

Beleg

Embedded Systems I



WS 24/25

Intelligentes Türsteuerungssystem mit RFID und Keypad („Smart Door“)

ausgearbeitet von:

Tawfik Eddihi

Salah Eddine Mashat

Matrikelnummer:

82588

82705

Seminargruppe:

22-EIB-IAS

Studiengang:

Elektrotechnik und Informationstechnik

Studienprofil:

Informationstechnik/Automatisierungssysteme

Betreuer:

Professor *Andreas Pretschner*

Abgabetermin:

17.03.2025

Inhaltsverzeichnis

1 Einleitung	3
1.1 Motivation und Zielsetzung	3
1.2 Funktionsweise des entwickelten Systems	4
2 Planungsphase	5
2.1 Projektplan und Zeitmanagement.....	5
2.2 Aufgabenübersicht	5
3 Anforderungsanalyse und verwendete Komponenten	6
3.1 Anwendungsbereiche	6
3.2 Übersicht und Beschreibung der Hardwarekomponenten	6
3.3 Verwendete Softwarekomponenten	7
4 Systemarchitektur und Design	8
4.1 Aufbau und Struktur	8
4.2 Blockdiagramm des Systems	8
5 Hardwareimplementierung	9
5.1 Verwendete Hardware und Pin-Zuweisung.....	9
5.2 Schaltplan des Türsteuerungssystems.....	10
5.3 Praktischer Hardwareaufbau	10
6 Softwareentwicklung und Implementierung	11
6.1 Entwicklungsumgebung (PlatformIO).....	11
6.2 Konfiguration der PlatformIO-Umgebung.....	11
6.3 Pin-Zuweisung und Initialisierung der Komponenten	11-12
7 Hauptfunktionen des Systems	13
7.1 Initialisierung des Systems	13
7.2 RFID-Kartenprüfung und -verifizierung	13
7.3 PIN-Code-Verifizierung	14
7.4 Türsteuerung	14
7.5 Loop-Funktion – Kontinuierliche RFID- und PIN-Überprüfung	15
8 Tests und Validierung	16
8.1 Serielle Monitor-Ausgabe zur Echtzeitüberwachung	16
8.2 Video-Demonstration der Systemfunktionalität.....	16
8.3 Fehleranalyse und Lösungsansätze	17
8.4 Ergebnisse und Diskussion	18
9 Fazit.....	18

1. Einleitung

1.1 Motivation und Zielsetzung

- **Zielsetzung:**

Das Ziel dieses Projekts ist die Entwicklung eines sicheren und effizienten Zugangskontrollsystems, das sowohl RFID-Technologie als auch eine PIN-basierte Authentifizierung kombiniert. Das System ermöglicht autorisierten Personen den Zutritt durch das Scannen einer registrierten RFID-Karte oder die Eingabe eines gültigen PIN-Codes.

Dieses Zugangskontrollsystem eignet sich für verschiedene Anwendungsbereiche, darunter Büros, Forschungs- und Entwicklungszentren, industrielle Anlagen sowie private Wohnräume, in denen eine erhöhte Sicherheit erforderlich ist. Durch die doppelte Authentifizierungsmöglichkeit bietet das System nicht nur Flexibilität, sondern erhöht auch den Schutz vor unbefugtem Zugriff. Darüber hinaus ermöglicht die Implementierung eines LCD-Displays eine intuitive Benutzerführung und visuelles Feedback.

Ein weiteres Ziel des Projekts ist die modulare und skalierbare Entwicklung, sodass das System problemlos an unterschiedliche Anforderungen angepasst oder um weitere Sicherheitsfunktionen erweitert werden kann.

- **Themenüberblick und Motivation**

In der heutigen Zeit sind eingebettete Systeme (Embedded Systems) essenziell für moderne Automatisierungs- und Sicherheitstechnologien. Sie ermöglichen eine präzise Erfassung, Verarbeitung und Reaktion auf verschiedene Eingaben in Echtzeit. In diesem Projekt wurde ein intelligentes Türsteuerungssystem entwickelt, das auf einem Arduino Mega 2560 basiert und durch RFID-Authentifizierung sowie PIN-Code-Eingabe eine zuverlässige Zugangskontrolle gewährleistet.

Das Hauptaugenmerk lag auf der Entwicklung eines autarken und reaktionsschnellen Systems, das ohne externe Netzwerkverbindungen oder Cloud-Dienste auskommt. Dies gewährleistet nicht nur eine unabhängige Funktionalität, sondern reduziert auch potenzielle Sicherheitsrisiken durch externe Angriffe.

Technisch betrachtet setzt das System auf eine Kombination mehrerer Module, die über unterschiedliche Kommunikationsprotokolle arbeiten:

- RFID-Modul (MFRC522) über das SPI-Protokoll zur Identifikation von autorisierten Benutzern
- LCD-Display (16x2) für die visuelle Ausgabe von Systemzuständen und Benutzerhinweisen
- Keypad (4x4-Matrix) zur numerischen PIN-Eingabe als alternative Authentifizierung
- Servo-Motor zur Steuerung des Türschlosses
- LEDs & Buzzer als optische und akustische Signale für Benutzerinteraktionen

Die Implementierung dieses Systems erforderte eine präzise Synchronisation der Komponenten, um eine zuverlässige Benutzererfahrung und Systemstabilität zu gewährleisten. Eine der größten Herausforderungen war die fehlerfreie Verarbeitung paralleler Eingaben sowie die korrekte Synchronisation der RFID- und Keypad-Authentifizierung, um eine nahtlose Benutzerführung zu ermöglichen.

Durch die erfolgreiche Umsetzung dieses Projekts konnten tiefgehende Kenntnisse in Mikrocontroller-Programmierung, Sensortechnik, Hardware-Schnittstellen und eingebetteten Systemen erlangt werden. Insbesondere das Zusammenspiel verschiedener Kommunikationsprotokolle (SPI, I²C, digitale I/O) sowie die Optimierung der Reaktionszeiten und Energieeffizienz stellten einen bedeutenden Lernprozess dar.

