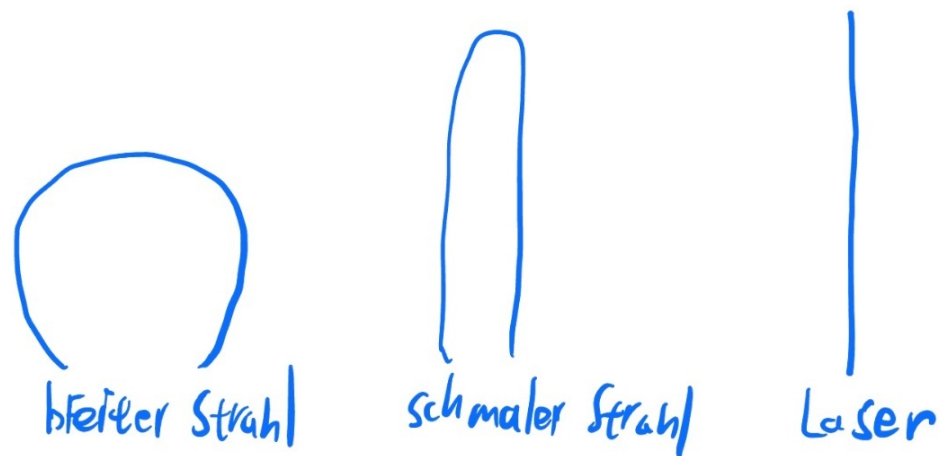


Abbildung 16: Vergleich von verschiedene Strahlen

Ultraschallsensoren werden hauptsächlich verwendet, um den Abstand von Objekten und deren Orientierung relativ zum Sensor zu erfassen. Der Breitstrahlsensor erkennt jedes Objekt innerhalb des Strahlbereichs. Er kann die Entfernung des Objekts erkennen, aber er kann nicht den Azimut des Objekts erkennen. Der maximale Fehlerwinkel beträgt etwa 100 Grad. Wenn es verwendet wird, um festzustellen, ob ein Objekt vorhanden ist ist ein breiter Strahl Sensor optimal. Ein schmaler Strahl kann einen genaueren Azimutwinkel erzielen als ein breiter Strahl, aber er ist nicht ideal, um die Entfernung zu einem Punkt genau zu messen. Im Vergleich dazu ähnelt der Laserstrahl einer Linie, die die Entfernung eines Punktes genau messen kann.

5.2.2 Messergebnis von dem Lasersensor

Ich habe drei Fälle zum Testen eingerichtet.

