

---

# **BACHELORARBEIT**

---

Herr  
**He Ziang**

**Zweiseitige  
Raumzugangskontrolle  
mit Zählung**

Mittweida, 2023

Elektro- und Informationstechnik

---

# **BACHELORARBEIT**

---

## **Zweiseitige Raumzugangskontrolle mit Zählung**

Autor:  
**Herr**

**He Ziang**

Studiengang:  
**Elektro- und Informationstechnik**

Seminargruppe:  
**EI19wA-BC**

Erstprüfer:  
**Prof. Dr. -Ing. Michael Kuhl**

Einreichung:  
**Mittweida, 14.08.2023**

Verteidigung/Bewertung:  
**Mittweida, 2023**

Elektro- und Informationstechnik

---

**BACHELOR THESIS**

---

**Two-sided room access control  
with counting**

author:

**Mr.**

**He Ziang**

course of studies:

**Electro- and Informationstechnology**

seminar group:

**EI19wA-BC**

first examiner:

**Prof. Dr. -Ing. Michael Kuhl**

submission:

**Mittweida, 14.08.2023**

defence/ evaluation:

**Mittweida, 2023**

## **Bibliografische Beschreibung:**

He Ziang:

Zweiseitige Raumzugangskontrolle mit Zählung. - 2023. Seiten 41.  
Mittweida, Hochschule Mittweida, Fakultät Ingenieurwissenschaften,  
Bachelor- arbeit, 2023

# Inhalt

<b>Inhalt</b> .....	<b>I</b>
<b>Abbildungsverzeichnis</b> .....	<b>III</b>
<b>Abkürzungsverzeichnis</b> .....	<b>IV</b>
<b>1. Einleitung</b> .....	<b>1</b>
<b>2. Aufgabenstellung</b> .....	<b>2</b>
<b>3. These</b> .....	<b>3</b>
<b>4. Funktionales Design des Systems</b> .....	<b>4</b>
<b>4.1. Hardware</b> .....	<b>6</b>
4.1.1. Arduino Mega 2560 .....	6
4.1.2. HC-SR04 .....	7
4.1.3. Micro-SD-Karten-Modul .....	9
4.1.4. LCD .....	10
4.1.5. Servo .....	11
4.1.6. Tasten .....	12
4.1.7. Schaltplan .....	13
<b>4.2. Software</b> .....	<b>13</b>
4.2.1. Softwarekonzept .....	14
4.2.2. Flussdiagramm .....	15
4.2.3. Bibliotheksdateien .....	16
4.2.4. Servo-Programm .....	17
4.2.5. SD-Karten-Modul-Programm .....	18
4.2.6. Probleme .....	19

<u>Inhalt</u>	<u>II</u>
<b>5. Technischer Aufbau</b> .....	<b>21</b>
<b>6. Experimenteller Teil</b> .....	<b>23</b>
<b>6.1 Simulations-Experimente</b> .....	<b>23</b>
6.1.1 Breadboard-Simulation .....	23
6.1.2 Breadboard-Simulation .....	23
<b>6.2 Experimentelles Flussdiagramm</b> .....	<b>24</b>
<b>6.3 Beispiel-Referenz</b> .....	<b>25</b>
<b>6.4 Simulationslauf</b> .....	<b>25</b>
<b>7. Zusammenfassung</b> .....	<b>27</b>
<b>8. Literaturverzeichnis</b> .....	<b>28</b>
<b>9. Vollständiger Software-Code</b> .....	<b>29</b>
<b>10. Selbstständigkeitserklärung</b> .....	<b>33</b>

# Abbildungsverzeichnis

Abbildung 1: Arduino Mega2560 .....	6
Abbildung 2:HC-SR04 .....	8
Abbildung 3:HC-SR04 .....	8
Abbildung 4: Modellnummer von HC-SR04 .....	8
Abbildung 5: Modellnummer von HC-SR04 .....	9
Abbildung 6: Micro-SD-Karten-Modul .....	9
Abbildung 7: LCD .....	10
Abbildung 8: LCD .....	10
Abbildung 9: Servo .....	12
Abbildung 10: Tasten .....	12
Abbildung 11: Schaltplan .....	13
Abbildung 12: Softwarekonzept .....	14
Abbildung 13: Flussdiagramm .....	15
Abbildung 14: Servo-Programm .....	17
Abbildung 15: SD-Karten-Modul-Programm .....	18
Abbildung 16: Probleme .....	20
Abbildung 17: PCB-Leiterplatte .....	21
Abbildung 18: Gelötete PCB-Leiterplatte .....	22
Abbildung 19: Breadboard-Simulation .....	23
Abbildung 20: Breadboard-Simulation .....	24
Abbildung 21: Experimentelles Flussdiagramm .....	24
Abbildung 22: Beispiel-Referenz .....	25
Abbildung 23: Praktische Erfahrung .....	25
Abbildung 24:Praktische Erfahrung .....	26

## Abkürzungsverzeichnis

µC	Mikrocontroller
SD-Karte	Secure Digital Memory Card
PC	Personal Computer
o.ä.	oder Ähnliche
ggf.	gegebenenfalls
TF-Karte	TransFlash-Karte
LCD	Liquid Crystal Display
PWM	Pulse Width Modulation
UART	Universal Receiver/Transmitter Asynchronous
USB	Universal Serial Bus
ICSP	in-circuit serial programming
usw.	und so weiter
PCB	printed circuit board



# 1. Einleitung

Der technologische Fortschritt hat die Automatisierung und Intelligenz in unserem Leben stark gefördert. Im täglichen Leben existieren Türen in den unterschiedlichsten Lebenswelten als Durchgänge, durch die Menschen gehen müssen. Die Steuerungsanforderungen für Türen sind vielfältig, einschließlich der Notwendigkeit, dass Sicherheitsschleusen einer komplexen Authentifizierung unterzogen werden müssen, um einen Alarm zu öffnen oder auszulösen. Die automatischen Sensortüren in der Mall können die Ankunft von Personen automatisch erkennen und sich automatisch öffnen, während die Kühlhaustüren die Funktion haben, sich automatisch zu öffnen und zu schließen. Mit der zunehmenden Automatisierung können Menschen den Zustand der Tür, d.h. offen oder geschlossen, genau bestimmen, was zu einem Trend zur Diversifizierung der Nachfrage nach Türsteuerung führt. Darüber hinaus gibt es im Kontext globaler Epidemien ein klareres Verständnis für die Bedrohung durch Infektionskrankheiten. Daher ist es in dieser besonderen Zeit besonders wichtig geworden, den Zustand der Tür zu

überwachen und die Anzahl der Personen zu erfassen, die durch die Tür gehen. Wenn ein Feuer ausbricht, kann ein System, das automatisch die Anzahl der Personen aufzeichnet, dabei helfen, festzustellen, wie viele Personen sich noch im Raum befinden. Daher kann der Einsatz automatisierter Systeme anstelle herkömmlicher manueller Registrierungsmethoden diesen Prozess erleichtern.

## 2. Aufgabenstellung

### **Bearbeitungszeitraum:**

Ziel der Arbeit ist der prototypische Aufbau eines Systems zur Überwachung eines Raumes mittels einer elektronischen Zugangskontrolle inklusive der Zählung der Personen die den Raum betreten oder verlassen.

Hierzu sollen laufend eine Zählung vorgenommen, die Daten über einen vorgegebenen Mikrocontroller ( $\mu\text{C}$ ) auf einer SD-Karte gespeichert und über eine Anzeige sowie über einen PC ausgegeben werden.

Die Einstellung der Parameter kann über Eingabeelemente (Tasten, Touch-Pad ) direkt auf dem  $\mu\text{C}$  oder über einem PC und die serielle Schnittstelle stattfinden.

Variablen, Schwellwerte u.ä. sind nicht fest im Code zu hinterlegen, sondern von außen einstellbar zu gestalten.

Für die Elektronik soll eine eigene Leiterplatte entwickelt und aufgebaut werden.

Diese Zielstellung ist nur eine grobe Vorgabe und ist durch den Studenten selbst weiter zu einem sinnvollen System zu detaillieren.

### **Teilaufgaben der Bachelorarbeit:**

- Begriffsbestimmung und Definitionen (Glossar, Schlüsselworte)
- Einarbeitung in die Thematik
- Erarbeitung eines Systemkonzeptes (Hardware, Software, realer Aufbau)
- unter Nutzung eines vorgegebenen Mikrokontrollers, ggf. Übergabe der Parameter über einen PC
- Entwurf und Umsetzung der Schaltung inklusive eigener Leiterplatte
- Durchführung und Auswertung von Basisversuchen
- Bewertung, Darstellung und Dokumentation der Projektergebnisse

### 3. These

Das Ziel dieser Arbeit ist es, die Aufzeichnung und Speicherung von Personen durch Systemdesign, Hardwaredesign und Programmierung zu realisieren. Das Systemdesign wird die Integration von Sensoren, Aktoren und SD-Modulen berücksichtigen. Das Hardware-Design umfasst Schaltkreise und Verbindungslayouts, um eine ordnungsgemäße Verbindung zum Arduino Mega 2560 zu gewährleisten. Die Programmierung ermöglicht es, Echtzeit-Sensordaten mit Zählern zu interagieren und die Datensätze auf einer angeschlossenen TF-Karte zu speichern. Erfüllen Sie durch integriertes Anwendungsdesign und -implementierung die Anforderungen an die Personenzählung und -speicherung.