1.2 Funktionsweise des entwickelten Systems

Das System überprüft die Identität des Benutzers entweder durch Scannen einer RFID-Karte oder durch die Eingabe eines PIN-Codes. Die eingegebenen Daten werden mit einer gespeicherten Autorisierungsliste verglichen. Falls eine Übereinstimmung vorliegt, öffnet sich das Türschloss durch einen Servo-Motor. Andernfalls wird der Zutritt verweigert und eine Warnanzeige über LED und Buzzer ausgegeben.

Der nachfolgende Prozessablauf veranschaulicht die Funktionsweise des Systems detailliert:

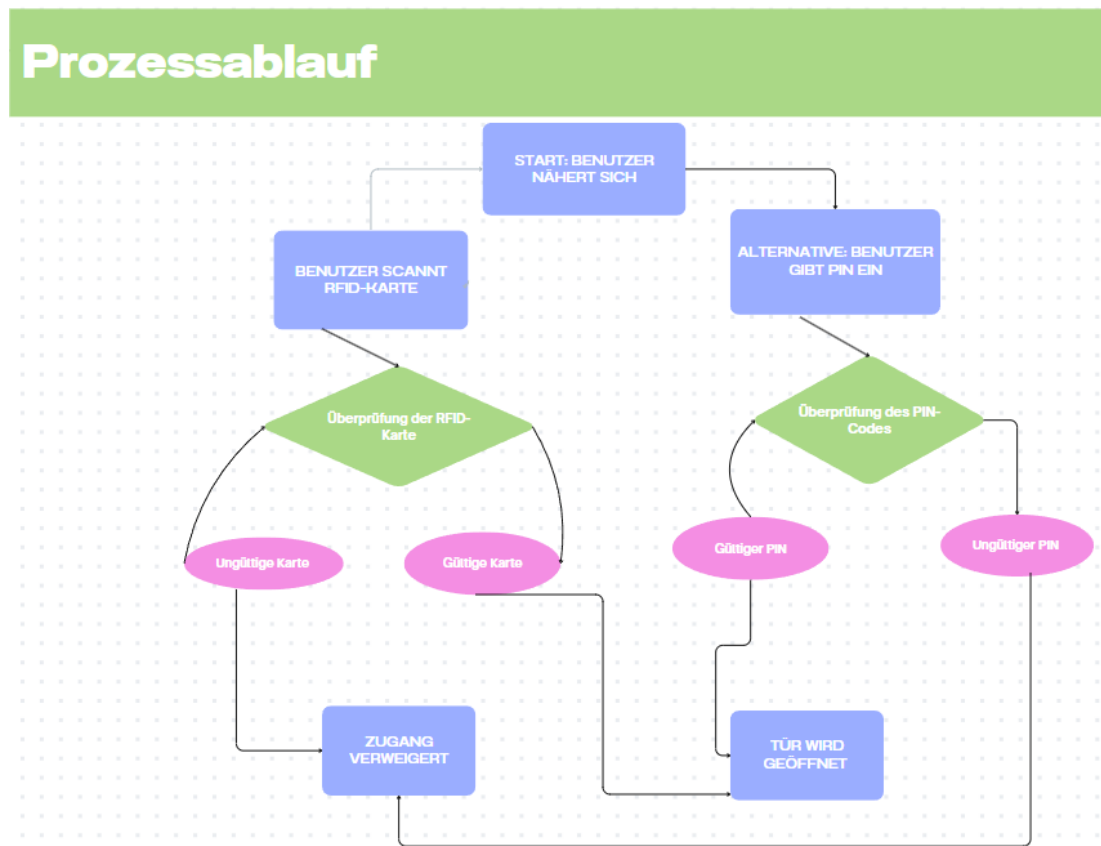


Abbildung 1: Prozessablauf der Zutrittskontrolle

2. Planungsphase

2.1 Projektplan und Zeitmanagement

- **Projektzeitplan:**

Dieser Abschnitt beschreibt die Phasen der Entwicklung, von der Anforderungsanalyse bis zur Integration und Dokumentation.

Phase	Beschreibung	Dauer
Anforderungsanalyse	Festlegung der Anforderungen für RFID- & Keypad-Systeme	1 Woche
Hardware-Planung	Auswahl der Bauteile (RFID-Reader, Keypad, LCD, LEDs, Buzzer, Arduino Mega 2560, Servo)	1 Woche
Schaltplan-Design	Erstellen eines Schaltplans mit Fritzing oder KiCad	1 Woche
Software-Entwicklung	Implementierung des Codes zur Steuerung der Tür mit RFID & PIN-Eingabe	2 Wochen
Integration & Tests	Hardware- und Software-Integration, Fehlerbehebung & Optimierung	1 Woche
Dokumentation	Erstellung einer technischen Dokumentation	1 Woche

2.2 Aufgabenübersicht

In diesem Abschnitt werden die Hauptaufgaben innerhalb des Projekts detailliert beschrieben.

Aufgabe	Beschreibung
Schaltplan erstellen	Zeichnen des elektronischen Aufbaus mit Verbindungen zwischen den Bauteilen
Hardware-Komponenten vorbereiten	Anschließen von RFID-Modul, Keypad, LCD, Servo-Motor, LEDs und Buzzer an den Arduino Mega 2560
Entwicklung des Codes	Implementierung der Logik zur RFID- und PIN-Überprüfung sowie zur Steuerung des Türöffnungsmechanismus
Testing & Debugging	Durchführung von Tests zur Überprüfung der korrekten Funktionsweise aller Module
Optimierung & Fehlerbehebung	Verbesserung des Codes für schnellere und genauere Authentifizierung
Erstellung der Dokumentation	Technische Dokumentation mit Beschreibung der Architektur, Schaltpläne, Code und Testprotokolle

Mit diesen Aufgaben und Zeitplänen wird sichergestellt, dass das Projekt strukturiert und effizient durchgeführt wird.

