

Projekt 9: Fingerabdruck Zugangskontrolle

Projektteam:

Johannes Maier und Jonas Lochner

Projektziel:

Ansteuerung eines Fingerabdrucksensors mithilfe eines Raspberry Pi 2B. Nach der Validierung des Fingers, soll eine Kiste automatisch geöffnet werden.

Ausarbeitung:

Software: Es sollen 2 Programmteile erstellt werden, ein Benutzerteil und ein Adminteil. Der Admin soll volle Zugangs Berechtigungen haben und neue Fingerabdrücke abspeichern können. Der Benutzer soll lediglich den Verschluss Mechanismus bedienen können. Jedoch öffnet die Box sich nur wenn der Fingerabdruck im Sensor hinterlegt wurde.

Anwendermodus:

Taster wird gedrückt und der Sensor wird aktiviert. Der Sensor wartet bis ein Finger aufgelegt wurde und startet dann mit dem Scanvorgang. Nach Beendigung des Scanvorgangs wird der Fingerabdruck verifiziert. Über die LEDs wird angezeigt, ob der Anwender Zugang hat oder nicht.

Admin-Modus:

Der Admin-Modus wird über einen Fingerabdruck eines Admins erreicht oder über direktzugriff auf den Raspberry. Der zugehörige Programmteil wird über den Anwendermodus gestartet. In diesem Modus können für beide Modi weitere Fingerabdrücke hinzugefügt oder entfernt werden.

Hardware:

Komponenten:

- Holzkiste(39,6 x 29,7 x 23 x 1 cm)
- 2x Metalplatten (20 x 4 x 0,01 cm)
- Raspberry 2B+
- Fingersensor: GT -511C3 (Datenblatt im Anhang)
- Gleichstrom-Getriebemotor
- Spannungswandler
- Schrittmotorplatine
- Anzeige LEDs (rot, grün)
- 2x Endstop-Taster
- Taster zur Aktivierung des Sensors

Verkabelung

Die Stromversorgung erfolgt über ein externes Netzteil, das 12V liefert. Die 12V werden für die Schrittmotorplatine und den Getriebemotor benötigt. Die Spannung wird in dem Spannungswandler auf 5V geteilt und dient nun der Stromversorgung des Raspberry Pi. Dieser steuert über die GPIO-Pins die weiteren Module an.

Box-Bau

Zunächst werden die Löcher für die LEDs (5mm Durchmesser), den Taster (13mm Durchmesser), Stromversorgungskabel (5mm Durchmesser) und für den Sensor (2x16mm Durchmesser) gebohrt. Im nächsten Schritt muss mit einem Handfräser die Innenseite der Kiste im Bereich des Sensors und der LEDs, bei einer Holzstärke von 10mm, um 6mm ausgefräst werden, damit die LEDs und der Sensor aus der Kiste heraussehen. Danach können mit einer Feile die Löcher für den Sensor auf die richtigen Maße und Form gefeilt werden.

Für den Öffnungs-Mechanismus werden die Metallplatten vorbereitet. Hierzu werden die Platten entgratet. An einer Platte müssen auf einer Seite 4 Löcher für die Befestigung an den Motor gebohrt werden, sowie auf der anderen Seite ein Loch zur Verbindung zur anderen Metallplatte. Bei der anderen Platte werden an beiden Enden jeweils ein Loch gebohrt. Nun können die Platten miteinander verbunden werden, an den Motor befestigt werden und an den Kistendeckel angeschraubt werden. Zuletzt müssen die Endstops am Kistenrand angeschraubt werden und per Software konfiguriert werden.