## 4. Funktionales Design des Systems

Der Entwurf dieses Projekts zielt darauf ab, den Bau von zwei Türen zu realisieren, von denen eine als Eingang und die andere als Ausgang fungiert. Dieses Design ist so konzipiert, dass es einen bequemen Zugang bietet, um den Bedürfnissen der Benutzer gerecht zu werden.

Bei früheren Entwürfen stellten wir fest, dass es nicht möglich war, Personen, die gleichzeitig eintraten, effektiv zu erkennen und zu zählen. Um dieses Problem zu lösen, planen wir einen physischen Ansatz, d. h. die Gestaltung der Ein- und Ausgänge. Konkret werden wir einen schmalen Designeingang in Kombination mit einem Türmechanismus verwenden, der sich automatisch öffnet und schließt.

Durch die Verwendung eines schmalen Eingangsdesigns ist es möglich, den Durchgang von jeweils nur einer Person zu begrenzen und sicherzustellen, dass die eintretenden Personen einzeln erkannt und gezählt werden. Dieses Design kann genauere Statistiken über den Personenfluss liefern, wodurch Verwirrung und ungenaue Zählung von zu vielen Personen gleichzeitig vermieden werden.

Gleichzeitig wird ein Türmechanismus verwendet, der sich automatisch öffnet und schließt, um sicherzustellen, dass sich die Tür erst nach dem Passieren des Eingangs durch eine Person schließt und den Durchgang der nächsten Person automatisch öffnet. Dieser automatisierte Türmechanismus kann die Personalfrequenz effizient verwalten, mehr Sicherheit und Sprachfluss bieten und gleichzeitig die Notwendigkeit manueller Eingriffe reduzieren.

Es wird erwartet, dass durch die Übernahme dieses Ansatzes für das physische Design das Problem gelöst wird, dass Personen gleichzeitig eintreten, die bisher nicht in der Lage waren, sie zu erkennen und zu zählen. Dies liefert genaue Daten und ein besseres Verständnis der Mobilität für eine effektive Ressourcenplanung und -verwaltung.

Um die Funktion der Aufzeichnung der Anzahl der Personen zu realisieren, werden ein SD-Kartenmodul, zwei Ultraschallsensoren, zwei Servos und drei Tasten an den Arduino Mega 2560 angeschlossen.

Ein Ultraschallsensor ist ein Sensor, der zur Entfernungsmessung verwendet wird und den Abstand zwischen einem Objekt und einem Sensor berechnet, indem ein Ultraschallimpuls gesendet und sein Echo empfangen wird. Durch den Anschluss von zwei Ultraschallsensoren an die entsprechenden Pins des Arduino Mega 2560 kann der Abstand zwischen verschiedenen Objekten oder Hindernissen und dem Sensor in Echtzeit überwacht werden. Wenn eine Person einen Ultraschallsensor passiert, überträgt der Sensor Daten an den Mega2560. Ein Servo ist ein Motor, der Winkel regeln kann. Durch die Verbindung der beiden Servos mit den Pins des Arduino Mega 2560 kann der Öffnungs- und Schließvorgang der Tür durch Ändern des Winkels des Servos gesteuert werden. Durch Programmierung des Drehwinkels des Servos kann die Funktion des automatischen Öffnens und Schließens der Tür realisiert werden. Der Zweck der Verwendung des Servos besteht hier darin, die Anzahl der gleichzeitig ein- und ausgehenden Personen zu begrenzen.

Ein SD-Modul ist ein Gerät zum Lesen und Schreiben von SD-Karten, das über den Anschluss an einen Arduino Mega 2560 kommunizieren kann. Das SD-Modul kann verwendet werden, um Daten einfach auf die TF-Karte zu schreiben, um eine langfristige Speicherung und Sicherung der Daten zu realisieren.

Es werden auch LCD-Module eingesetzt. Seine Hauptfunktion besteht darin, die Anzahl der ein- und ausgehenden Personen anzuzeigen.

Zum Schluss werden noch drei Tasten an die Pins des Arduino Mega 2560 angeschlossen. Mit diesen Tasten kann das Öffnen und Schließen der Tür manuell gesteuert werden. Indem Sie den entsprechenden Code schreiben, können Sie den Zustand der Schaltfläche überwachen und entsprechende Aktionen ausführen, z. B. das Öffnen oder Schließen der Tür und das Nullsetzen der Daten.

Durch die Verbindung dieser Hardwarekomponenten mit dem Arduino Mega 2560 und das Schreiben des entsprechenden Codes kann die Anzahl der Personen angezeigt und gespeichert werden.

## 4.1. Hardware

Dieser Abschnitt befasst sich mit den Hauptkomponenten des Systems und ihren Aufgaben. Der Aufbau des Schaltplans und der Leiterplatte wird ebenfalls in diesem Abschnitt gezeigt.

### 4.1.1. Arduino Mega 2560

Das Arduino Mega 2560 ist ein Mikrocontroller-Board, das auf dem ATmega2560 basiert. Es verfügt über 54 digitale Ein-/Ausgangspins (von denen 15 als PWM-Ausgänge verwendet werden können), 16 analoge Eingänge, 4 UART (serielle Hardware-Ports), einen 16-MHz-Quarzoszillator, einen USB-Anschluss, eine Strombuchse, einen ICSP-Header und eine Reset-Taste.

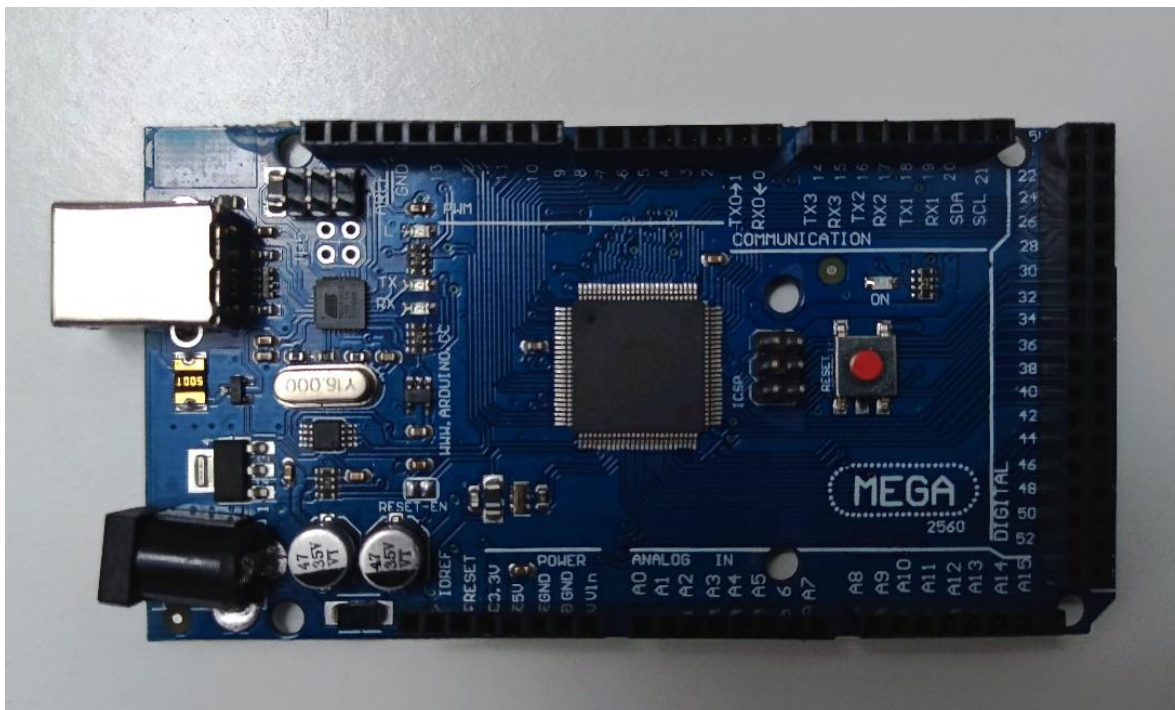


Abbildung 1: Arduino Mega2560

In diesem System wird der Arduino Mega 2560 verwendet, um kompilierte Programme auszuführen und die Aufgaben des Datenempfangs, der Datenverarbeitung und der Eingabe der verarbeiteten Daten in eine Datenbank zu übernehmen.

Als Entwicklungsboard wird der Arduino Mega 2560 aufgrund seiner leistungsstarken Rechenleistung und seiner reichlich vorhandenen Pin-Ressourcen häufig in IoT- und Embedded-Systemen eingesetzt. Durch das Schreiben eines Programms wird es als Gerät konfiguriert, das eine bestimmte Funktion ausführt.

In diesem System wird der Arduino Mega 2560 als Datenverarbeitungseinheit programmiert. Er empfängt Daten, die von Sensoren oder anderen externen Geräten übermittelt werden, und verarbeitet sie. Dies kann das Parsen von Daten, Berechnungen, Filtern oder andere komplexe algorithmische Vorgänge umfassen, um nützliche Informationen zu extrahieren.