3. Anforderungsanalyse und Komponenten

3.1 Anwendungsbereiche

Das entwickelte Zugangskontrollsystem eignet sich für private, kommerzielle und industrielle Einsatzgebiete, in denen Sicherheit und kontrollierter Zugang erforderlich sind. Beispiele sind Bürogebäude, Labore, Produktionsstätten oder private Wohnbereiche.

3.2 Hardwarekomponenten

3.2.1 Übersicht verwendeter Hardware und Beschreibung

Die folgende Tabelle zeigt die für das Projekt verwendeten Hardwarekomponenten und deren Funktionen:

Bauteil	Beschreibung
Arduino Mega 2560	Mikrocontroller zur Steuerung aller Komponenten
RFID RC522 Modul	RFID-Scanner zur Kartenauthentifizierung
LCD 16x2 Display (ohne I2C)	Anzeige von Statusmeldungen
Keypad 4x4	Eingabe von PIN-Codes als alternative Authentifizierung
Servo-Motor	Steuert das Türschloss durch Drehbewegung
Grüne LED	Leuchtet, wenn Zugang gewährt wurde
Rote LED	Leuchtet bei falscher Karte oder falschem PIN
Buzzer	Gibt ein akustisches Signal bei Fehlern oder Bestätigungen
Potentiometer (10K Ω)	Zur Kontrasteinstellung des LCDs
Widerstände (220 Ω)	Begrenzung des Stromflusses für LEDs
Breadboard & Jumper-Kabel	Zur Verdrahtung aller Komponenten

3.3 Softwarekomponenten

Neben der Hardware ist auch die richtige Softwareumgebung entscheidend für das Projekt.

Software	Verwendungszweck
PlatformIO(VSCode)	Hauptentwicklungsumgebung zur Programmierung des Arduino Mega.
Arduino IDE	Alternative IDE für schnelle Tests und Debugging.
Fritzing	Erstellung des Schaltplans für die Hardware.
GitHub/Git	Versionskontrolle und Speicherung des Codes.
SerialMonitor (Teleplot)	Debugging, Überwachung von RFID-Scans und PIN-Eingaben sowie Echtzeit-Visualisierung der Daten mit Teleplot.
C++(MinGW Compiler)	Entwicklung und Kompilierung von Arduino-Code mit C++ für die Mikrocontroller-Programmierung.

4. Systemarchitektur und Design

4-1 Aufbau und Struktur

Das System kombiniert Sensoren und Aktoren, um eine sichere und effiziente Zugangskontrolle zu gewährleisten. Der Arduino Mega 2560 verarbeitet Eingaben von RFID-Scanner und Keypad, um die Authentifizierung des Benutzers zu überprüfen.

4-2 Blockdiagramm des Systems

Das Blockdiagramm zeigt den Prozessablauf:

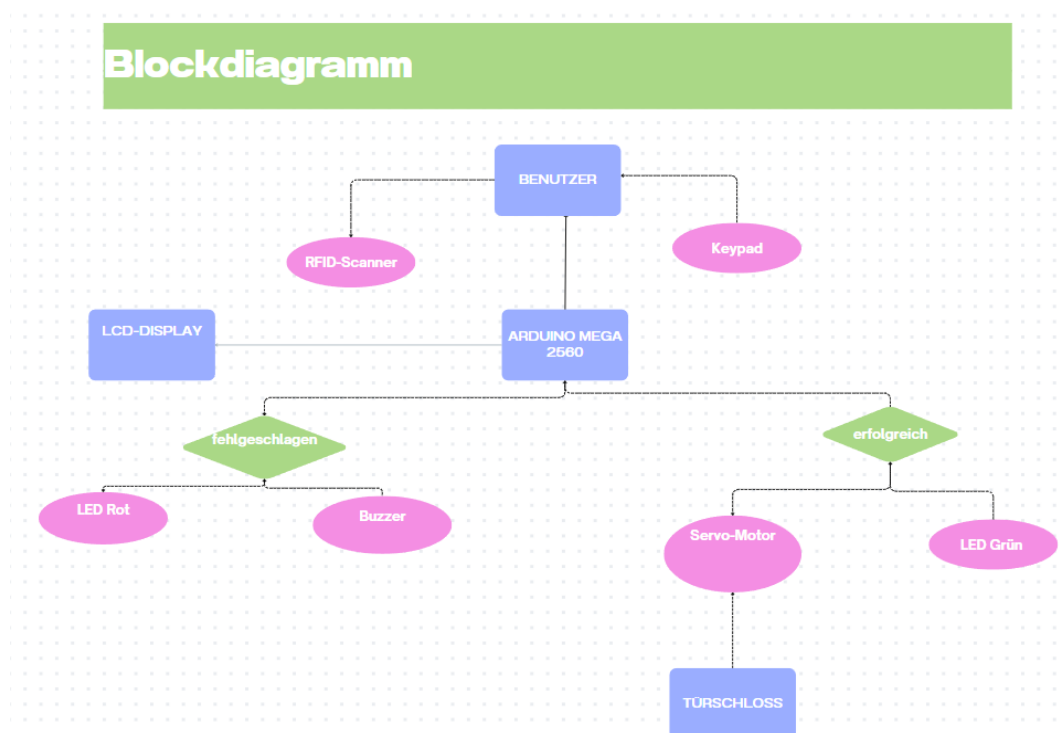


Abbildung 3-2: Blockdiagramm des Systems

5. Hardwareimplementierung

5-1 Verwendete Hardware und Pin-Zuweisung

Die folgende Tabelle zeigt die verwendeten Hardware-Komponenten und deren Pin-Zuweisung zum Arduino Mega 2560. Das System nutzt eine Kombination aus RFID-Modul, LCD-Display, Keypad, Servo-Motor, LEDs und Buzzer, um eine sichere und effiziente Zugangskontrolle zu ermöglichen.