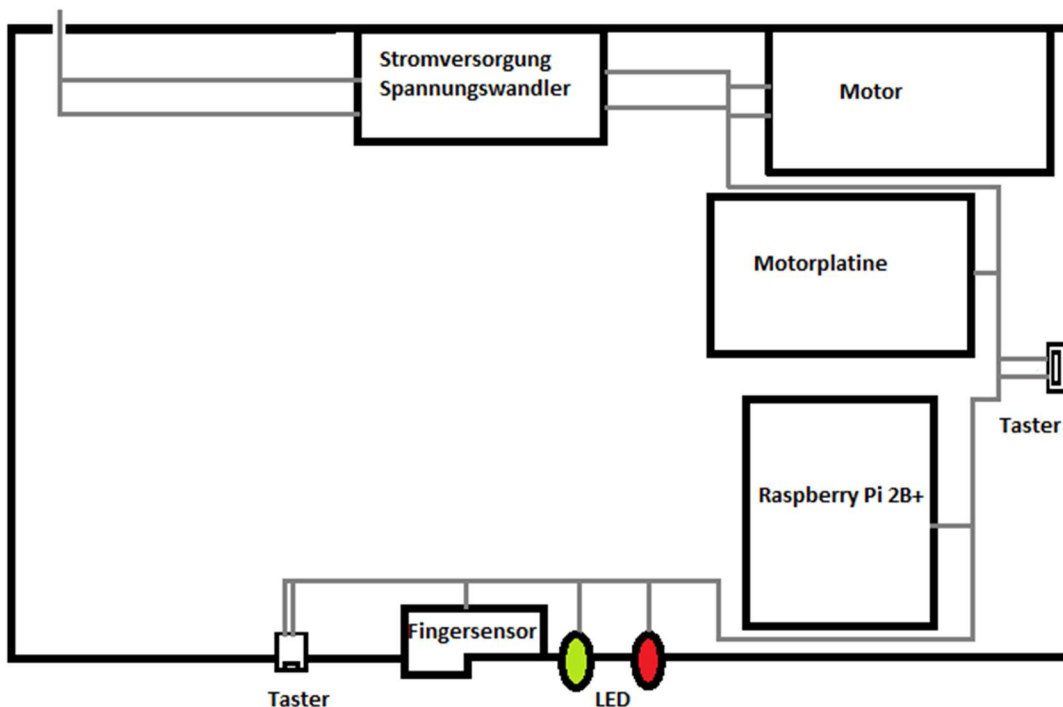


Abbildung 1-Layout

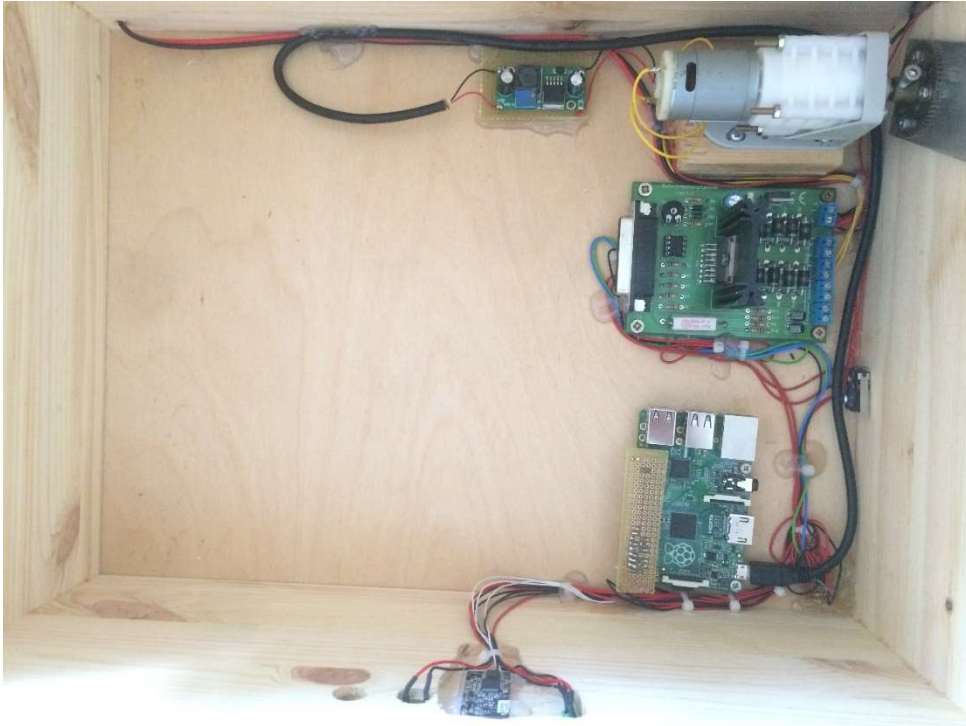


Abbildung 2 - Innenansicht



Abbildung 3 – Öffnungsmechanismus



Abbildung 4 – Außen-Vorderansicht

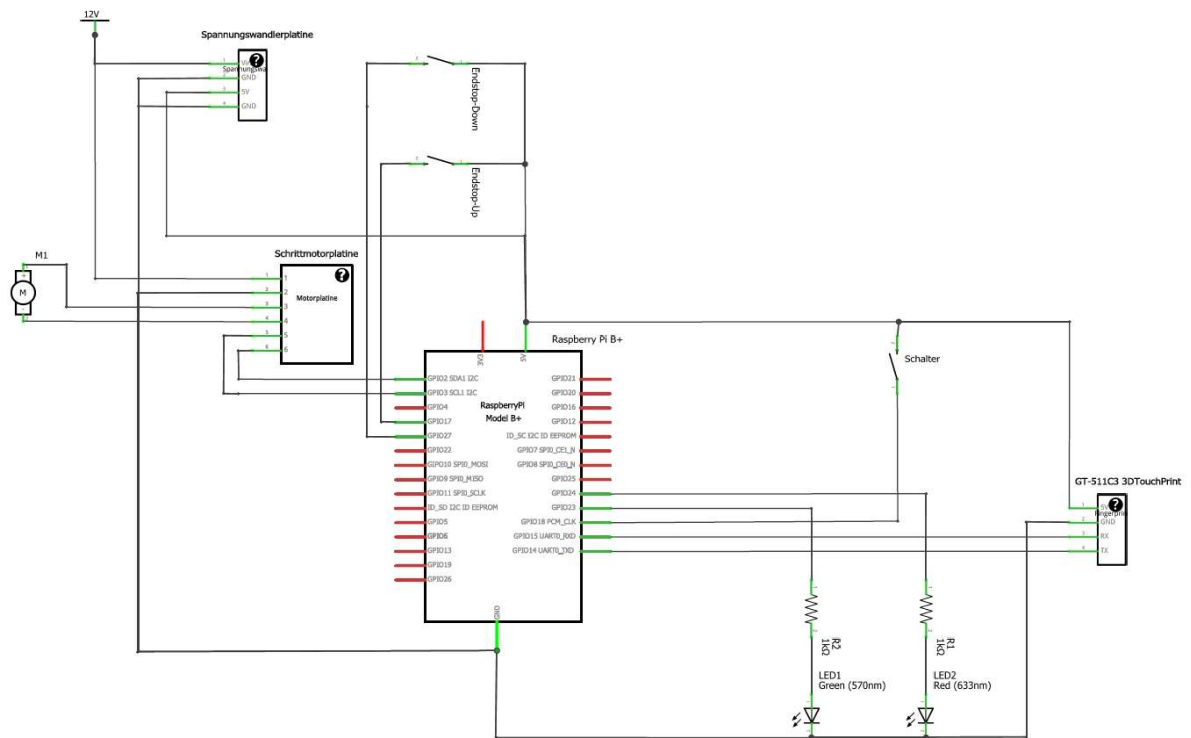


Abbildung 5- Schaltplan

Motor und seine Ansteuerung



Der Motor wird über eine Schrittmotorplatine mit 12 V Spannung und Strom versorgt. Die Platine sorgt für die Umsetzung der Signale des Raspberry auf eine höhere Spannung, sodass dadurch der Motor angetrieben werden kann. Mittels eines Potis auf der Platine kann der Verstärkungsfaktor eingestellt werden.

Damit der Motor nicht weiterdreht als es möglich ist, sprich er nicht durchdreht oder einen Getriebschaden davonträgt sind zwei Endstopper eingebaut. Es handelt sich um zwei Taster die gedrückt werden sobald die Box vollständig geöffnet oder geschlossen ist. Die Taster sind über Signalinputs an den Raspberry angeschlossen und das Programm überwacht diese während der Motor aktiv ist.