Zusammenfassend lässt sich sagen, dass der Arduino Mega 2560 eine wichtige Rolle im System spielt, da er als Kerngerät für die Datenverarbeitung und Eingabedatenbanken dient. Mit seinen leistungsstarken Funktionen und seiner Flexibilität bei der Programmierung ermöglicht es eine effiziente und zuverlässige Datenverarbeitung und unterstützt zuverlässig den Systembetrieb und das Datenmanagement.

#### **4.1.2. HC-SR04**

Ultraschallsensoren spielen eine wichtige Rolle bei der Erkennung der Anwesenheit des menschlichen Körpers in diesem System. Es wird verwendet, um zu sehen, ob jemand den erkannten Bereich durchquert.

Das Herzstück des Ultraschall-Abstandssensors HC-SR04 sind zwei Ultraschallsensoren. Einer wird als Sender verwendet, der das elektrische Signal in einen 40 KHz Ultraschallimpuls umwandelt. Der Empfänger lauscht auf die gesendeten Impulse. Wenn sie empfangen werden, wird ein Ausgangsimpuls erzeugt, dessen Breite verwendet werden kann, um zu bestimmen, wie weit sich der Impuls bewegt.

Die geringe Größe des Sensors erleichtert den Einsatz in jedem Robotikprojekt und bietet eine hervorragende berührungslose Entfernungserfassung zwischen 2 cm und 400 cm (ca. 1 Zoll bis 13 Fuß) mit einer Genauigkeit von 3 mm. Da er mit 5 Volt arbeitet, kann er direkt an den Arduino oder einen anderen 5-V-Logik-Mikrocontroller angeschlossen werden.

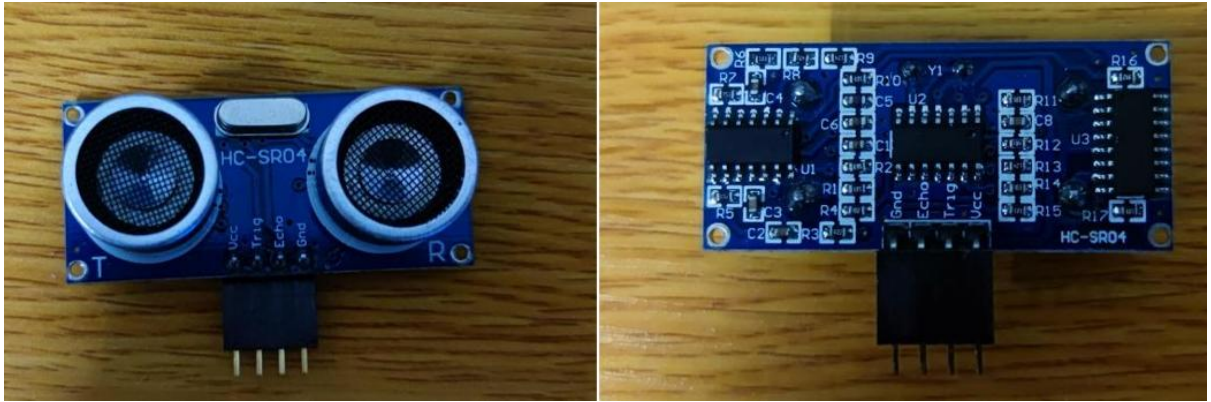


Abbildung 2.

3:HC-SR04

Wenn ein Impuls mit einer Dauer von mindestens  $10\ \mu\text{s}$  (10 Mikrosekunden) an den Triggerpin angelegt wird. Der Sensor sendet acht Schallimpulse mit 40 kHz aus. Acht Ultraschallimpulse bewegen sich durch die Luft, weg vom Sender. Gleichzeitig geht der Echo-Pin hoch und beginnt, den Anfang des Echosignals zu bilden. Wenn diese Impulse nicht zurückreflektiert werden, tritt das Echosignal ab und kehrt nach 38 Millisekunden (38 Millisekunden) auf niedrig zurück. Ein Impuls von 38 ms zeigt also an, dass es innerhalb des Sensorbereichs keine Blockierung gibt.

[2]

Operating Voltage工作电压	直流5V
Operating Current工作电流	15毫安
Operating Frequency运行频率	40K赫兹
Max Range最大范围	4m
Min Range最小范围	2厘米
Ranging Accuracy测距精度	3毫米
Measuring Angle测量角度	15度
Trigger Input Signal触发输入信号	10 $\mu\text{s}$ TTL脉冲
Dimension尺寸	45 x 20 x 15毫米

Abbildung 4: Modellnummer von HC-SR04



### 4.1.3. Micro-SD-Karten-Modul

SD-Karten haben ein breites Anwendungsspektrum in der modernen Technologie und in elektronischen Geräten und bieten komfortable Datenspeicher- und Übertragungslösungen für Benutzer und Geräte. Seine Bedeutung liegt in den Bereichen Datenmanagement, Backup, eingebettete Systeme und mobile Geräte.

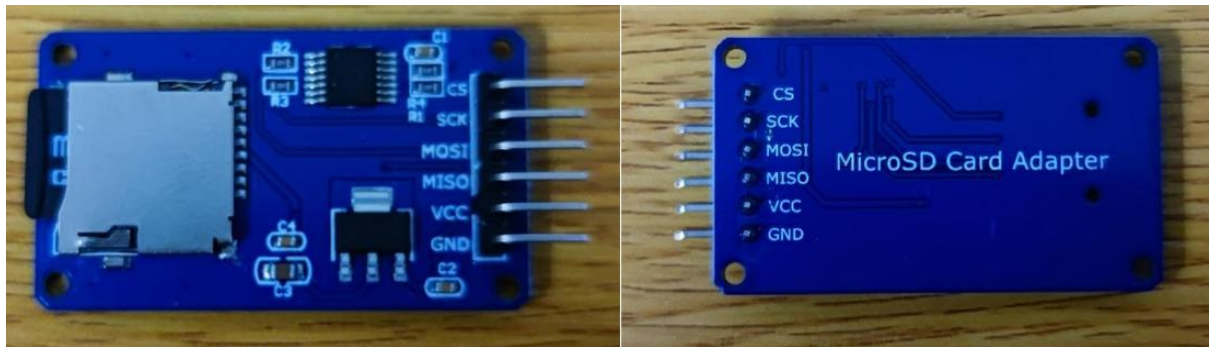


Abbildung 5.

#### 6: Micro-SD-Karten-Modul

Zum Beispiel:

1. **Speicherkapazität:** SD-Karten haben große Speicherkapazitäten, die von wenigen Gigabyte bis zu mehreren Terabyte reichen, was sie ideal für die Speicherung von Daten in tragbaren Geräten und eingebetteten Systemen macht.
2. **Datensicherung:** Mit der SD-Karte können wichtige Daten wie Fotos, Videos, Dokumente und persönliche Dateien usw. gesichert werden. Es bietet eine zuverlässige und tragbare Möglichkeit, wichtige Informationen vor versehentlichem Datenverlust oder Geräteausfall zu schützen.
3. **Datenübertragung:** Die SD-Karte ist ein praktisches Datenübertragungsmedium, mit dem Benutzer einen schnellen und einfachen Datenaustausch erreichen können, indem sie Daten auf der SD-Karte speichern und zwischen verschiedenen Geräten übertragen.
4. **Eingebettetes System:** SD-Karten werden häufig in eingebetteten Systemen wie Einplatinencomputern, industriellen Steuerungsgeräten und Internet-of-Things-Geräten verwendet. Sie bieten eine flexible Speicherlösung, die es Geräten ermöglicht, Daten zu speichern und zu lesen.

Bei diesem Design wird hauptsächlich der Komfort von SD-Kartenmodulen auf Embedded-Systemen genutzt. Es zeichnet die Daten der Anzahl der ein- und ausgehenden Personen auf und kann leicht abgerufen und auf einem PC gelesen und geschrieben werden.

Jede Standard-Micro-SD-Karte arbeitet mit 3,3 V. Daher können wir es nicht direkt an eine Schaltung anschließen, die 5-V-Logik verwendet. Tatsächlich kann jede Spannung über 3,6 V die Micro-SD-Karte dauerhaft beschädigen. Deshalb; Das Modul verfügt über einen integrierten Ultra-Low-Dropout-Regler, der die Spannung von 3,3 V – 6 V auf  $\pm 3,3$  V umwandelt.

Auf dem Modul befindet sich außerdem ein 74LVC125A-Chip, der die Schnittstellenlogik von 3,3 V-5 V in 3,3 V umwandelt. Dies wird als Logikpegelverschiebung bezeichnet. Das bedeutet, dass Sie das Board verwenden können, um mit 3,3-V- und 5-V-Mikrocontrollern wie dem Arduino zu interagieren.

[3]

#### 4.1.4. LCD

Das LCD 1602 ist ein Punktmatrixmodul zur Anzeige von Buchstaben, Zahlen, Zeichen usw. Es besteht aus 5x7 oder 5x11 Punktmatrixpositionen, von denen jede ein Zeichen anzeigen kann. Es gibt einen Punktabstand zwischen zwei Zeichen und einen Abstand zwischen den Zeilen, wodurch die Zeichen von den Zeilen getrennt werden. Modell 1602 bedeutet, dass es 2 Reihen mit 16 Zeichen anzeigen kann.