Bauteil	RFID RC522 Pin / LCD Pin / Servo Pin / Keypad Pin / LEDs & Buzzer	Arduino Mega 2560 Pin
RFID RC522	SDA (SS)	53
	SCK	ICSP-Pin 3 (SCK)
	MOSI	ICSP-Pin 4 (MOSI)
	MISO	ICSP-Pin 1 (MISO)
	GND	ICSP-Pin 6 (GND)
	RST	10
	3.3V	ICSP-Pin 2 (3.3V)
LCD16x2(ohne I2C)	RS	7
	E	6
	D4	5
	D5	4
	D6	3
	D7	2
	V0	Mittelpin Potentiometer
Keypad 4x4	ROW1	22
	ROW2	23
	ROW3	24
	ROW4	25
	COL1	26
	COL2	27
	COL3	28
	COL4	29
Servo-Motor	Signal (Gelb)	9
LEDs& Buzzer	Grüne LED	A0 (220Ω)
	Rote LED	A1 (220Ω)
	Buzzer	A2

5-2 Schaltplan des Türsteuerungssystems

Die Schaltung zeigt die Verkabelung des Arduino Mega 2560 mit RFID-Reader, Keypad, LCD-Display, LEDs, Buzzer und Servo-Motor zur Zugangskontrolle per RFID-Karte oder PIN-Code.

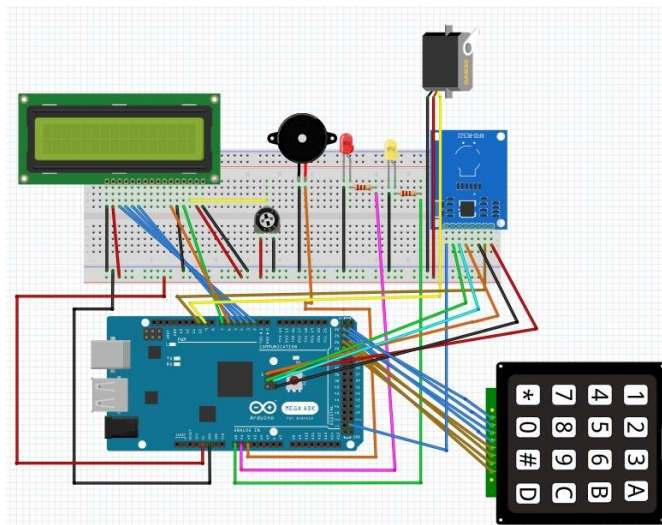


Abbildung 5-2: Schaltplan des Türsteuerungssystems

5-3 Praktischer Hardwareaufbau

Das Bild veranschaulicht die praktische Umsetzung der Türsteuerung mit allen Komponenten zur Authentifizierung und Steuerung.

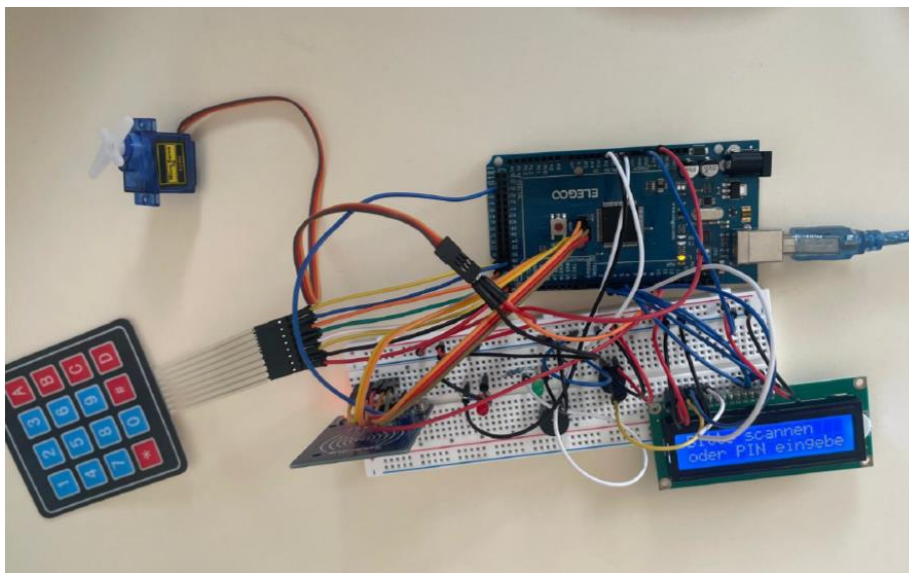


Abbildung 5-2: Hardwareaufbau des Türsteuerungssystems

6. Softwareentwicklung und Implementierung

6.1 Entwicklungsumgebung (PlatformIO)

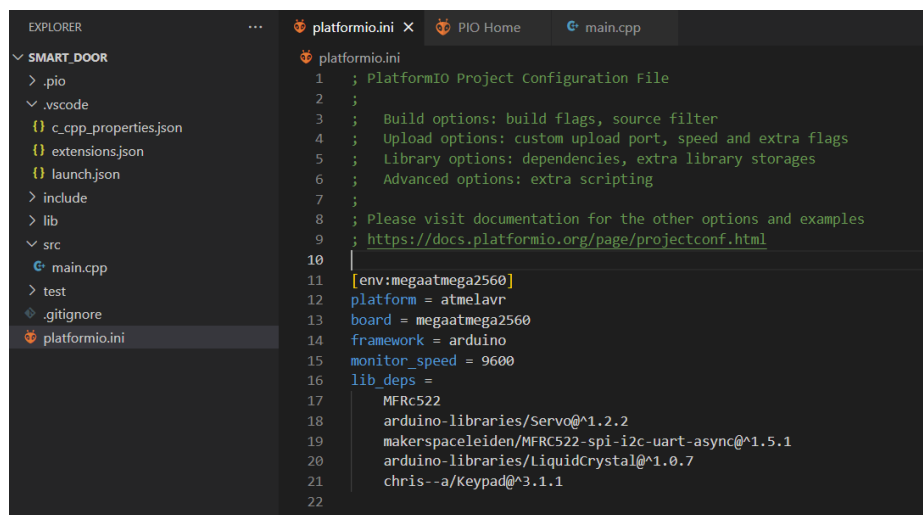
Die Programmierung wurde in PlatformIO innerhalb von Visual Studio Code durchgeführt. PlatformIO bietet eine erweiterte Entwicklungsumgebung mit einer besseren Bibliotheksverwaltung und einer optimierten Build-Umgebung für den Arduino Mega 2560.