Fingerabdrucksensor GT511C3



Eigenschaften

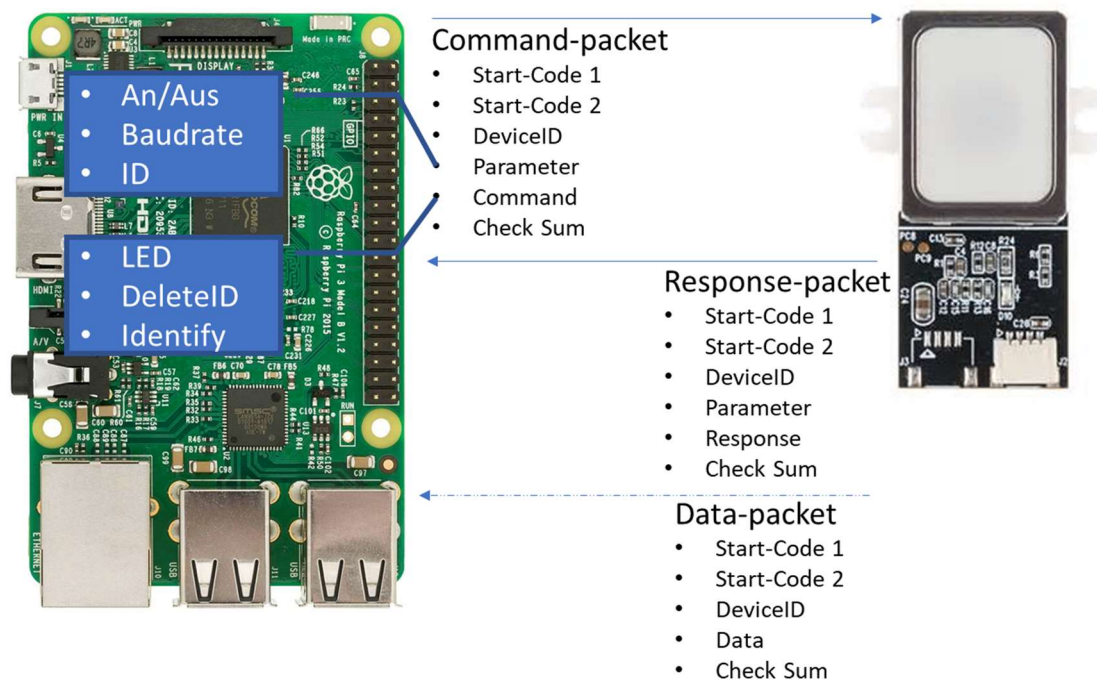
Der Fingersensor benötigt 5 V Versorgungsspannung und benutzt als Kommunikationsschnittstelle das UART und USB-Protokoll. Es ist möglich ihn mit einer Baudrate von 115200 und er kann RX sowie TX-Signale zwischen 3,3V und 6 V verarbeiten. Die Fingerabdruckerkennung und Speicherung von Fingerabdrücken geschieht auf dem Sensor selber. Die Kommunikation zwischen Raspberry und dem Sensor beschränkt sich daher nur auf Befehle und Antworten auf diese Befehle. Das heißt es fallen keine großen Datenmengen an die zwischen den beiden hin und her getauscht werden müssten.

Der Speicherplatz auf dem Sensor beschränkt sich auf 200 Fingerabdrücke. Es ist aber möglich die Daten die auf dem Sensor gespeichert sind herunterzuladen und durch andere zu ersetzen.

Es ist auch möglich die Bilder, die vom Finger erstellt werden, auf den Rechner zu übertragen.

Kommunikationsprotokoll

Die Kommunikation zwischen Raspberry und dem Fingersensor geschieht mit Paketen die über UART verschickt werden. Diese Pakete sind aufgeteilt in drei Typen: Command-Packet, Response-Packet und Data-Packet.



Alle Pakete enthalten zwei Start-Codes, die Device-ID des Sensors und eine Checksum.

Das Commando-packet wird vom Raspberry an den Fingersensor geschickt. Wie der Name schon sagt, ist es ein Befehl an den Sensor, eine bestimmte Aktion auszuführen. Zusätzlich zum Kommando wird falls nötig noch ein Parameter. (Zum Beispiel Kommando: VerifyID; Parameter:ID-Nummer)

Der Befehlssatz ist im Datenblatt vom Fingersensor festgehalten.

Das Response-packet ist die Antwort des Sensors auf ein Commando-packet. Es enthält einen Parameter (Zum Beispiel die ID-Nummer eines Identifizierten Fingers) und die Response. Diese kann entweder ein ACK oder NACK sein.