[4]

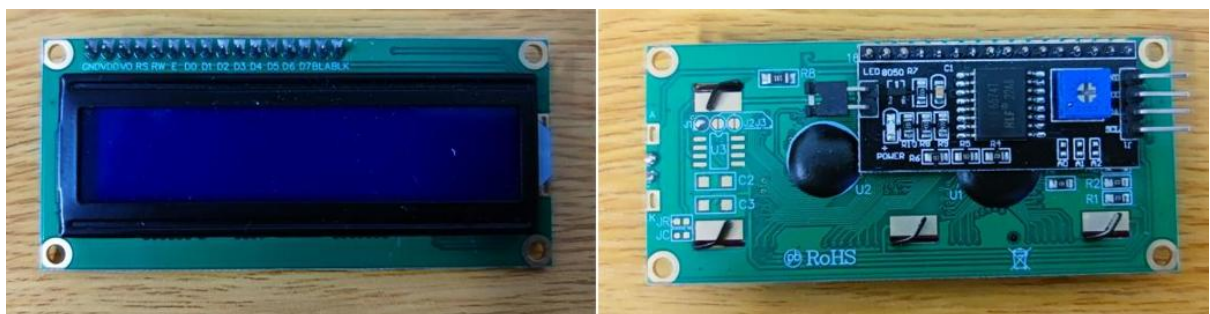


Abbildung 7.

8:LCD

Bei diesem System wird der LCD-Bildschirm hauptsächlich verwendet, um die Anzahl der Personen anzuzeigen, die die Tür betreten und verlassen. Es wird sich mit dem Arduino Mega 2560 auf I2C-Weise verbinden.

#### **4.1.5. Servo**

Das Servo (Servo) des Arduino ist ein häufig verwendetes Ausgabegerät, das zur Steuerung des Winkels und der Position eines mechanischen Systems verwendet wird. Es handelt sich um eine spezielle Art von elektrischem Aktuator, der häufig zur Simulation von Türen, Servomechanismen, Roboterarmen und anderen Anwendungen verwendet wird, die eine präzise Steuerung von Winkel und Position erfordern.

Arduino-Servos bestehen in der Regel aus einem Gleichstrommotor und einem eingebauten Feedback-Mechanismus, um den Winkel durch ein PWM-Signal (Pulsweitenmodulation) zu steuern. Im Inneren des Servos befinden sich ein Potentiometer und ein Komparator, der den aktuell eingestellten Korrekturausgang des Servos überwacht, um den eingestellten Winkel stabil zu halten.

Über den PWM-Ausgangspin des Arduino können wir Steuersignale an das Servo senden. Die Pulsbreite des PWM-Signals bestimmt die Winkelposition des Servos, in der Regel zwischen 1ms und 2ms. Zum Beispiel dreht eine Pulsbreite von 1 ms das Servo in die minimale Winkelposition, während eine Pulsbreite von 2 ms das Servo in die maximale Winkelposition dreht.

Die Servos von Arduino werden häufig in der Robotik, in Drohnen, Kamera-Gimbals, intelligenten Autos und anderen Projekten eingesetzt, und komplexe Bewegungen und Bewegungen können durch präzise Steuerung des Winkels und der Position des Servos erreicht werden.

Bei dieser Konstruktion wird das Servo verwendet, um das Öffnen und Schließen von Türen zu simulieren. Präzises Öffnen und Schließen von Türen erfordert eine präzise Winkelsteuerung.

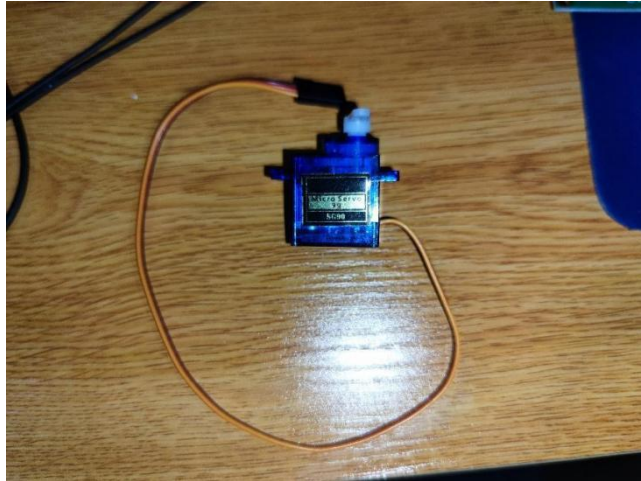


Abbildung 9: Servo

#### 4.1.6. Tasten

Bei dieser Arbeit wurden 3 Schlüssel verwendet. Diese Tasten entsprechen: Türeingangstaste, Ausstiegstaste, Datenlöschstaste. Wenn die Taste gedrückt wird, wird der mit der Taste verbundene Pin geerdet, und der Pegel des Pin-Anschlusses ändert sich von High auf Low.

Masse, wechselt der Pegel des Pin-Ports von high auf low und löst das entsprechende Programm aus.

[6]



Abbildung 10: Tasten

### 4.1.7. Schaltplan

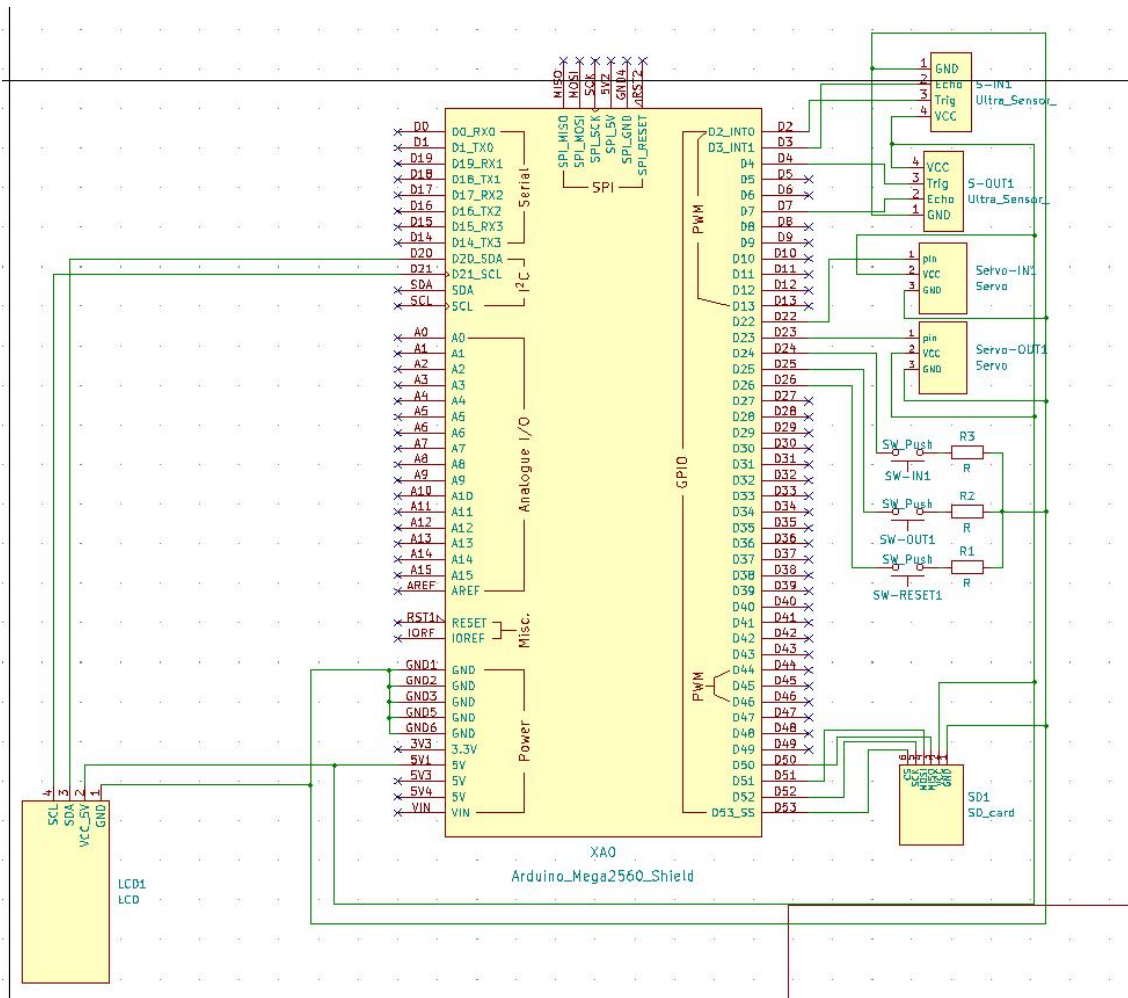


Abbildung 11:Schaltplan

## 4.2. Software

Dieser Abschnitt konzentriert sich auf den Prozess der Programmimplementierung und erörtert die spezifische technische Implementierung der Lösung.