6.2 Konfiguration der PlatformIO-Umgebung

Die Datei platformio.ini wurde entsprechend konfiguriert, um alle benötigten Bibliotheken einzubinden. Die wichtigsten Einstellungen umfassen:

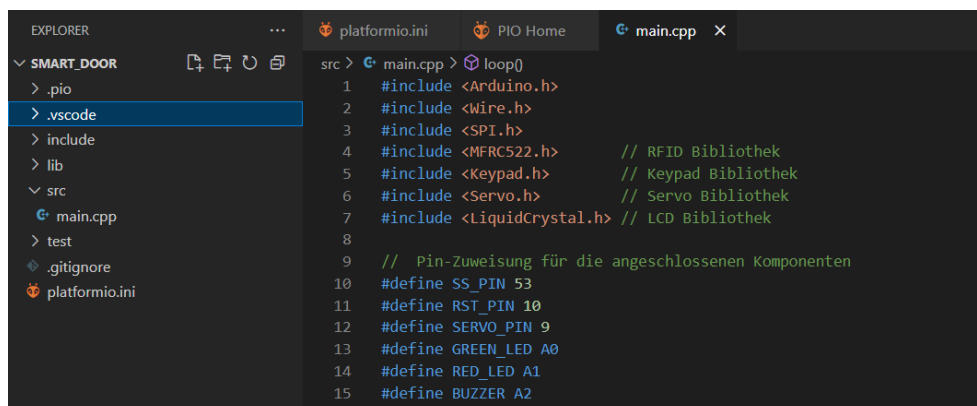
- Board: megaatmega2560
- Framework: Arduino
- Monitor Speed: 9600 Baud
- Bibliotheken: MFRC522, Servo, Keypad, LiquidCrystal

Die folgende platformio.ini Datei zeigt die vollständige Konfiguration:



```
platformio.ini
1 ; PlatformIO Project Configuration File
2 ;
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:megaatmega2560]
12 platform = atmelavr
13 board = megaatmega2560
14 framework = arduino
15 monitor_speed = 9600
16 lib_deps =
17     MFRC522
18     arduino-libraries/Servo@1.2.2
19     makerspaceleiden/MFRC522-spi-i2c-uart-async@1.5.1
20     arduino-libraries/LiquidCrystal@1.0.7
21     chris--a/Keypad@3.1.1
22
```

6.3 Konfiguration und Bibliotheken



```
src > main.cpp > loop()
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SPI.h>
4 #include <MFRC522.h> // RFID Bibliothek
5 #include <Keypad.h> // Keypad Bibliothek
6 #include <Servo.h> // Servo Bibliothek
7 #include <LiquidCrystal.h> // LCD Bibliothek
8
9 // Pin-Zuweisung für die angeschlossenen Komponenten
10 #define SS_PIN 53
11 #define RST_PIN 10
12 #define SERVO_PIN 9
13 #define GREEN_LED A0
14 #define RED_LED A1
15 #define BUZZER A2
16
```

Quellcode: PlatformIO-Konfiguration und verwendete Bibliotheken

- **Pin-Zuweisung und Initialisierung der Komponenten**

Dieses Code-Segment definiert die **Pin-Verbindungen**, initialisiert die **RFID-, Keypad- und LCD-Module** und legt die **autorisierte UID-Liste** sowie den **PIN-Code** für die Zugangskontrolle fest.

```

 9  // Pin-Zuweisung für die angeschlossenen Komponenten
10  #define SS_PIN 53
11  #define RST_PIN 10
12  #define SERVO_PIN 9
13  #define GREEN_LED A0
14  #define RED_LED A1
15  #define BUZZER A2
16
17  // Initialisierung der Bauteile
18  MFRC522 rfid(SS_PIN, RST_PIN);
19  Servo doorServo;
20  LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
21
22  // Keypad Konfiguration (4x4 Matrix)
23  const byte ROWS = 4;
24  const byte COLS = 4;
25  char keys[ROWS][COLS] = {
26      {'1', '2', '3', 'A'},
27      {'4', '5', '6', 'B'},
28      {'7', '8', '9', 'C'},
29      {'*', '0', '#', 'D'}
30  };
31  byte rowPins[ROWS] = {22, 23, 24, 25};
32  byte colPins[COLS] = {26, 27, 28, 29};
33  Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
34
35  // Gültige RFID-Tags (Hier eigene Karten-UIDs eintragen)
36  byte authorizedUID1[] = {0xF3, 0x93, 0x17, 0x2A}; // Erste autorisierte Karte
37  byte authorizedUID2[] = {0x63, 0x18, 0x1C, 0x2D}; // Zweite autorisierte Karte
38
39  // PIN-Code für den Zugang
40  String correctPIN = "1234"; // ⚠ Hier deinen 4-stelligen PIN ändern!
41

```

Quellcode: Pin-Konfiguration und Komponenteninitialisierung

7. Hauptfunktionen des Codes

7.1. Initialisierung

Die Initialisierung erfolgt in der `setup()`-Funktion. Hier werden die seriellen Kommunikationsschnittstellen, das RFID-Modul, das LCD-Display, das Keypad, der Servo-Motor sowie die LEDs und der Buzzer konfiguriert. Das LCD zeigt die erste Benutzeraufforderung an, um den Status des Systems anzuzeigen.

```

41
42 // Setup-Funktion (wird einmalig beim Start ausgeführt)
43 void setup() {
44     Serial.begin(9600);
45     SPI.begin();
46     rfid.PCD_Init();
47     doorServo.attach(SERVO_PIN);
48     doorServo.write(0); // Tür bleibt am Anfang geschlossen
49
50     pinMode(GREEN_LED, OUTPUT);
51     pinMode(RED_LED, OUTPUT);
52     pinMode(BUZZER, OUTPUT);
53
54     lcd.begin(16, 2);
55     lcd.print("Bitte scannen");
56     lcd.setCursor(0, 1);
57     lcd.print("oder PIN eingeben");
58     delay(2000);
59     lcd.clear();
60 }
61

