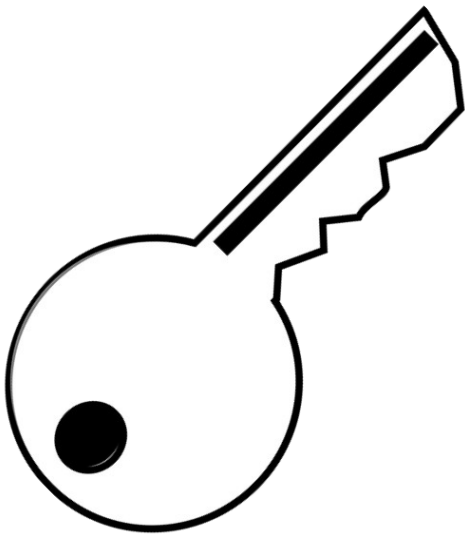


# Authetifizierungs- methoden



HBF IT 21  
Schuljahr 2022/23  
Benjamin Wende



## Inhaltsverzeichnis

Vorwort:.....	2
Einleitung:.....	2
Teamaufbau / Arbeitsaufteilung:.....	3
Hauptteil:.....	3
Aufgabenstellung:.....	3
Zeitplanung:.....	4
Kostenaufstellung:.....	5
Webserver:.....	6
OLED Display:.....	7
RFID Modul:.....	8
Nummernfeld +PCF8574 Portexpander:.....	9
Fingerabdruckscanner:.....	9
KI Kamera „Huskylens“:.....	10
3D Modell erstellen und ausdrucken:.....	11
Einbau der Komponenten:.....	11
RFID Karten:.....	12
Fazit:.....	12
Potenzielle Sicherheitslücken:.....	13
Problembehandlung:.....	13
Keypad Kommunikation:.....	13
Dauerschleife der Funktionen:.....	13
Huskylens.h Datei:.....	14
Kommunikation Fingerabdruckscanner:.....	14
Verzeichnisse:.....	14
Erklärung:.....	17
Anlagen:.....	17

## **Vorwort:**

Jeder Schüler der HBF IT muss ein Abschlussprojekt zum Ende seiner Ausbildung planen und realisieren. Dieses sollte seinem derzeitigen Wissen angepasst, der Größe des Teams entsprechend sein sowie den zeitlichen Rahmen einhalten. Unsere Ausbildung beinhaltet eine ausführliche Einführung in die Welt der Mikrocontroller und der unendlichen Möglichkeiten Programme mit dem echten Leben zu verbinden. Für das Team war dies wohl eines der spannendsten Teile der Ausbildung weshalb es sich entschlossen hat das Projekt: „Authentifizierungsmethoden“ durchzuführen. Dieses beinhaltet zum überwiegenden Teil Fachwissen aus dem IoT Fachunterricht. Beim Projekt wird auf eine Vielzahl verschiedenster Bauteile eingegangen und wie diese mit dem Mikrocontroller zu verbinden sind.

## **Einleitung:**

Das Projekt Authentifizierungsmethoden fokussiert sich auf die Integration verschiedenster Bauteile aus dem Fachbereich IoT. Es zeigt auf, wie viele verschiedenen Möglichkeiten es gibt sich zu identifizieren. Demnach kann eine Authentifizierung durch 3 Wege erreicht werden: Wissen, Haben und Biometrie. Diese 3 Wege werden in diesem Projekt dargestellt und sollen ein besseres Verständnis für das Verifizieren geben. In diesem Projekt werden die 3 Wege durch 4 einzelne Beispiele demonstriert: Fingerabdruckscanner, RFID- Scanner, Gesichtserkennung und Pin-Eingabe. Als Mikrocontroller nutzt das Team den ESP2866 mit dem WiFi Modul, der in der Schule bereits genutzt wurde. Die Kommunikation mit dem Nutzer soll mittels WLAN fähigem Gerät und einem OLED Display passieren. Der Nutzer wählt sich in das WLAN des Projektes ein. Dann verbindet er sich mit der auf dem Display angezeigten IP Adresse. Auf dieser Seite kann er nun die Projektarbeit verwalten und steuern.

Das Projekt wird programmiert mit Arduino Version 2.0.4 auf einem von der Stadt Koblenz geliehenen Laptop. Der Mikrocontroller sowie das OLED Display wurde während dem IoT Fachunterricht beschafft und stehen für das Projekt zur Verfügung. Die KI-Kamera Huskylens wurde von der Carl Benz Schule in Koblenz zur Verfügung gestellt.