Das Data-packet beinhaltet Daten die vom Sensor an den Raspberry auf Anfrage gesendet werden. Diese Daten können zum Beispiel die gespeicherte Datenbank der Fingerabdrücke sein oder Bilder von Fingerabdrücken.

Vorhandene Bibliothek FingerPi

Für die Ansteuerung des Fingersensors existierte im Vorfeld schon eine Python-Bibliothek die speziell für den Raspberry geschrieben wurde. Diese Bibliothek beinhaltet die Initialisierung des Fingersensors, die Paketgenerierung und Versendung dieser, sowie deren Empfang. Der Befehlssatz des Fingersensors kann mit dieser Bibliothek schon voll ausgenutzt werden.

Allerdings mussten mehrere Korrekturen im Sourcecode durchgeführt werden, da entweder Funktionen den falschen Befehl ausführten, oder es zu Abstürzen kam.

Aufgrund der Bescheidenheit unsererseits sehen wir hier aber von einer Aufzählung der einzelnen Fehler und Änderungen ab. Es könnte aber auch daran liegen das wir es versäumt haben die Fehler zu dokumentieren.

Die Bibliothek enthält ein Demo-Programm mit dem der Fingersensor getestet werden kann, ob er korrekt angesprochen wird und ob er auch korrekt antwortet.

Ablauf eines Fingerabdruckscans

Hier soll kurz erläutert werden wie ein Fingerabdruck eingelesen und in der Datenbank hinterlegt wird.

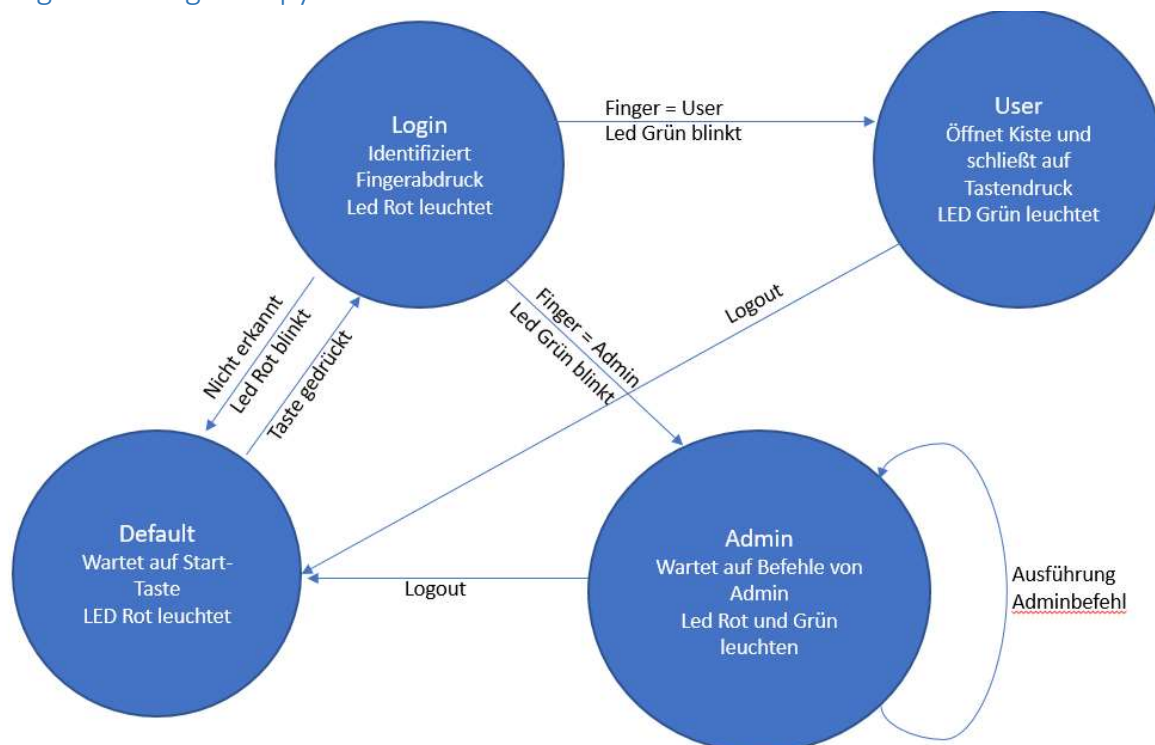
Zum einlesen eines Fingerabdrucks muss immer die Led des Fingersensors leuchten. Wenn der Scan durchgeführt wurde kann dieser dann zur Verifizierung oder zum Vergleich mit einer Datenbank hergenommen werden.