(1) rechteckiges Objekt

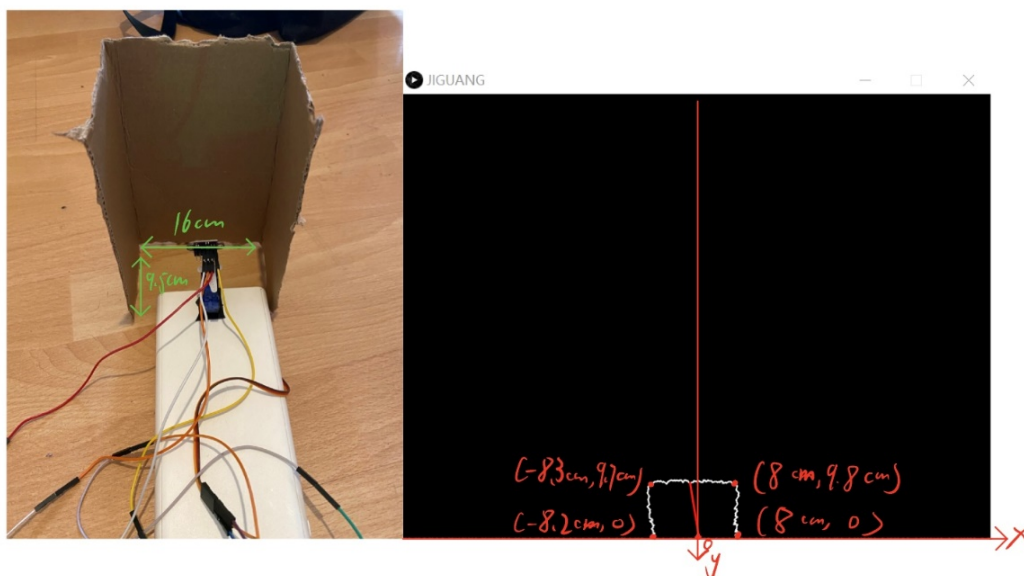


Abbildung 17: Test von rechteckiges Objekt

(2) gekrümmte Fläche

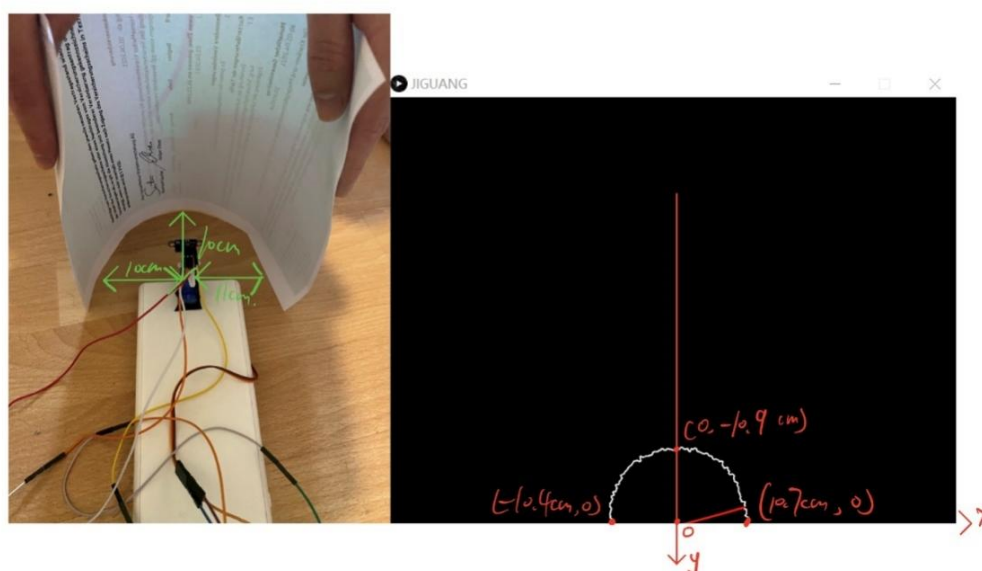


Abbildung 18: Test von gekrümmte Fläche

(3) Welle

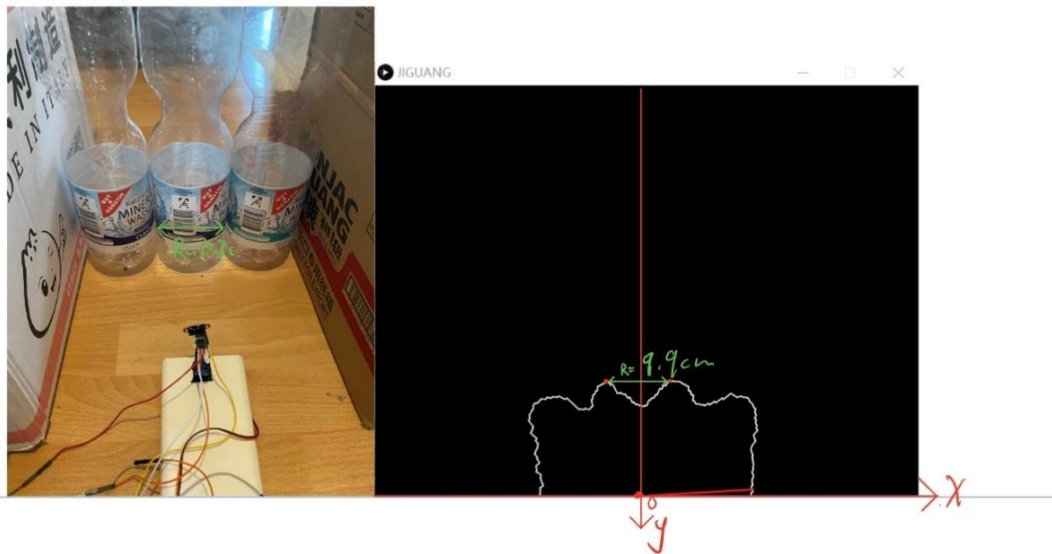


Abbildung 19: Test von Welle

Aus den obigen drei Testergebnissen ist ersichtlich, dass das erste und das zweite Ergebnis erfolgreich sind und die Messung der dritten Wellenform nicht genau genug ist.

Gründe für die Fehlmessung im dritten Fall:

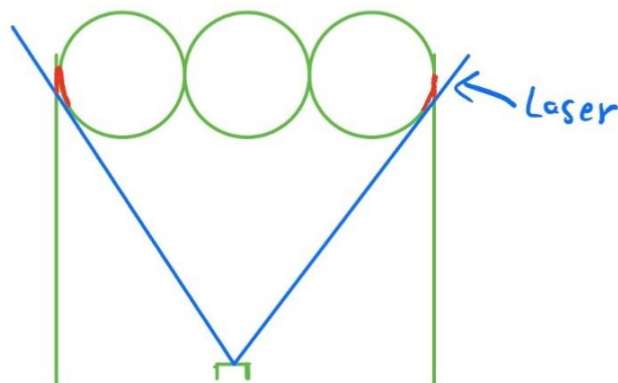


Abbildung 20: Analyse von der Fehlmessung der Wellw

Der Lasersensor kann nur die Tangentialpunkte des linken und rechten Kreises und den Laser am Maximum messen, sodass der rote Bereich der Messblindbereich ist, was zu einer ungenauen Messung führt.

6 Zusammenfassung

Das gesamte System kann durch Drehen des Entfernungssensors die geometrische Form innerhalb eines Halbkreises mit einem Radius von 1 Meter scannen und den Start und Stopp über Tasten steuern. Es besteht aus Entfernungssensor, Servo, Tastenmodul und Arduino Mega 2560. Darüber hinaus ist es erforderlich, die gemessene Form über die Verarbeitungssoftware Processing auf dem Computer anzuzeigen. Bei der Processing wird der gemessene Abstand durch den Vektor, die mathematische Formel der trigonometrischen Funktion, in Koordinaten umgewandelt, und die Form wird durch diese Punkte gezeichnet.

Während des Tests kann festgestellt werden, dass die Entfernung zum Punkt nicht genau gemessen werden kann, da der Ultraschallstrahl zu breit ist. Der Lasersensor wurde jedoch erfolgreich getestet, um den Form zu messen. Der Messfehler liegt innerhalb von 1 cm. Der Lasersensor hat jedoch noch Mängel: Beim Auftreffen auf einen Kreis oder eine Stelle, auf die der Laserstrahl nicht direkt gerichtet werden kann, entsteht ein Messblindfleck. Für diesen Mangel müssen mehrere Sensoren hinzugefügt werden, um Messblindfleck bei der Messung zu vermeiden oder durch optische 3D-Technologie. Ich hoffe, dass ich im Masterstudiengang professionelles Wissen weiter studieren kann, um dieses Projekt zu verbessern.