## Teamaufbau / Arbeitsaufteilung:

*Tabelle 1: Teamaufbau / Arbeitsaufteilung*

<b>Teamleiter</b>	<b>IoT / Komponenten</b>	<b>Programmieren</b>	<b>Dokumentation</b>
Benjamin Wende	Benjamin Wende	Benjamin Wende	Benjamin Wende

## Hauptteil:

### Aufgabenstellung:

Auszug aus dem Projektantrag:

*„Darlegung verschiedener Authentifizierungsmethoden anhand einer Konstruktion mit Fingerabdrucksensor, RFID-Lesegerät, KI-Kamera und Pin-Eingabe. Basierend auf dem Konzept Wissen, Haben, Biometrie. Gesteuert über einen lokalen Webserver in einem passwortgeschütztem WLAN zur Verwaltung. OLED Display zur Kommunikation mit dem Nutzer.“*

Dieser Auszug beschreibt die Aufgabenstellung die sich das Team gestellt hat passend. Der Projektantrag ist im Anhang zu finden.

## Zeitplanung:

Die Zeitplanung sieht wie folgt aus:

*Tabelle 2: Zeitplanung*

Arbeitsschritt:	Angedachter Zeitraum:	Stunden:
Aufsetzen Webserver	30.01.2023-05.02.2023	5 std
Integration Display	30.01.2023-05.02.2023	5 std
Integration Pin-Eingabe + PCF8574	30.01.2023-12.02.2023	5 std
Integration RFID-Leser	05.02.2023-12.02.2023	5 std
Integration Fingerabdrucksensor	11.02.2023-26.02.2023	15 std
Integration KI-Kamera	13.02.2023-03.03.2023	15 std
3D Modell modellieren und drucken	13.02.2023-03.03.2023	5 std
Polierarbeiten (+Puffer)	27.02.2023-03.03.2023	15 std
Dokumentation	kein Zeitraum	10 std
		GESAMT:
		80 std

## Kostenaufstellung:

*Kosten die nicht angefallen sind werden durch Klammern gekennzeichnet.*

*(Notiz beachten)*

*Tabelle 3: Kostenaufstellung*

Bauteil:	Kosten:	Notiz:
DollaTek Blau Licht Optischer Fingerabdruckleser Sensormodul (siehe Quelle 2)	17,99€	
AZDelivery RFID Kit RC522 mit Reader, Chip und Card(siehe Quelle 3)	5,49€	
4x4 Matrix Array Keypad 8Pin 16 Key Membran Keyboard (siehe Quelle 4)	0,95€	
Huskylens Camera (siehe Quelle 5)	0,00€ (54,66€)	Von der Schule gestellt
Elegoo Jumper Wire 40x 20cm Female-Female, Male-Female, Male-Male Kabel (siehe Quelle 6)	6,99€	
DollaTek 3Pcs PCF8574 IO Erweiterungsplatine I/O- Expander I2C-Bus (siehe Quelle 7)	6,99€	
0,96 Zoll OLED SSD1306 Display I2C 128 x 64 Pixel	0.00€(7,99€)	Während der Ausbildung gekauft
ESP8266 serial WIFI Witty cloud Development Board	0.00€(3,70€)	Während der Ausbildung gekauft
	<b>GESAMT:</b>	
	38,41€ (103,76€)	

## Webserver:

Bei diesem Projekt bildet der Webserver und die damit verbundene HTML Seite die Hauptkommunikationsschnittstelle zwischen Nutzer und Mikrocontroller. Der Nutzer hat dadurch die Möglichkeit den ESP zu steuern und neue Passwörter zu vergeben. Dabei verbindet er sich mit dem vom ESP ausgestrahltem Netz und gibt die auf dem OLED Display gezeigte IP Adresse an. Der Webserver schickt eine HTML Seite zurück, welche eine Formular beinhaltet. Die HTML Seite wurde so klein wie möglich gehalten und so intuitiv wie möglich gestaltet.