### 4.2.1. Softwarekonzept

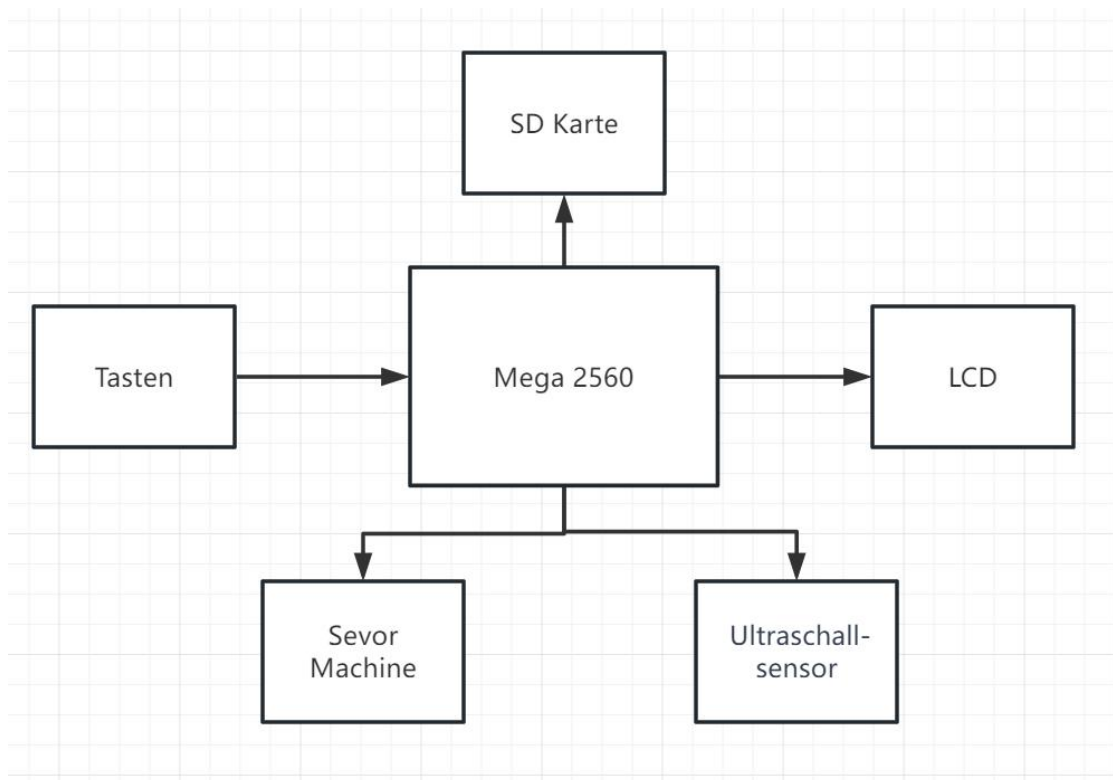


Abbildung 12: Softwarekonzept

Wenn eine externe Steuertaste gedrückt wird, empfängt der Arduino Mega 2560 ein Pegelsignal. Über den Mega 2560 werden das SD-Kartenmodul, das Servomotormodul und das LCD-Modul von ihm gesteuert. Mit dem Servomotormodul wird das Öffnen und Schließen der Tür simuliert, mit dem Sondensensor wird erkannt, ob jemand vorbeigefahren ist, mit dem LCD-Modul wird die spezifische Anzahl der Ein- und Ausgänge angezeigt und mit dem SD-Kartenmodul wird die Anzahl der Ein- und Ausfahrten aufgezeichnet. Gleichzeitig wird innerhalb des Schlüsselmoduls ein Daten-Reset-Taster gesetzt. Mit dieser Schaltfläche kann die Anzahl der aufgezeichneten Ein- und Ausgänge auf den Ausgangszustand zurückgesetzt werden, so dass die Daten nachgezählt und verwaltet werden können. Ein solches Konstruktionsschema kann die automatische Steuerung der Tür und die Überwachung des Ein- und Ausstiegs des Personals effektiv realisieren, die Datenerfassung und -verwaltung erleichtern und die Anforderungen an Echtzeitüberwachung und -statistik erfüllen.

## 4.2.2. Flussdiagramm

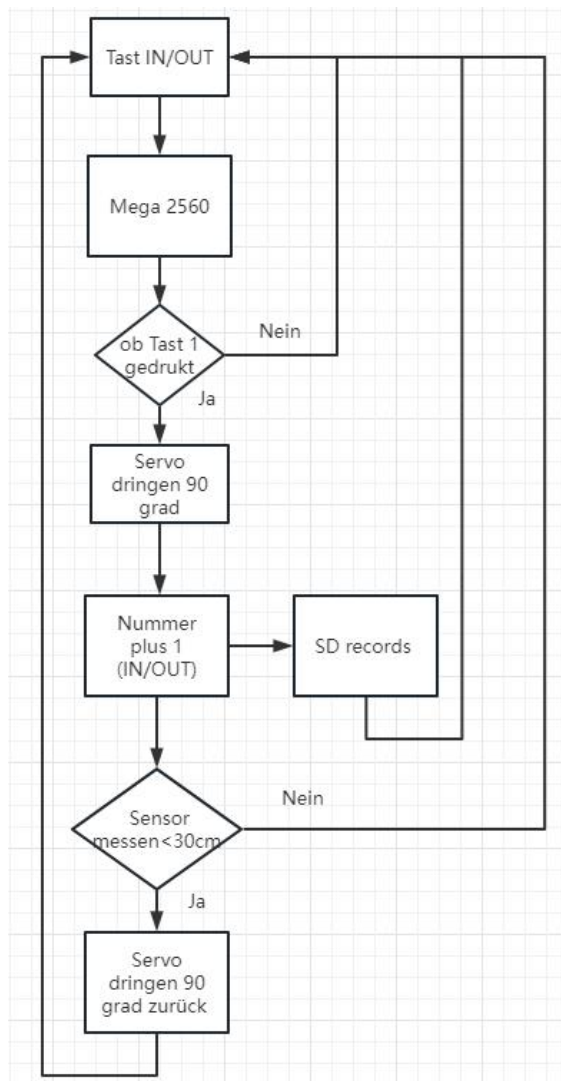


Abbildung 13: Flussdiagramm

Das Ablaufdiagramm des Hauptprogramms ist in der Abbildung dargestellt. Nehmen wir als Beispiel die Türsprechtaste: Wenn die Türsprechtaste gedrückt wird, erhält der Arduino Mega 2560 ein Low-Level-Signal. Nach dem Empfang des Low-Level-Signals steuert der Mega 2560 den Servomotor so, dass er sich um 90° dreht, während die Anzahl der eintretenden Personen um eins erhöht wird und die SD-Karte die relevanten Daten aufzeichnet. An diesem Punkt ist die Tür gleichbedeutend mit dem Öffnen und Personen können hindurchgehen. Ein Ultraschallsensor wird hinter der Tür platziert, und wenn der erkannte Abstand weniger als 30 cm beträgt, wird der Servomotor erneut ausgelöst, wodurch er sich um -90° dreht und das Schließen der Tür simuliert.

Wenn jemand rausgeht, ist der Prozess ähnlich, wie wenn er hereinkommt. Der Arduino Mega 2560 zeigt die Anzahl der Türen über ein LCD an und protokolliert die relevanten Daten auf einer SD-Karte.

Das Design zeigt den Ausführungsprozess beim Betreten und Verlassen der Tür durch ein Flussdiagramm deutlich. Das Zusammenspiel des Arduino Mega 2560 und der einzelnen Module ermöglicht die automatische Steuerung von Türen und die genaue Überwachung des Ein- und Ausstiegs des Personals. Gleichzeitig gewährleistet es durch LCD-Display und SD-Kartenaufzeichnung Echtzeitstatistiken und die Verwaltung der Anzahl der ein- und ausgehenden Personen, was die Datenanalyse erleichtert. Ein solches Prozessdesign weist eine hohe Machbarkeit und Effizienz bei der Implementierung und Anwendung von Anschnittsystemen auf.

### **4.2.3. Bibliotheksdateien**

Bibliotheksdateien spielen in Softwareprogrammen eine äußerst wichtige Rolle. Die Hauptfunktion von Bibliotheksdateien besteht darin, gekapselte Funktionen und Datenstrukturen bereitzustellen, die Programmierern die Ausführung einiger allgemeiner Aufgaben erleichtern sollen. Diese Funktionen und Datenstrukturen werden vollständig in Bibliotheksdateien getestet und optimiert, um ihre Stabilität und Zuverlässigkeit im Programm zu gewährleisten. Durch das Aufrufen dieser Funktionen und Datenstrukturen im Programm können Entwickler die von Bibliotheksdateien bereitgestellten Funktionen nutzen, wodurch der Programmierprozess vereinfacht, die Wiederverwendung von Code verbessert und die Entwicklung beschleunigt wird.

Eine Bibliotheksdatei ist eine wiederverwendbare Softwareressource, die in der Regel bereits gekapselte Funktionen, Methoden und Datenstrukturen enthält. Diese Funktionen und Datenstrukturen sind so konzipiert, dass sie bestimmte Probleme lösen oder allgemeine Funktionen wie Dateibearbeitung, Datenverarbeitung, Grafikzeichnung usw. implementieren. Durch die Kapselung dieser Funktionen in einer Bibliotheksdatei müssen Programmierer nicht immer wieder denselben Code schreiben, sondern können Funktionen in der Bibliothek direkt aufrufen, um bestimmte Aufgaben auszuführen.