Literatur

„Die optische 3D-Formmessungstechnik“ Online:

<https://patents.google.com/patent/CN100520287C/zh>

Prinzip von der Ultraschallentfernungsmessung Online:

<https://baike.baidu.com/item/%E8%B6%85%E5%A3%B0%E6%B3%A2%E6%B5%8B%E8%B7%9D/8544144>

Prinzip von der Lasersensorentfernungsmessung Online:

<https://baike.baidu.com/item/TOF%E6%8A%80%E6%9C%AF/5198119?fr=aladdin>

<https://blog.csdn.net/limingmin2020/article/details/124141523>

„Arduino Mega 2560“ Online: <https://arduino-projekte.info/arduino-mega-2560/>

„PWM SCL SDA“ Online:

https://blog.csdn.net/hypecycle/article/details/110674844?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_baidulandingword~default-1-110674844-blog-124407506.pc_relevant_aa_2&spm=1001.2101.3001.4242.2&utm_relevant_index=4

„HC-SR04“ Online: <https://www.sparkfun.com/products/15569>

„Servomotor“ Online: <https://edistechlab.com/sg-90-servo-motor/?v=3a52f3c22ed6>

„Servomotor-Programm“ Online: <http://www.xjxhd.cn/?id=678006>

„LasersensorProgramm“ Online:

<https://jingyan.baidu.com/article/5d6edee2ce952299eadeec9d.html>

„Bildprogramm“ Online: https://blog.csdn.net/fangjin_kl/article/details/40428655 und

https://blog.csdn.net/qq_38249231/article/details/103773813

Anlagen

Arduinoprogramm	A-I
Processingprogramm	A-III

Anlagen, Arduinoprogramm

Arduino IDE:

```
#include <Servo.h>
#include <Wire.h>
#include <DFRobot_VL53L0X.h>

Servo myservo;
DFRobotVL53L0X VL53L0X;

int v, angle;
const int key_pin = 2;
volatile bool start_flag = false;
volatile uint32_t last_time;

void setup()
{
    Serial.begin(9600);
    myservo.attach(9);
    Wire.begin();
    VL53L0X.begin(0x50);
    VL53L0X.setMode(Continuous, Low);
    VL53L0X.start();
    pinMode(key_pin, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(key_pin), key_zhong_duan, FALLING);
}

void loop()
{
    if (start_flag)
    {
        delay(10);
        if (angle == 180)
            v = -1;
        else if (angle == 0)
            v = 1;

        angle += v;
        myservo.write(angle);
        int distance = int(VL53L0X.getDistance());
        Serial.println("#" + String(angle) + "," + String(distance));
    }
}

void key_zhong_duan()
{
    if (millis() - last_time > 300)
```

```
{  
  last_time = millis();  
  start_flag = !start_flag;  
}  
}
```

Anlagen, Processingprogramm

Processing:

```
import processing.serial.*;

Serial myPort;

int angle, distance;
PVector[] points = new PVector[181];

void setup()
{
    size(800,600);
    myPort = new Serial(this, "COM3" , 9600);
    myPort.bufferUntil('\n');
    for(int i=0;i<181;i++)
    {
        points[i] = new PVector(width/2,height);
    }
}

void draw()
{
    background(0);

    push();
    translate(width/2, height);
    noFill();
    stroke(255);
    strokeWeight(2);
    beginShape();
    for (int i=0; i<181; i++)
    {
        vertex(points[i].x, points[i].y);
        //strokeWeight(dist(points[i].x, points[i].y,0,0)/60);
        //point(points[i].x, points[i].y);
    }
    endShape();

    rotate(radians(-angle));
    stroke(255, 0, 0);
    strokeWeight(3);
    line(0, 0, distance, 0);
    pop();
}
```

```
void serialEvent(Serial p)
{
  if (p.read()=='#')
  {
    String inString = trim(p.readString());
    String inData[] = split(inString, ',');
    angle = parseInt(inData[0]);
    distance = constrain(parseInt(inData[1]), 0, width/2);
    points[angle].x = distance*cos(radians(angle));
    points[angle].y = - distance*sin(radians(angle));
    print(inString);
  } else
  {
    while (p.available()>0)
    {
      p.read();
    }
  }
}
```


Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 25.08.2022

Yuwen Zhou