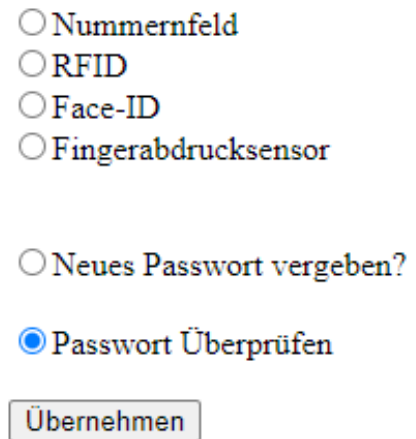
Zum Aufsetzen eine Webserver wird auf die Bibliotheken ESP8266WebServer für den Webserver und ESP8266WiFi für den AP vom Nutzer esp8266 auf Github zurückgegriffen.

Das HTML Dokument finden sie auf Github unter dem Link im Verzeichnis (Ordnername: HTML Scipt Webserver).

Die Anmeldedaten für das Netzwerk sind folgende:

SSID: Authentifizierungsmethoden

Passwort:WLAN-auth2023



☐ Nummernfeld  
☐ RFID  
☐ Face-ID  
☐ Fingerabdrucksensor

☐ Neues Passwort vergeben?  
☒ Passwort Überprüfen

Übernehmen

*Schaubild 1: HTML Form zur Kommunikation mit dem ESP*

## OLED Display:

Das OLED Display ist unser wichtigstes Tool zum Übermitteln von Nachrichten an den Nutzer.

Dieses verfügt über ein 128 mal 64 Pixel großes Monochrom OLED Display und wird mit dem SSD1306 Treiber betrieben. Es wird mittels I<sup>2</sup>C Protokoll an den ESP angeschlossen. Diesem wird eine Spannung von 5V zur Verfügung gestellt. Für das OLED Display greift das Team auf die Bibliothek Adafruit\_SSD1306 vom Nutzer adafruit auf Github zu.

Die Ausgabe der Startseite erfolgt über die Methode `startseite()`. Die meisten anderen Ausgaben werden über die Methode `displaytext()` ausgegeben. Dieser wird der dazustellende Text übergeben, sowie optional ein Delay. Bei einigen Texten kann es jedoch vorkommen, dass diese größer als eine Zeile sind und damit nicht mehr korrekt angezeigt werden. In diesem Fall werden manuell Einstellungen getroffen und der Text wird folglich ausgegeben.



*Schaubild 2: Startseite des OLED Displays*

Ein Beispielcode zum OLED Display finden sie auf Github unter dem Link im Verzeichnis (Ordnername: OLED Bild).



## RFID Modul:

Das RFID Modul RC522 kann RFID Tags lesen und beschreiben. Dieses wird mittleres SPI an den ESP angeschlossen.

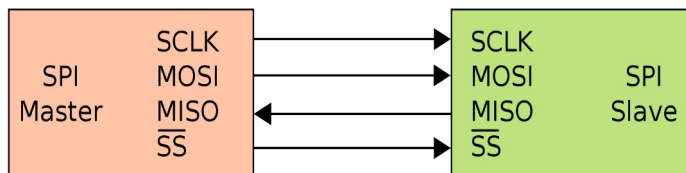


Schaubild 3: SPI Protokoll

Quelle:

[https://de.wikipedia.org/wiki/Datei:SPI\\_single\\_slave.svg](https://de.wikipedia.org/wiki/Datei:SPI_single_slave.svg)



Schaubild 4: RFID Sensor RC522 mit Chip und Karte

Quelle:

<https://www.amazon.com/SunFounder-Mifare-Reader-Arduino-Raspberry/dp/B07KGBJ9VG>

Im Code wird das RFID Modul über zwei Methoden angesteuert. Bei `RFIDnewpsswd()` wird der Nutzer aufgefordert den RFID Tag vor das RFID Symbol zu halten. Dabei wird die Seriennummer gespeichert. Bei `RFIDcompare()` wird die vorher eingespeicherte Seriennummer mit der Vorgehaltenen verglichen. Eine Ausgabe über das OLED Display informiert den Nutzer über eine richtige oder falsche Eingabe.

Durch Recherche konnte ermittelt werden, dass dieses Model auch mit 5 V Betriebsspannung arbeiten kann. Dies erspart dem Team zusätzliche Arbeit beim Verkabeln und zudem hat es bei Tests für eine höhere Reichweite gesorgt.