Bibliotheksdateien vereinfachen nicht nur den Programmierprozess, sondern können auch die Wartbarkeit und Lesbarkeit Ihres Codes verbessern. Durch die Verwendung der Funktionen und Datenstrukturen in Bibliotheksdateien können Programmierer den Code modularer und übersichtlicher organisieren, was das Verständnis und die Wartung erleichtert.



Gleichzeitig werden die Funktionen in der Bibliotheksdatei getestet und optimiert, was dazu beiträgt, Fehler und Schwachstellen im Programm zu reduzieren und die Stabilität und Zuverlässigkeit des Programms zu verbessern.

#### 4.2.4. Servo-Programm

```
if (buttonState == 0)
{
  servoblau.write(90);
  presstime++;
}
if (dist1 < 30) {
  servoblau.write(-90);
}
```

Abbildung 14: Servo-Programm

In der Abbildung erfolgt die Steuerung des Servos durch die Verwendung der Funktion `if()`, um den Zustand des Tasters zu bestimmen. `servoblau.write()` ist eine Befehlsanweisung in einer Bibliotheksdatei, in der Sie die Winkeldrehung des Servos leicht erreichen können, indem Sie die Anzahl der Winkel in Klammern eingeben.

Die Funktion `if()` ist eine bedingte Anweisung in der Programmierung, die verwendet wird, um verschiedene Codeblöcke basierend auf bestimmten Bedingungen auszuführen. Hier verwenden wir die Funktion `if()`, um den Zustand der Taste zu erkennen, d.h. ob sie gedrückt wird oder nicht. Wenn ein Tastendruck erkannt wird, wird der entsprechende Codeblock ausgelöst, der das Servo steuert.

`servoblau.write()` ist eine Methode, die in einer Servobibliothek bereitgestellt wird und einen Parameter akzeptiert, den einzustellenden Servowinkel. Durch das Schreiben des entsprechenden Winkelwertes in Klammern kann die Winkeldrehung des Servos einfach realisiert werden. Zum Beispiel bedeutet `servoblau.write(90)` das Drehen des Servos in eine Position von 90 Grad, während `servoblau.write(0)` das Drehen des Servos in eine Position von 0 Grad bedeutet.

Diese Methode bietet Flexibilität und Komfort für die Steuerung des Servos, das die Winkelposition des Servos je nach Bedarf präzise steuern kann.

Zusammenfassend lässt sich sagen, dass durch die Kombination der Funktion `if()` und der Befehlsanweisung `servoblau.write()` eine flexible Steuerung des Servos einfach erreicht werden kann, was eine effektive Lösung für die erfolgreiche Umsetzung des Projekts und die präzise Bewegung des mechanischen Systems darstellt.

#### 4.2.5. SD-Karten-Modul-Programm

```
void CARD() {  
  Serial.print("IN:");  
  Serial.print(presstime);  
  Serial.print("  OUT:");  
  Serial.print(presstime2);  
  Serial.println();  
  
  myFile = SD.open("test.txt", FILE_WRITE);  
  if (myFile) {  
    myFile.print("IN:");  
    myFile.print(presstime);  
    myFile.print("  OUT:");  
    myFile.print(presstime2);  
    myFile.println();  
    myFile.close(); // close the file  
  }  
}
```

Abbildung 15:SD-Karten-Modul-Programm

Gemäß dem in der obigen Abbildung gezeigten Programmablauf erstellt das System eine neue Textdatei mit dem Namen "test.txt" auf der SD-Karte und zeichnet darin Daten wie "IN:1 OUT:1" auf, wobei jeder Datensatz eine separate Zeile belegt.

Um den Speicherplatz der SD-Karte effektiv zu sparen, verfolgt das System eine intelligente Designstrategie, d. h. die SD-Karte liest und schreibt Daten nur, wenn die Taste gedrückt wird. Diese Strategie stellt sicher, dass Daten nur dann gespeichert werden, wenn wirklich Daten aufgezeichnet werden müssen, wodurch unnötige Datenredundanz und Speicherverschwendung vermieden werden.

Durch solche Sparmaßnahmen kann das System die Speicherplatzeffizienz der SD-Karte optimieren und die Verschwendung von SD-Kartenkapazität durch häufiges Datenloggen vermeiden. Diese intelligente Lese-/Schreibstrategie sorgt für eine effiziente Aufzeichnung und Speicherung von Daten und reduziert gleichzeitig die Lese- und Schreiblast der SD-Karte, was dazu beiträgt, die Lebensdauer der SD-Karte zu verlängern.

Durch die Kombination des Datenaufzeichnungsbetriebs mit der Tastenauslösung erreicht das System daher ein ausgewogenes Verhältnis zwischen SD-Kartenspeicher und Datenmanagement, was nicht nur die Genauigkeit und Integrität der Datenaufzeichnung gewährleistet, sondern auch die Speicherressourcen der SD-Karte voll ausnutzt, was eine effektive Lösung für den langfristigen Betrieb des Systems und die Datenverwaltung darstellt.

#### 4.2.6. Probleme

Beim Abschließen des Programms und der Durchführung des Breadboard-Tests trat das folgende Problem auf: In dem Moment, in dem der Knopf gedrückt wurde, kam es zu einem unkontrollierten zufälligen Anstieg der Anzahl der Personen, die die Tür betreten und verlassen. Nach sorgfältiger Analyse liegt die Ursache dieses Problems in der Verarbeitungsgeschwindigkeit des Prozessors und dem Tasten-Jitter.

Beim Tastenzittern wird beim Drücken oder Loslassen einer Taste der Kontaktpunkt der Taste in kurzer Zeit mehrmals ein- und ausgeschaltet, was zu kleinen elektrischen Signalschwankungen führt. Dieses Phänomen tritt in realen Schaltungen sehr häufig auf und kann zu einer Fehlauslösung oder Mehrfachauslösung führen. Der Prozessor ist sehr schnell und kann eine Änderung des Zustands der Taste schnell erkennen, aber aufgrund des Einflusses von Tastenzittern kann es sein, dass sie fälschlicherweise als mehrfaches Drücken erkannt wird, was zu einem zufälligen Anstieg der Anzahl der Personen führt.

Um dieses Problem zu lösen, haben wir versucht, dem Programm die Funktion `delay()` hinzuzufügen. Die Funktion `delay()` kann nach dem Drücken einer Taste eine kurze Verzögerung verursachen, wodurch verhindert wird, dass der Prozessor zu schnell auf Änderungen des Tastenzustands reagiert. Mit der richtigen Verzögerung haben wir die Auswirkungen des Tastenwackelns erfolgreich unterdrückt und sichergestellt, dass die Zählung nur einmal gezählt wurde, wenn die Taste gedrückt wurde, wodurch das Problem des zufälligen Anstiegs der Anzahl der gezählten Personen gelöst wurde.

Diese Methode löst die Probleme, die durch Tastenzittern verursacht werden, bis zu einem gewissen Grad, muss aber auch debuggt und entsprechend der spezifischen Situation optimiert werden. Wir müssen ein Gleichgewicht zwischen Latenz und Benutzerfreundlichkeit herstellen, um sicherzustellen, dass die Reaktionsgeschwindigkeit des Benutzers bei der

Verarbeitung von Schlüssel-Jitter nicht wesentlich beeinträchtigt wird. Durch sinnvolles Debuggen und Optimieren wird schließlich eine effektivere Lösung realisiert, die die Genauigkeit und Stabilität der Anzahl der gezählten Personen gewährleistet.

```
if (buttonState2 == 0)
{
  servoblau2.write(90);
  presstime2++;
  delay(500);
}
```

Abbildung 16:Probleme

## 5. Technischer Aufbau

In dieser Studie haben wir eine Leiterplatte genommen und die elektronischen Komponenten im Kapitel Elektronische Hardware darauf gelötet. Wir aktivieren dann die Funktionen dieses Boards, indem wir es über Pins direkt in den Arduino Mega 2560 stecken.

Als grundlegende Komponenten, die für integrierte Schaltkreise erforderlich sind, kann diese Leiterplatte elektronische Komponenten auf Dünnschicht drucken, wodurch die Leiterplatte kompakter, zuverlässiger und einfacher herzustellen ist.

Im Vergleich zu handgelöteten Leiterplatten sind Leiterplatten einfacher herzustellen, da Schaltungskomponenten automatisch bestückt werden können, was die Produktionseffizienz verbessert. Gleichzeitig werden Leiterplatten mit höherer Präzision und Genauigkeit hergestellt als handgelötete Leiterplatten, wodurch ihre Zuverlässigkeit und Stabilität verbessert wird. Darüber hinaus kann das Design der Leiterplatte in einer Computerdatei gespeichert werden, um es bei Bedarf zuverlässig zu kopieren und zu ändern.