```

Quellcode: Setup-Funktion zur Initialisierung der Komponenten

7.2. RFID-Kartenprüfung

Diese Funktion prüft, ob eine gültige RFID-Karte gescannt wurde. Falls eine neue Karte erkannt wird, wird die UID (eindeutige Kennung) mit den gespeicherten autorisierten Karten verglichen. Wenn die Karte autorisiert ist, wird die Tür geöffnet. Falls nicht, leuchtet die rote LED auf und ein Warnsignal ertönt.

```

61
62 // Funktion zur Überprüfung der RFID-Karte
63 bool checkRFID() {
64     if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) {
65         return false;
66     }
67
68     Serial.print(" RFID erkannt: ");
69     for (byte i = 0; i < rfid.uid.size; i++) {
70         Serial.print(rfid.uid.uidByte[i], HEX);
71         Serial.print(" ");
72     }
73     Serial.println();
74
75     // Überprüfung der UID für autorisierte Karten
76     bool authorized1 = true;
77     bool authorized2 = true;
78
79     for (byte i = 0; i < 4; i++) {
80         if (rfid.uid.uidByte[i] != authorizedUID1[i]) {
81             authorized1 = false;
82         }
83         if (rfid.uid.uidByte[i] != authorizedUID2[i]) {
84             authorized2 = false;
85         }
86     }
87
88     rfid.PICC_HaltA();
89     return authorized1 || authorized2; // Wenn eine der Karten übereinstimmt, wird true zurückgegeben
90 }

```

Quellcode: Kartenprüfung

7.3. PIN-Code-Verifizierung

Falls der Benutzer keinen gültigen RFID-Tag besitzt, kann er stattdessen einen vierstelligen PIN-Code über das Keypad eingeben. Das System überprüft, ob der eingegebene PIN mit dem gespeicherten PIN übereinstimmt. Ist der PIN korrekt, öffnet sich die Tür, andernfalls gibt das System eine Fehlermeldung aus.

```

108 // Funktion zur PIN-Eingabe
109 bool checkPIN() {
110     String enteredPIN = "";
111     lcd.clear();
112     lcd.print("PIN: ");
113
114     while (true) {
115         char key = keypad.getKey();
116         if (key) {
117             if (key == '#') { // Bestätigung
118                 Serial.println(" Eingegebener PIN: " + enteredPIN);
119                 return enteredPIN == correctPIN;
120             } else if (key == '*') { // Löschen
121                 enteredPIN = "";
122                 lcd.setCursor(5, 0);
123                 lcd.print(" "); // Alte Eingabe löschen
124                 lcd.setCursor(5, 0);
125             } else if (enteredPIN.length() < 4) {
126                 enteredPIN += key;
127                 lcd.print("*"); // Zeigt * statt Zahlen für Sicherheit
128             }
129         }
130     }
131 }

```

Quellcode: PIN-Verifizierung

7.4. Türsteuerung

Wenn entweder eine gültige RFID-Karte oder ein korrekter PIN-Code erkannt wurde, wird die Tür geöffnet. Dabei wird der Servo-Motor auf 90° bewegt, die grüne LED leuchtet, und nach drei Sekunden wird die Tür automatisch wieder geschlossen.

```

92 // Funktion zum Öffnen der Tür
93 void openDoor() {
94     doorServo.write(90); // Tür öffnen
95     digitalWrite(GREEN_LED, HIGH);
96     digitalWrite(RED_LED, LOW);
97     lcd.clear();
98     lcd.print("Tür geöffnet");
99     delay(3000);
100     doorServo.write(0); // Tür schließen
101     digitalWrite(GREEN_LED, LOW);
102     lcd.clear();
103     lcd.print("Bitte scannen");
104     lcd.setCursor(0, 1);
105     lcd.print("oder PIN eingeben");
106 }

```

Quellcode: Türsteuerung

7.5. Loop-Funktion – Kontinuierliche RFID- und PIN-Überprüfung

Die Loop-Funktion läuft kontinuierlich und überwacht fortlaufend, ob eine RFID-Karte gescannt oder eine PIN eingegeben wird. Sie verarbeitet die eingehenden Daten unmittelbar und entscheidet, ob der Zugang gewährt oder verweigert wird, und gibt entsprechende Rückmeldungen aus.

```

133 // Loop-Funktion (läuft kontinuierlich)
134 void loop() {
135     lcd.setCursor(0, 0);
136     lcd.print("Bitte scannen");
137     lcd.setCursor(0, 1);
138     lcd.print("oder PIN eingeben");
139
140     // RFID-Kartenprüfung
141     if (checkRFID()) {
142         lcd.clear();
143         lcd.print(" RFID OK");
144         Serial.println(" RFID akzeptiert");
145         openDoor();
146     } else if (rfid.PICC_IsNewCardPresent()) { // Falls eine falsche Karte erkannt wurde
147         Serial.println(" Falsche Karte erkannt!");
148         lcd.clear();
149         lcd.print(" Falsche Karte!");
150         digitalWrite(REDF_LED, HIGH);
151         digitalWrite(BUZZER, HIGH);
152         delay(1000);
153         digitalWrite(REDF_LED, LOW);
154         digitalWrite(BUZZER, LOW);
155         lcd.clear();
156         lcd.print("Bitte scannen");
157         lcd.setCursor(0, 1);
158         lcd.print("oder PIN eingeben");
159     }
160 }

```

```

160
161 // PIN-Eingabe über das Keypad
162 char key = keypad.getKey();
163 if (key == 'A') { // Starte PIN-Eingabe mit Taste "A"
164     lcd.clear();
165     lcd.print("PIN eingeben...");
166     if (checkPIN()) {
167         Serial.println(" PIN akzeptiert");
168         lcd.clear();
169         lcd.print(" PIN OK");
170         openDoor();
171     } else {
172         Serial.println(" Falscher PIN!");
173         lcd.clear();
174         lcd.print(" Falscher PIN!");
175         digitalWrite(REDF_LED, HIGH);
176         digitalWrite(BUZZER, HIGH);
177         delay(1000);
178         digitalWrite(REDF_LED, LOW);
179         digitalWrite(BUZZER, LOW);
180     }
181 }
182 }
183

```

Quellcode zur kontinuierlichen RFID- und PIN-Prüfung

8. Tests und Validierung

Nach der Implementierung des Zutrittssystems wurden umfassende Funktionstests durchgeführt, um die Zuverlässigkeit und das korrekte Zusammenspiel der verwendeten Komponenten zu gewährleisten. Dabei lag der Fokus auf der korrekten Erkennung von RFID-Karten und PIN-Eingaben sowie der korrekten Funktion von Servo, LEDs, LCD und Buzzer.