## Nummernfeld +PCF8574 Portexpander:

Die Einbindung des Nummernfeldes erwies sich als schwierig. Das Nummernfeld könnte man auch ohne Portexpander an den ESP anschließen. Jedoch würden 8 Pins dafür benötigt werden. Diese 8 Pins werden jetzt vom Portexpander gestellt. Dieser ist mittels I<sup>2</sup>C Protokoll am ESP angeschlossen. So werden 4 Kabel gespart.

Für diese Kombination an Bauteilen gibt es bereits die Bibliothek „Keypad\_I2C“ vom Nutzer joeyyoung auf Github.

Diese greift mittels I<sup>2</sup>C Adresse auf den Portexpander zu und verwaltet die In- und Outputs der Pins. Voraussetzung für die Bibliothek ist die Bibliothek „Keypad“ von Arduino.

Im Code wird das Nummernfeld über zwei Methoden angesteuert. Keypadnewpsswd( ) lässt den Nutzer ein neues Passwort erstellen. Keypadcompare( ) vergleicht das eingespeicherte Passwort mit dem eingegebenen Passwort. Bei beiden Methoden sieht der Nutzer seinen Pin in anonymisierter Form auf dem Display.

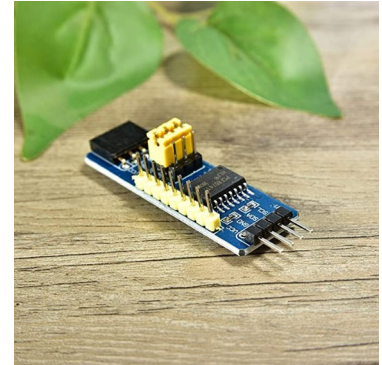


Schaubild 5: PCF8574 Portexpander

Quelle:

<https://www.amazon.de/Comimark-PCF8574-PCF8574T-Expander-Raspberry/dp/B07X3KWQZ7>

## Fingerabdruckscanner:

Der Fingerabdruckscanner von Dollatek hat einen optischen Sensor, welcher anhand von Pillarleisten Fingerabdrücke erkennen und speichern kann. Diese sind die günstigsten Fingerabdruckscanner und sind schnell auszutricksen. Er wird mittels UART an den ESP angeschlossen.

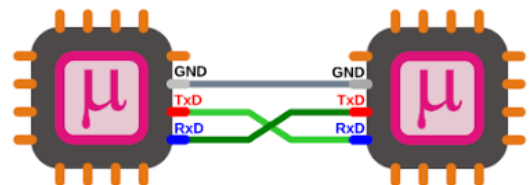


Schaubild 6: UART Protokoll Siehe: [http://www.mathe-mit-methode.com/schlaufuchs\\_web/elektrotechnik/mikrocontroller\\_lernmaterial/mikrocontroller\\_allgemein/mikrocontroller\\_ext\\_hardware/mikrocontroller\\_uart.html](http://www.mathe-mit-methode.com/schlaufuchs_web/elektrotechnik/mikrocontroller_lernmaterial/mikrocontroller_allgemein/mikrocontroller_ext_hardware/mikrocontroller_uart.html)

Im Code wird die Huskylens über zwei Methoden angesteuert. `FingerprintEnroll()` speichert den Fingerabdruck eines Nutzers ein. `Fingerprintcompare()` vergleicht den Fingerabdruck mit dem Eingespeicherten. Ausgegeben wird bei richtiger Eingabe außerdem, wie sehr der Fingerabdruck dem eingespeicherten Abdruck ähnelt. Ein Wert von 30 % ist dabei immer noch akzeptabel. Der Nutzer wird über das Display genau instruiert was er zu tun hat. Aufgrund des komplexen Vorganges während des Einspeichern kann es in vereinzelt Fällen zu Fehlern kommen. Bei Fehlern springt der ESP auf die Startseite zurück.



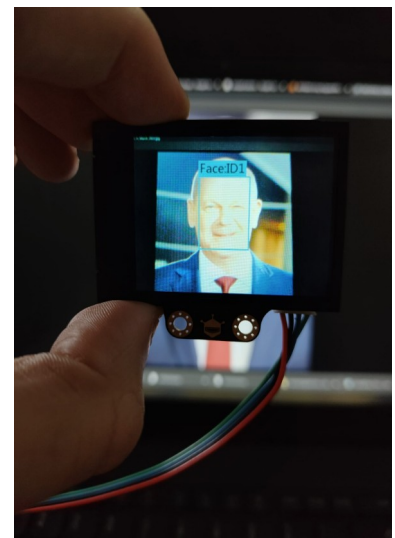
*Schaubild 7:  
Fingerabdruckscanner  
von Dollatek*