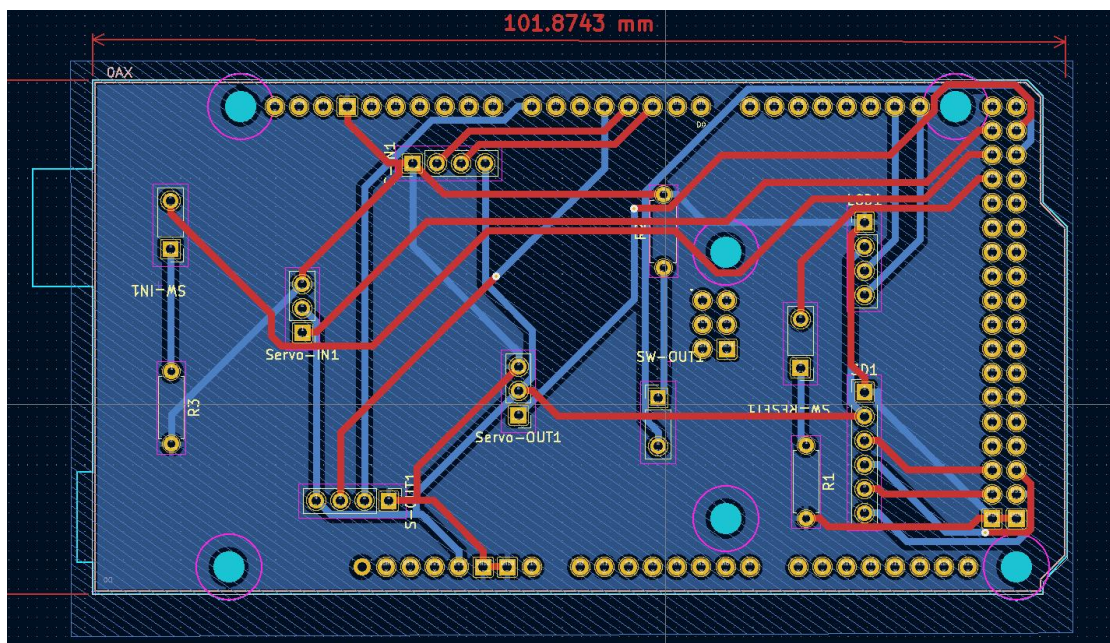


Abbildung 17:PCB-Leiterplatte

Diese PCB-Anwendung bietet eine effektive Hardware-Unterstützung für unsere Forschung, die es uns ermöglicht, die Funktionen des Arduino Mega 2560 effizienter zu implementieren. Durch das Design und die Herstellung von Leiterplatten konnten wir die Leistung und Stabilität elektronischer Systeme verbessern und die Entwicklung und Anwendung dieser

Forschung weiter vorantreiben. Generell bietet der Einsatz von Leiterplatten eine solide Grundlage für unsere Forschungsarbeit und gewährleistet den zuverlässigen Betrieb des Systems auf Hardware-Ebene.

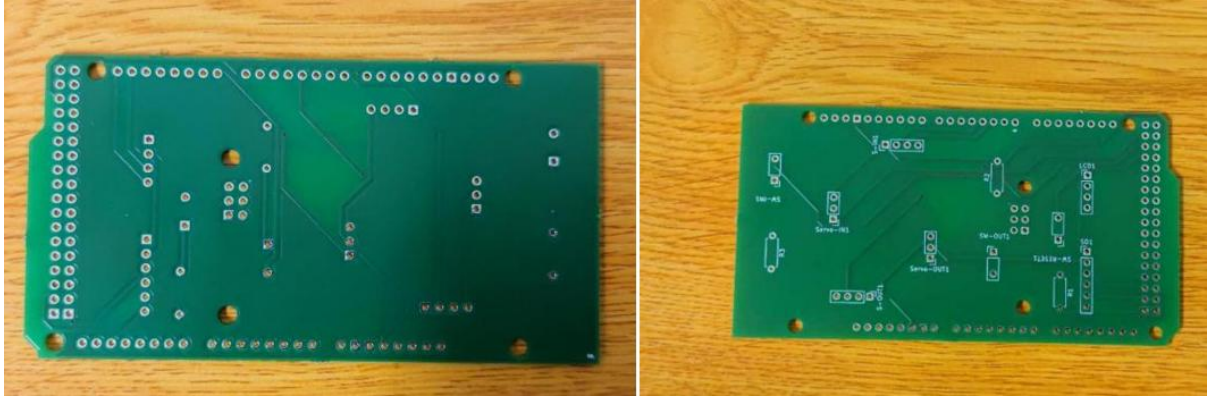


Abbildung 18: Gelötete PCB-Leiterplatte

## 6. Experimenteller Teil

### 6.1 Simulations-Experimente

#### 6.1.1 Breadboard-Simulation

Wenn jemand durch die Tür geht, drücken Sie den Knopf, und nachdem die Person vorbeigegangen ist, dreht sich das Servo um 90 Grad. Wenn eine Person 30 cm vorwärts geht, erkennt der Ultraschallsensor, dass jemand vorbeigeht. Das Servo dreht sich um 90 Grad zurück und kehrt in seinen ursprünglichen Zustand zurück

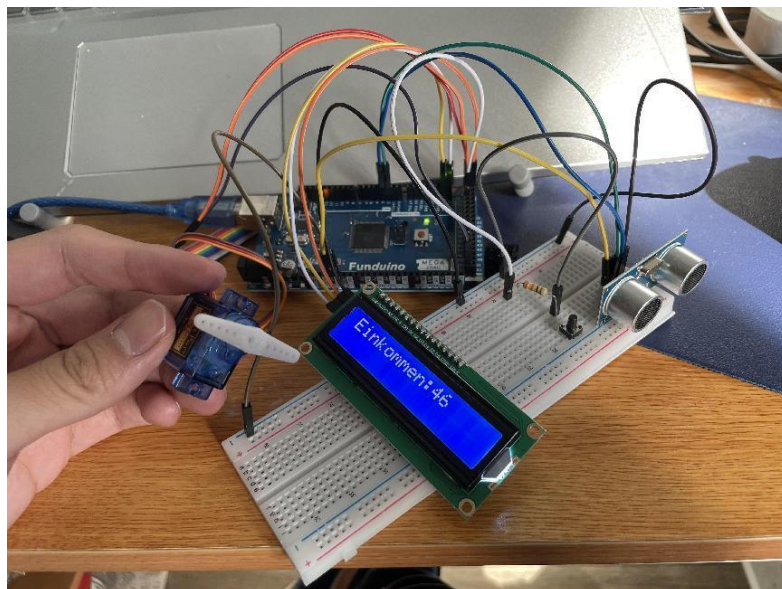


Abbildung 19: Breadboard-Simulation

Das LCD erhöht oder verringert die Anzahl der Personen durch Drücken der Taste. Die Daten werden dann auf die SD-Karte übertragen und gespeichert

#### 6.1.2 Breadboard-Simulation

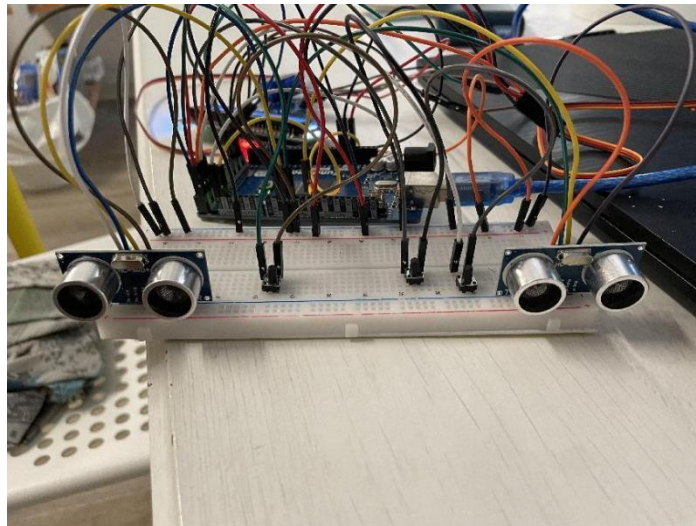


Abbildung 20: Breadboard-Simulation

In meinem System werden zwei Türen verwendet, daher ist es natürlich, zwei Sätze identischer Komponenten zu haben. Zwei Servos, zwei Ultraschallsensoren, zwei Tasten. Ich habe auch eine dritte Taste, um die Daten zurückzusetzen

## 6.2 Experimentelles Flussdiagramm

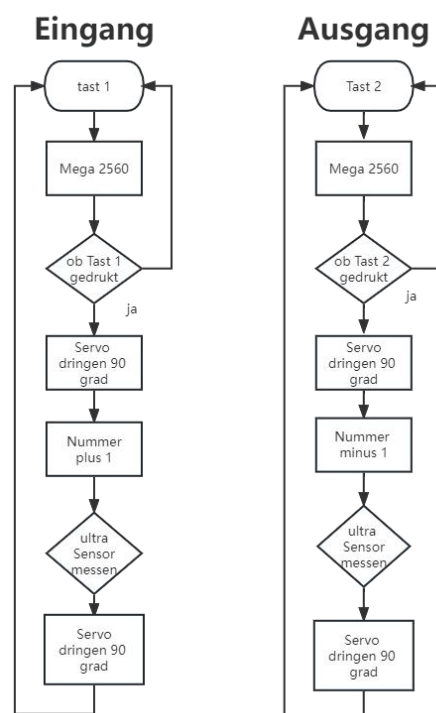


Abbildung 21: Experimentelles Flussdiagramm



Das Verfahren für die Ein- und Ausfahrt ist fast das gleiche, nur die Zählung ist unterschiedlich

### 6.3 Beispiel-Referenz



Abbildung 22: Beispiel-Referenz

Dies ähnelt meinem idealen Türmodell, und wenn es in der Praxis angewendet wird, kann dieses Beispiel eine wichtige Referenz sein

### 6.4 Simulationslauf

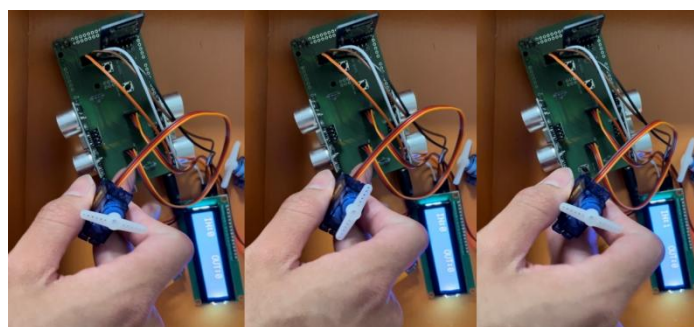


Abbildung 23: Praktische Erfahrung

Wie im Anweisungsdiagramm gezeigt. Wenn eine Person den Eingang passiert, drücken Sie die Taste, der Servo dreht sich um 90 Grad, und das LCD-Display zeigt die Anzahl der Teilnehmer plus eins an. Wenn eine Person den Ultraschallsensor passiert, dreht sich der Servo wieder um 90 Grad und die Tür kehrt in ihren ursprünglichen Zustand zurück.