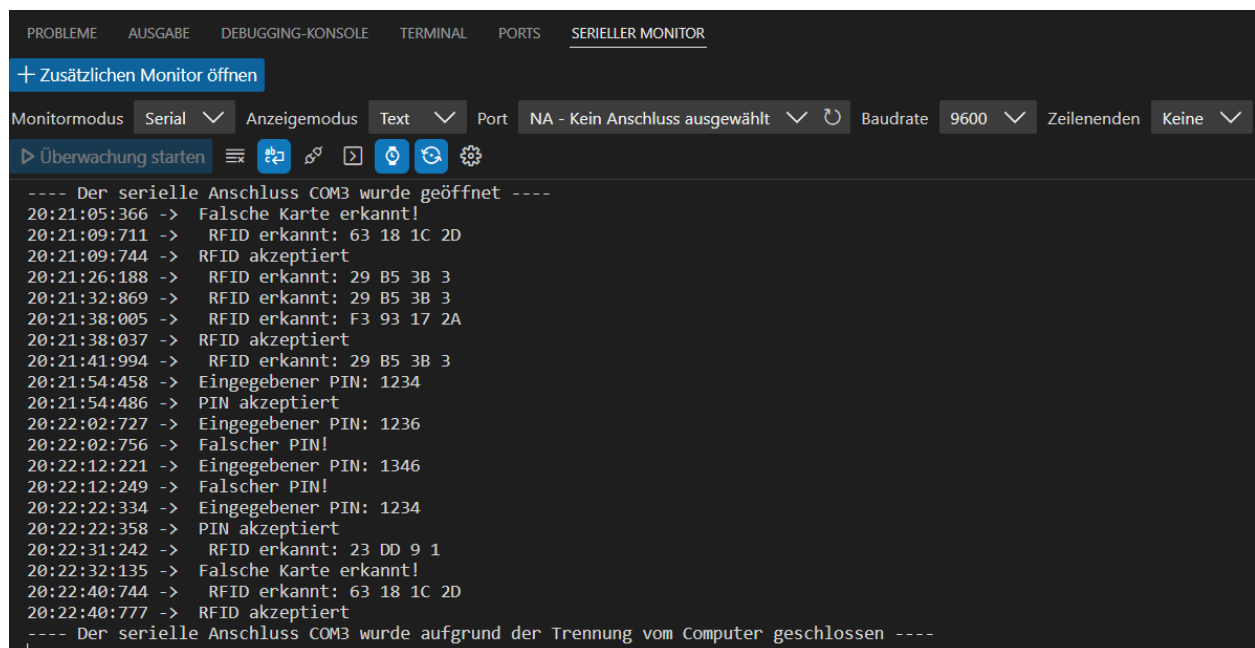
8.1 Serielle Monitor-Ausgabe zur Echtzeitüberwachung

Die Ausgabe auf dem Seriellen Monitor zeigt in Echtzeit, ob gescannte RFID-Karten oder eingegebene PIN-Codes gültig sind:

- Gültige Karten und korrekte PINs werden mit der Meldung „**RFID akzeptiert**“ bzw. „PIN akzeptiert“ bestätigt.
- Bei einer ungültigen Karte erfolgt die Ausgabe „**Falsche Karte erkannt!**“ zusammen mit akustischem Signal und roter LED.
- Bei falscher PIN erscheint die Meldung „**Falscher PIN!**“.

Dadurch können Fehler schnell identifiziert und behoben werden.

Hier Bilder der erfolgreichen & fehlerhaften Versuche (Serial Monitor):



```

PROBLEME  AUSGABE  DEBUGGING-KONSOLE  TERMINAL  PORTS  SERIELLER MONITOR
+ Zusätzlichen Monitor öffnen
Monitormodus Serial  Anzeigemodus Text  Port NA - Kein Anschluss ausgewählt  Baudrate 9600  Zeilenenden Keine
Überwachung starten
---- Der serielle Anschluss COM3 wurde geöffnet ----
20:21:05:366 -> Falsche Karte erkannt!
20:21:09:711 -> RFID erkannt: 63 18 1C 2D
20:21:09:744 -> RFID akzeptiert
20:21:26:188 -> RFID erkannt: 29 B5 3B 3
20:21:32:869 -> RFID erkannt: 29 B5 3B 3
20:21:38:005 -> RFID erkannt: F3 93 17 2A
20:21:38:037 -> RFID akzeptiert
20:21:41:994 -> RFID erkannt: 29 B5 3B 3
20:21:54:458 -> Eingegebener PIN: 1234
20:21:54:486 -> PIN akzeptiert
20:22:02:727 -> Eingegebener PIN: 1236
20:22:02:756 -> Falscher PIN!
20:22:12:221 -> Eingegebener PIN: 1346
20:22:12:249 -> Falscher PIN!
20:22:22:334 -> Eingegebener PIN: 1234
20:22:22:358 -> PIN akzeptiert
20:22:31:242 -> RFID erkannt: 23 DD 9 1
20:22:32:135 -> Falsche Karte erkannt!
20:22:40:744 -> RFID erkannt: 63 18 1C 2D
20:22:40:777 -> RFID akzeptiert
---- Der serielle Anschluss COM3 wurde aufgrund der Trennung vom Computer geschlossen ----

```

Abbildung Ausgabe des Seriellen Monitors (RFID & PIN)

8.2 Video-Demonstration

Zur praktischen Demonstration der Funktionalität wurde ein kurzes Video aufgenommen, das zeigt, wie das System zuverlässig und fehlerfrei auf gültige RFID-Karten und korrekte PIN-Eingaben reagiert.