Quelle: <https://www.amazon.de/DollaTek-Optischer-Fingerabdruckleser-Sensormodul-Mega2560/dp/B07PRMXXXN>

## KI Kamera „Huskylens“:

Die Kamera Huskylens vom Hersteller DFROBOT wurde für dieses Projekt ausgewählt weil sie einen Rechenchip besitzt welcher mit künstlicher Intelligenz ausgestattet ist. Auch Gesichtserkennung kann die Kamera schon von Haus aus. Leider verfügt die Huskylens nicht über die Möglichkeit sich über den Programmcode anlernen zu lassen. Dies muss der Nutzer manuell machen.

Sie wird mittels I<sup>2</sup>C Protokoll an den ESP angeschlossen und mit 3,3V Spannung versorgt. Der Stecker der Huskylens kann einfach an die Platine des Displays (welches auch über I<sup>2</sup>C angeschlossen ist) angeschlossen werden. Da das Team die Kamera erst kurz vor Projektende von der Schule erhalten hat, konnte der vorher geschriebene Code nur in geringem Maße getestet werden.



*Schaubild 8: Huskylens  
erkennt den vorher  
eingescannten Politiker*

In dem Code wird die Huskylens über zwei Methoden angesteuert. Huskylensnewface() zeigt dem Nutzer wie er ein altes Gesicht löscht und ein neues Gesicht einspeichert. Huskylenscompare() vergleicht die Personen vor der Kamera mit der ersten eingescannten Person. Eine Ausgabe über das OLED Display informiert den Nutzer über eine richtige oder falsche Eingabe.

Da die Huskylens Eigentum der Schule ist, wurde sie nicht fest eingebaut. Sie liegt dem Projekt bei.

### 3D Modell erstellen und ausdrucken:

Mithilfe des 3D Modellierungsprogrammes Tinkercad wurden die 3D Modelle für das Projekt konstruiert. Die Vorderseite wurde mit dem 3D Drucker in der Schule gedruckt und die Schrägen mit einem privaten 3D Drucker. Die Dateien finden sie auf Github unter dem Link im Verzeichnis (Ordnername: 3D Modelle).



Schaubild 9: 3D Modell der Platte

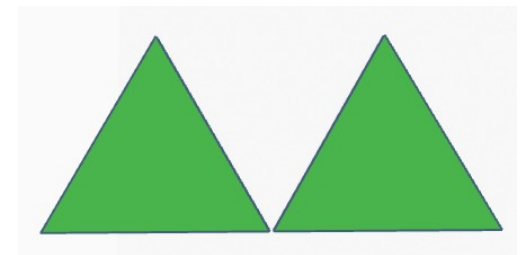


Schaubild 10: 3D Modell der Schrägen

### Einbau der Komponenten:

Die Komponenten wurden mit Heißkleber in Position gebracht und final verkabelt. Die Kabel wurden mittels Kabelbinder fixiert. Außerdem wurden bewusst unterschiedliche Farben genutzt um Abschnitte, wie zum Beispiel die Stromversorgung erkennen zu können.