Die Prozedur für das hinterlegen eines Fingerabdrucks in der Datenbank sieht wie folgt aus (kurz Enrollment):

Es gibt 5 Phasen die in einem Enrollment durchlaufen werden.

1. EnrollStart: das Enrollment wird gestartet und ein freier Platz in der Datenbank gesucht
2. Enroll1: Der Finger wird das erste Mal gescannt
3. Enroll2: Der Finger wird das zweite Mal gescannt
4. Enroll3: Der Finger wird das dritte Mal gescannt und anschließend wird daraus ein Muster erstellt, dass in der Datenbank abgespeichert wird.
5. Der Finger wird ein letztes Mal gescannt, um ihn mit dem neuen Muster in der Datenbank zu vergleichen.

Programm FingerGo.py



Das Programm ist der ausführende Programmteil. Hier werden alle Initialisierungen getätigt und alle Zustände geschaltet. Es wird ein Finger eingelesen und mit einer, auf dem Fingersensor hinterlegten Datenbank verglichen. Dabei wird unterschieden zwischen einfachem User und Admin. Dann öffnet sich entweder die Kiste oder es können, falls man Admin ist weitere Fingerabdrücke in der Datenbank abgespeichert werden. Zum Schluss kann die Kiste geschlossen werden.

Mögliche Zustände:

Default-Modus: In diesem Zustand startet das Programm. Der Zustand wird verlassen sobald die grüne Taste gedrückt wird. In diesen Modus kehrt das Programm aus jedem anderen Zustand zurück falls ein Fingersensor nicht erkannt oder aus den anderen Modi ausgeloggt wird.

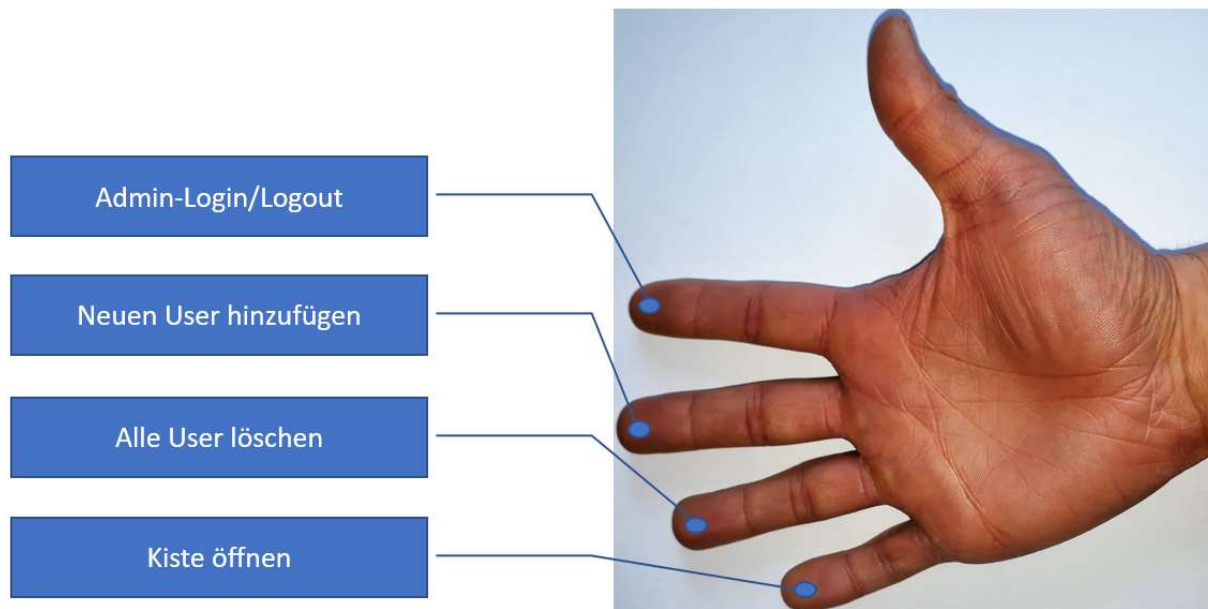
Login-Modus: In diesen Zustand schaltet das Programm bei drücken der grünen Taste, wenn es sich im Default-Modus befindet. In diesem Modus wird der Fingersensor aktiviert und der Finger eingescannt. Anschließend identifiziert das Programm den Fingerabdruck anhand einer vorher erstellen Datenbank. Sollte der Finger sich nicht in der Datenbank befinden schaltet das Programm zurück in den Default-Modus.