8.3 Fehleranalyse und Lösungsansätze

Während der Testphase traten verschiedene Probleme auf, die systematisch analysiert und behoben wurden:

➔ **Problem 1: RFID-Karte wurde nicht erkannt**

Ursache: Falsche Verdrahtung oder unzureichende Stromversorgung des RFID-Moduls.

Lösung:

- ✓ Überprüfung und Korrektur der Verdrahtung, insbesondere der SPI-Verbindungen.
- ✓ Sicherstellung der Stromversorgung mit 3.3V statt 5V, um eine Überlastung des Moduls zu vermeiden.

➔ **Problem 2: LCD-Display zeigte fehlerhafte Zeichen oder blieb leer**

Ursache: Falsche Kontrasteinstellung oder lose Verbindungen.

Lösung:

- ✓ Anpassung des Kontrasts über das Potentiometer.
- ✓ Überprüfung und Neuverkabelung der Datenleitungen.

➔ **Problem 3: Servomotor reagierte nicht auf Befehle**

Ursache: Falsche Ansteuerung oder unzureichende Stromversorgung.

Lösung:

- ✓ Anschluss des Servos an einen PWM-fähigen Pin.
- ✓ Testlauf mit externem Netzteil, um Stromschwankungen zu vermeiden.

➔ **Problem 4: Ungültige RFID-Karten wurden nicht korrekt als falsch erkannt**

Ursache: Fehler in der Code-Logik zur UID-Prüfung sowie fehlerhafte Verbindung der SPI-Pins des RFID-Moduls.

Lösung:

- ✓ Anpassung des Code-Blocks zur Kartenüberprüfung, sodass bei einer falschen Karte unmittelbar eine Fehlermeldung ausgegeben wird.
- ✓ Zudem wurde festgestellt, dass bei direkter Nutzung der digitalen Pins D50 (MISO), D51 (MOSI) und D52 (SCK) des Arduino Mega 2560 keine korrekte Kommunikation mit dem RFID-Modul zustande kam. Erst nach Umstellung auf die SPI-Verbindung über den ICSP-Anschluss funktionierte das RFID-Modul einwandfrei und zuverlässig.

8.4 Ergebnisse und Diskussion

Nach Abschluss der Implementierung und umfassenden Testphase konnte bestätigt werden, dass das entwickelte RFID- und PIN-basierte Zugangskontrollsystem alle definierten Anforderungen erfüllt und zuverlässig arbeitet. Die einzelnen Komponenten (RFID-Scanner, Keypad, LCD-Display, LEDs, Buzzer und Servomotor) interagieren stabil miteinander und ermöglichen eine flexible und sichere Zutrittssteuerung. Dabei sorgen visuelle und akustische Signale für eine hohe Benutzerfreundlichkeit und eine klare Systemrückmeldung.

Hauptergebnisse im Überblick:

- **Zuverlässige RFID-Erkennung:** RFID-Karten wurden korrekt verifiziert.
- **Stabile PIN-Funktion:** PIN-Code-Eingaben wurden fehlerfrei erkannt.
- **Klare Benutzerführung:** Das LCD-Display zeigte verständliche und eindeutige Statusmeldungen.
- **Effektive Signalisierung:** LEDs und Buzzer gaben klare Rückmeldungen über Erfolg oder Fehlversuche.
- **Robuste Mechanik:** Der Servomotor steuerte das Türschloss zuverlässig und präzise.

Mögliche Erweiterungen zur weiteren Optimierung:

- **Cloud-Anbindung:** Integration einer Online-Datenbank für die Fernverwaltung und Protokollierung der Zugriffe.
- **Biometrische Erweiterung:** Zusätzliche Authentifizierung mittels Fingerabdruckscanner zur Erhöhung der Sicherheit.
- **Kabelloser Fernzugriff:** Erweiterung des Systems um eine kabellose Authentifizierung (z.B. über WiFi oder Bluetooth) mittels mobiler App oder Webinterface, um eine zukunftssichere Steuerung zu ermöglichen.
- **Protokollierungssystem:** Implementierung eines internen Speichers zur detaillierten Erfassung aller Zugriffsversuche, um die Sicherheit weiter zu erhöhen und Analysen zu ermöglichen.

Diese Optimierungen würden das System noch flexibler, sicherer und benutzerfreundlicher gestalten und für zukünftige Anforderungen rüsten.

9 Fazit

Das Ziel dieser Arbeit war die Entwicklung eines RFID- und PIN-basierten Zugangskontrollsystems auf Basis eines Arduino Mega 2560. Die definierten Anforderungen konnten vollständig umgesetzt werden, wodurch eine flexible, zuverlässige und sichere Zutrittssteuerung gewährleistet wird.

Die Komponenten RFID-Scanner, Keypad, LCD-Display, Servomotor sowie LEDs und Buzzer arbeiten stabil zusammen und ermöglichen eine benutzerfreundliche und sichere Bedienung. Dabei wurden praktische Erfahrungen im Umgang mit verschiedenen Kommunikationsprotokollen wie SPI gesammelt, sowie wertvolle Kenntnisse im Bereich der Hardwareintegration und Mikrocontroller-Programmierung gewonnen.

Abschließend bietet das Projekt zahlreiche Möglichkeiten für zukünftige Erweiterungen, insbesondere durch die Integration von Cloud-Anbindungen, kabellosem Fernzugriff sowie zusätzlichen Sicherheitsmaßnahmen, beispielsweise durch biometrische Sensoren. Somit bildet dieses Projekt eine solide Grundlage für weiterführende Entwicklungen und Anwendungen im Bereich der eingebetteten System