Abbildung 24:Praktische Erfahrung

Ähnlich verhält es sich, wenn jemand am Ausgang vorbeigeht und die Taste drückt. Der Servo dreht sich um 90 Grad, und das LCD-Display zeigt die Anzahl der Personen, die den Ausgang verlassen, plus eins an. Wenn eine Person den Ultraschallsensor passiert, dreht sich der Servo wieder um 90 Grad und die Tür kehrt in ihren ursprünglichen Zustand zurück.

Das System verfügt auch über eine Taste zum Zurücksetzen der Daten.

## 7. Zusammenfassung

Der Entwicklungsprozess dieses Projekts durchläuft vier Hauptphasen: Analyseanforderungen, Hardware-Schaltungsentwurf, Softwareprogrammierung und -optimierung sowie Gesamtsystemtest. Bei der Implementierung gab es eine Reihe von Schwierigkeiten, von denen eine die Genauigkeit der Schlüsselzählung war.

Um das Problem der Genauigkeit der Schlüsselzählung zu lösen, wurden umfangreiche Recherchen und Datenüberprüfungen durchgeführt, einschließlich relevanter Materialien im Internet und eingehender Analysen, die von uns selbst durchgeführt wurden. Durch die Recherche bestehender Technologien und Lösungen wurden wertvolle Erfahrungen und Erkenntnisse gewonnen, die eine wichtige Referenz für die Problemlösung darstellen.

Nach vielen Experimenten und Optimierungen konnte das Problem der Schlüsselzählung erfolgreich gelöst und der stabile Betrieb des Systems realisiert werden. Dieses Ergebnis ist das Ergebnis vielfältiger Anstrengungen und umfassender Überlegungen, die es in verschiedenen Phasen des Projekts ermöglicht haben, Herausforderungen zu meistern und die Funktionalität und Leistung des Systems kontinuierlich zu verbessern und zu optimieren.

Generell konnten durch den systematischen Entwicklungsprozess und das Sammeln von Erfahrungen in der Problemlösung die Projektziele erfolgreich erreicht und ein relativ stabiler Betriebszustand erreicht werden. Bei der Umsetzung des Projekts wurde das Design der Hardwareschaltung und der Softwareprogrammierung berücksichtigt und angepasst, um sicherzustellen, dass das System die erwartete Leistung und Stabilität erreicht. Diese Errungenschaft liefert auch nützliche Lehren für die Entwicklung ähnlicher Projekte in der Zukunft.

## 8. Literaturverzeichnis

- [1] 05.07.2023. Online. Available:  
<https://docs.arduino.cc/hardware/mega-2560?queryID=undefined>
- [2] 05.07.2023. Online. Available:  
<https://zhuanlan.zhihu.com/p/339855039>
- [3] 05.07.2023. Online. Available:  
Micro <https://zhuanlan.zhihu.com/p/349832403>SD 卡模块与 Arduino 的接口 - 知乎 (zhihu.com)
- [4] 05.07.2023. Online. Available:  
[http://wiki.sunfounder.cc/index.php?title=LCD1602\\_Module](http://wiki.sunfounder.cc/index.php?title=LCD1602_Module)
- [5] 06.07.2023. Online. Available:  
**[Servo - 维基百科，自由的百科全书 \(wikipedia.org\)](#)**
- [6] 06.07.2023. Online. Available:  
<https://de.wikipedia.org/wiki/Tastaturbelegung>

## 9. Vollständiger Software-Code

```
#include <Servo.h>
// servo
Servo servoblau;
Servo servoblau2;

#include <SD.h>
#include <SPI.h>
File myFile;
int pinCS = 53; // Pin 10 on Arduino Uno

#include <MsTimer2.h>

#include <LiquidCrystal_I2C.h>
//lcd
LiquidCrystal_I2C lcd(0x27, 20, 4);

int pushButton = 24; //定义按键为管脚 24 IN
int buttonState = 0; //按键初始值为 0
int presstime = 0; //记录按了几次
int pushButton2 = 25; //定义按键为管脚 25 OUT
int buttonState2 = 0; //按键初始值为 0
int presstime2 = 0; //记录按了几次
int pushButton3 = 26; //定义按键为管脚 26 RESET
int buttonState3 = 0; //按键初始值为 0

//Sensor
volatile float dist1;
volatile float dist2;

//SD
void CARD() {
  Serial.print("IN:");
  Serial.print(presstime);
  Serial.print(" OUT:");
  Serial.print(presstime2);
  Serial.println();

  myFile = SD.open("test.txt", FILE_WRITE);
  if (myFile) {
    myFile.print("IN:");
    myFile.print(presstime);
    myFile.print(" OUT:");
    myFile.print(presstime2);
```

```
    myFile.println();
    myFile.close(); // close the file
}
// if the file didn't open, print an error:
else {
    Serial.println("error opening test.txt");
}
}
// TRIG 4 ECHO 7
float checkdistance_4_7() {
    digitalWrite(4, LOW);
    delayMicroseconds(2);
    digitalWrite(4, HIGH);
    delayMicroseconds(10);
    digitalWrite(4, LOW);
    float distance1 = pulseIn(7, HIGH) / 58.00;
    delay(10);
    return distance1;
}
//TRIG 2 ECHO 3
float checkdistance_2_3() {
    digitalWrite(2, LOW);
    delayMicroseconds(2);
    digitalWrite(2, HIGH);
    delayMicroseconds(10);
    digitalWrite(2, LOW);
    float distance2 = pulseIn(3, HIGH) / 58.00;
    delay(10);
    return distance2;
}

void setup() {
    Serial.begin(9600);
    //lcd
    lcd.init();
    lcd.backlight();

    //servo
    servoblau.attach(22); //入口舵机 pin22
    servoblau2.attach(23); //出口舵机 pin23

    //tasten
    pinMode(pushButton, INPUT_PULLUP); //定义管脚 24 为输入管脚 IN
    pinMode(pushButton2, INPUT_PULLUP); //定义管脚 25 为输入管脚 OUT
    pinMode(pushButton3, INPUT_PULLUP); //定义管脚 26 为输入管脚 RESET
```

```
//sensor
dist1 = 0; //入口 cm
dist2 = 0; //出口 cm
pinMode(4, OUTPUT);
pinMode(7, INPUT);
pinMode(2, OUTPUT);
pinMode(3, INPUT);

//SD
pinMode(pinCS, OUTPUT);
// SD Card Initialization
if (SD.begin()) {
    Serial.println("SD card is ready to use.");
} else {
    Serial.println("SD card initialization failed");
    return;
}
}

void loop() {

//LCD
lcd.setCursor(0, 0);
lcd.print("IN:");
lcd.print(presstime);
lcd.setCursor(8, 0);
lcd.print("OUT:");
lcd.print(presstime2);

//sensor
dist1 = checkdistance_4_7(); //T4 E7
dist2 = checkdistance_2_3(); //TRIG 2 ECHO 3

//tasten
buttonState = digitalRead(pushButton); //读取按键状态 in
buttonState2 = digitalRead(pushButton2); //读取按键状态 out
buttonState3 = digitalRead(pushButton3); //读取按键状态 reste

//IN
if (buttonState == 0) //如果按键按下并且上一次按下松开
{
    servoblau.write(90);
    presstime++; //打印出按了几次
    delay(500);
}
if (dist1 < 30) {
    servoblau.write(-90);
}
```

```
//OUT
if (buttonState2 == 0) //如果按键按下并且上一次按下松开
{
    servoblau2.write(90);
    presstime2++;
    delay(500);
}
if (dist2 < 30) {
    servoblau2.write(-90);
}

//RESET
if (buttonState3 == 0) {
    presstime = 0;
    presstime2 = 0;
    delay(20);
} //如果按键按下并且上一次按下松开

if (buttonState == 0 or buttonState2 == 0 or buttonState3 == 0) {
    CARD();
}
}
```



## 10. 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 29 .Juli.2023

Ziang He