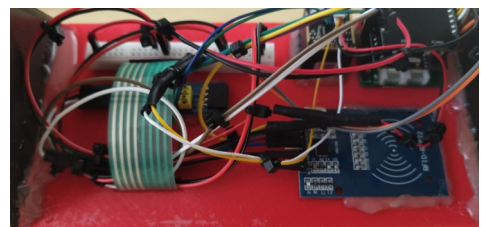


Schaubild 11: Verkabelung der Komponenten

## **RFID Karten:**

Die RFID Technologie ist mit der Zeit immer weiter geschrumpft. Mittlerweile findet man diese Technologie überall wieder. Egal ob bei der EC-Karte oder im Handy in Form eines NFC Chips. Diese RFID Chips werden in diesem Projekt bereits im Zuge der RFID Authentifizierung benutzt. Um die Bedienung zu vereinfachen wurden RFID Karten beschrieben mit sogenannten Tasks. Diese kann fast jedes NFC-fähige Gerät verarbeiten. Will sich jetzt ein Nutzer mit dem Netzwerk verbinden, aktiviert er die NFC Funktion auf seinem NFC-fähigem Gerät und hält die passende RFID Karte vor das Gerät. Eine Pop-up Nachricht sollte erscheinen, welche fragt ob er sich mit dem Netzwerk „Authentifizierungsmethoden“ verbinden möchte. Eine andere Karte leitet zur IP des Webserver mit dem Standard Browser weiter. Dies ermöglicht eine möglichst einfache Handhabung für den Nutzer. Zusätzlich sind die Tags mit einem Passwort versehen. Ein Beschreiben ohne dieses ist nicht möglich. Die Karten sind weiterhin zum Betrieb mit dem RFID Reader im Projekt nutzbar, weil diese Tasks separat als Datensatz gespeichert wird. Der RFID Reader im Projekt greift auf die Seriennummer eines Gerätes zu und Identifiziert damit den Tag.

## **Fazit:**

Als Fazit zieht das Team den Entschluss, dass das Endergebnis dem Projektantrag in allen Punkten gerecht wird. Hätte das Team noch mehr Zeit gehabt, würde sicherlich einiges überarbeitet werden. Besonders enttäuscht wurde das Team von der Huskylens, welche nicht über die Funktionen verfügt hat, welche sich das Team vorgestellt hat. Andere Komponenten haben reibungslos funktioniert und konnten problemlos mit dem ESP kommunizieren. Dies hat eine große Rolle gespielt, weil 5 Komponenten mit 3 unterschiedlichen Protokollen mit dem ESP kommunizieren müssen. Es wurden 8 Pins am ESP belegt und 22 Kabel benötigt.

**Der gesamte Programmcode sowie alle für das Projekt zusätzlich installierten Bibliotheken sind im Github Repository im Verzeichnis zu finden.**

## **Potenzielle Sicherheitslücken:**

Sicherheitslücken sind auch ein Teil dieses Projektes. Aufgabe für uns war es, möglichst alle Sicherheitslücken zu schließen. Angefangen bei den RFID Tags. Diese lassen sich mit vielen Geräten (unter anderem Handys mit NFC Funktion) beschreiben. Um unberechtigtes Beschreiben vorzubeugen, wurden die Tags mit einem Passwort geschützt. Leider gibt es auch Lücken, deren Schließung sich mit der Zeitplanung nicht vereinbaren lassen. Außerdem sind der Fingerabdrucksensor und die KI-Kamera Huskylens nicht auf dem aktuellen Stand der Technik (siehe Ultraschallfingerabdruckscanner, siehe Face ID von Apple). Diese Probleme zu lösen ist komplexer und erfordert mehr Zeit und mehr Budget.

## **Problembehandlung:**

Hier sind nun die größten Probleme gelistet, zusammen mit den gefundenen Lösungen:

### **Keypad Kommunikation:**

Beim Testen des Nummernfeldes an dem ESP wurde ein Kabel versehentlich an demselben Pin wie die Build In LED vom ESP Board angeschlossen. Daher kam an dem ESP ein falscher Wert an. Als Lösung wurde unter anderem die Anschaffung des Portexpanders beschlossen. Der Pin bleibt unbelegt.

### **Dauerschleife der Funktionen:**

Viele der im Code vorhandenen Funktionen werden permanent abgerufen, bis diese den boolean true zurückgeben. Der Watchdogtimer(kurz wdt) beobachtet wie häufig Schleifen aufgerufen werden und stuft diese dann ein. Die bei dem Projekt genutzten Funktionen werden vom wdt als „Für das System blockierend“ eingestuft. Er führt einen Neustart aus. Die Lösung bietet die Funktion `yield()`, welche immer wieder im Code zu finden ist. Diese setzt den Zähler des wdt immer wieder zurück. Er wird nicht ausgelöst. Eine Übergangslösung, welche aufgrund des Zeitdrucks geblieben ist.



## Huskylens.h Datei:

In der Bibliothek der Huskylens gibt es eine Methode, welche nicht immer einen Rückgabewert zurückgibt. Diese kann Arduino nicht kompilieren. Da diese Methode nicht von Relevanz für das Projekt ist, wurde sie gelöscht. Zudem wurde der Hersteller der Bibliothek über diesen Fehler informiert. Entfernt wird dieser Fehler aus dem Github Repository des Herstellers wahrscheinlich nicht.

## Kommunikation Fingerabdruckscanner:

Der Fingerabdruckscanner kommuniziert über das UART Protokoll, welches 2 Datenkabel benötigt (RX und TX). Eines dieser Kabel wurde ausversehen auf den selben Pin wie der Reset des RFID Moduls gelegt. Die Kommunikation mit dem Fingerabdruckscanner war dadurch fast unmöglich. Der Reset Pin vom RFID Modul wurde umgelegt.

## Verzeichnisse:

Den Code und andere Dateien finden sie hier:

<https://github.com/Rotmuetze/Scripts-HBFIT21>



*Schaubild 12: QR  
Code zum Github  
Repository*

## Abbildungsverzeichnis

Schaubild 1: HTML Form zur Kommunikation mit dem ESP.....	6
Schaubild 2: Startseite des OLED Displays.....	7
Schaubild 3: SPI Protokoll.....	8
Schaubild 4: RFID Sensor RC522 mit Chip und Karte.....	8
Schaubild 5: PCF8574 Portexpander.....	9
Schaubild 6: UART Protokoll.....	9
Schaubild 7: Fingerabdruckscanner von Dollatek.....	10

Schaubild 8: Huskylens erkennt den vorher eingescannten Politiker.....	10
Schaubild 9: 3D Modell der Platte.....	11
Schaubild 10: 3D Modell der Schrägen.....	11
Schaubild 11: Verkabelung der Komponenten.....	11
Schaubild 12: QR Code zum Github Repository.....	14

## **Tabellenverzeichnis**

Tabelle 1: Teamaufbau / Arbeitsaufteilung.....	3
Tabelle 2: Zeitplanung.....	4
Tabelle 3: Kostenaufstellung.....	5



## Glossar

Abk. Access Point.....	
AP.....	6
Abk. Hypertext Markup Language.....	
HTML.....	6
Auf RFID basierende Technologie, Übertragungsrate max. 424 kBit/s, Frequenz bei 13,56MHz.....	
NFC.....	12
Biologische Merkmale die einzigartig sind.....	
Biometrie.....	2
Einfacher Datentyp mit 2 Zuständen (true -> 1 ; false -> 0).....	
boolean.....	13
Eingebaute LED auf dem ESP Board.....	
Build In LED.....	13
Identifikationsnummer eines Gerätes in einem Netzwerk.....	
IP Adresse.....	2
Kamera welche unter anderem selbständig angelernte Gesichter erkennen kann.....	
HuskyLens.....	3
Künstliche Intelligenz.....	
KI.....	3
Mikrocontroller der Firma Espressif.....	
ESP2866.....	2
netzbasierter Versionsverwaltungsdienst.....	
Github.....	6
Protokoll mit vier Datenkabel (benannt MOSI, MISO, SS und SCK).....	
SPI.....	8
Protokoll mit zwei Datenkabel (benannt SDA und SCK).....	
I <sup>2</sup> C.....	9
Protokoll mit zwei Datenkabel (benannt TX und RX).....	
UART.....	9
Receiver Kabel bei UART Verbindung.....	
RX.....	14
Technologie für ein energiesparendes Display.....	
OLED.....	2
Technologie zum kontaktlosem Übertragen von geringen Datenmengen.....	
RFID.....	2
Transmitter Kabel bei UART Verbindung.....	
RX.....	14
(engl.) Kabelloses Lokales Netzwerk.....	
WLAN.....	2
(engl.) zurücksetzen.....	
Reset.....	14
(engl.) Internet der Dinge.....	
IoT.....	2

## Erklärung:

Ich versichere, dass ich die vorliegende Projektarbeit in allen Teilen selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe sowie dass alle wörtlichen und sinngemäßen Übernahmen aus anderen Quellen als solche kenntlich gemacht wurden.



---

Koblenz, 03.03.2023

Benjamin Wende

## Anlagen:

Projektantrag

Liste verwendeter Programme

Informationen zur Präsentation

Programme:	Bezeichnung:	Versionsnummer:	Online Anwendungen:	Bezeichnung:
	Arduino IDE	V 2.0.4		
	GIMP	V 2.10.20		Tinkercad
	LibreOffice	V 7.5.0		
	Google Chrome	V 110.0.5481.178		

Projektantrag			
Projektname:	Authentifizierungsmethoden	Projektnummer:	
Projektteam	Benjamin Wende	Datum:	13.1.2023
<b>Projektbeschreibung:</b>			
<p>Darlegung verschiedener Authentifizierungsmethoden anhand einer Konstruktion mit Fingerabdrucksensor, Gesichtserkennung wie zum Beispiel bei Face-ID, RFID-Lesegerät und Pin-Eingabe. Basierend auf dem Konzept Wissen, Haben, Biometrie. Gesteuert über einen lokalen Webserver in einem passwortgeschütztem WLAN zur Verwaltung. OLED Display zur Kommunikation mit dem Nutzer.</p> <p>Siehe: <a href="https://de.wikipedia.org/wiki/Authentifizierung#Methoden">https://de.wikipedia.org/wiki/Authentifizierung#Methoden</a></p>			
<b>Gesamtziel:</b>			
Um auf die vielen Möglichkeiten einer Authentifizierung aufmerksam machen.			
<b>Projektergebnisse:</b>			
Konstruktion mit Mikrocontroller, Fingerabdrucksensor, Gesichtserkennung wie zum Beispiel bei Face-ID, RFID-Lesegerät und Pin-Eingabe. Untergebracht in einer 3D gedruckten Hülle. Webserver zur Verwaltung.			
<b>Hauptaufgaben:</b>			
Komponente einkaufen, Libarys suchen und einbinden, einzelne Komponenten austesten, alle Kompetenten zusammen betreiben, Webserver einbinden und um Funktionen erweitern, 3D Modell konstruieren und drucken. Projektdokumentation verfassen.			
<b>Organisation:</b>			
Projektteam/ Teamsprecher	Benjamin Wende		
<b>Projektressourcen:</b>			
Projektbudget:	50€		
Sonstige Ressourcen:	Laptop (Ausgeliehen bei der Stadt Koblenz), ESP 2866, Micro USB Kabel, OLED Display. SSD1306 128x64, Kabel + +, - +, - -, Arduino IDE (Fingerabdrucksensor: DollaTek Blau Licht Optischer Fingerabdruckleser Sensormodul) RFID: RFID Reader RC522 Huskylens Nummernfeld: 4x4 Matrix Array Keypad		
<b>Termine:</b>			
Projektstart:	30.01.2023		
Meilensteine:	<ul style="list-style-type: none"> <li>Bauteile beschaffen</li> <li>Komponenten getrennt testen</li> <li>Ansteuerung der Bauteile über einen Webserver</li> <li>3D Druck anfertigen</li> <li>Gesamte Konstruktion testen und auf Funktion überprüfen</li> </ul>		
Ende Projekt:	03.03.2023		
<b>Auftraggeber:</b>		<b>Projektteam:</b>	
Datum:		Datum:	
Unterschrift:		Unterschrift:	

# Anlage zur Projektdokumentation

<b>Prüfungsteilnehmer:</b> <i>(Name, Anschrift)</i>	<b>Auftraggeber / Betrieb:</b> <i>(Name, Anschrift)</i>
<b>Prüfungsschwerpunkt</b> <i>(bitte ankreuzen)</i> <input type="checkbox"/> Anwendungsentwicklung <input type="checkbox"/> Systemintegration <input type="checkbox"/> Systemelektroniker	

<b>Thema der Projektarbeit:</b>
---------------------------------

<b>Präsentationsmittel</b> <i>(bitte ankreuzen)</i> <input type="checkbox"/> Flipchart/Tafel <input type="checkbox"/> Pinnwand <input type="checkbox"/> Tageslichtprojektor/Dokumentenkamera <input type="checkbox"/> Beamer
---

andere Präsentationsmittel: _____ <i>(müssen funktionsfähig mitgebracht werden)</i>
dafür notwendige Rüstzeit : _____ Minuten <i>(max. 15 Minuten)</i>

Mit der Unterschrift wird bestätigt, dass die Projektarbeit einschließlich Dokumentation

- a) vom Prüfungsteilnehmer selbstständig und in dem vorgesehenen Zeitrahmen angefertigt,
- b) sowie dass alle wörtlichen und sinngemäßen Übernahmen aus anderen Quellen als solche kenntlich gemacht wurden;
- c) sich nicht auf Betriebsgeheimnisse bezieht und dass keine datenschutzrechtliche Bedenken bestehen.

\_\_\_\_\_  
Ort, Datum

\_\_\_\_\_  
Ort, Datum

\_\_\_\_\_  
Unterschrift des Prüfungsteilnehmers

\_\_\_\_\_  
Unterschrift des Betreuers (inkl. Firmenstempel)