Im anderen Fall wechselt das Programm je nach Erkennung entweder in den Admin- oder in den User-Modus.

User-Modus: In diesem Modus wechselt das Programm nach erfolgreichem erkennen des Fingerabdrucks als User. Sobald die das Programm in diesen Modus schaltet öffnet sich die Kiste und bleibt geöffnet, bis der User durch drücken der grünen Taste entweder die Kiste wieder schließt. Zusätzlich läuft ein 5-Minuten-Timer, der den Schließmechanismus automatisch nach ablaufen der Zeit startet. Anschließend wird der User ausgeloggt und das Programm wechselt wieder in den Default-Modus.

Admin-Modus: In diesen Zustand wechselt das Programm beim Erkennen des Fingerabdrucks als Admin. In diesem Modus können alle User gelöscht, User hinzugefügt oder die Kiste geöffnet werden. Das alles geschieht über vorher festgelegte Finger des Admins (falls er/sie mehr als einen besitzt). Diese sogenannte Admin-Hand wird noch weiter unten erläutert. Nach ausführen aller gewünschten Befehle, kann sich der Admin ausloggen und das Programm wechselt wieder in den Default-Modus.

Admin-Hand



Das Bild veranschaulicht schon wie die Admin-Hand zu verstehen ist. Hier wird jedem Finger (Der Daumen ist kein Finger) ein Befehl zugeordnet, den der Admin ausführen kann. Wie eine stinknormale User-Hand zu einer Admin-Hand erhoben werden kann wird im Punkt Bedienung erklärt.

Hinter jedem Befehl steckt erst eine Identifizierung des Fingerabdrucks. Das Programm prüft nach erfolgreicher Identifizierung welcher Befehl für diesen Finger hinterlegt ist und führt ihn aus.

Logging

Die Kommunikation zwischen Raspberry und Fingersensor wird in einer Log-Datei abgespeichert, hauptsächlich um Debugging zu vereinfachen. Jedes Paket und Zustandsänderung bekommt dabei einen Zeitstempel. Es wird dafür die Logging-Bibliothek von Python benutzt.

Spezielle Funktionen `fingerfunc.py`

Die meisten Funktionen benötigen ein paar Übergabe Parameter, die sie vom Programm bekommen.

Enrollment(finger,id): Startet ein Enrollment(Hinzufügen zur Datenbank) eines neuen Fingerabdrucks. Für ein Erfolgreiches Enrollment muss der Finger dreimal korrekt eingelesen werden. Der Fingersensor erstellt automatisch nach erfolgreichem dritten Scan ein Muster des Fingerabdrucks und hinterlegt das in der Datenbank.

getfreeID(finger): Sucht einen freien Speicherplatz in der Datenbank und gibt dessen Adresse zurück

identifyFingerID(finger): Führt einen Fingerabdruck-Scan durch und identifiziert diesen. Gibt die ID des Fingerabdrucks zurück oder -1 falls nicht erkannt.

waitForFinger(finger,state): Überprüft ob ein Finger auf dem Sensor liegt oder nicht. Je nach Übergabeparameter wartet die Funktion bis der Finger vom Sensor oder auf den Sensor gelegt wird.

adminAction(finger,adminCommand): in dieser Funktion werden die Adminbefehle in Befehle für den Sensor umgewandelt und ausgeführt

getAdminCommand(id): Hier werden die IDs (also die Fingerabdrücke) den Adminbefehlen zugeordnet. Rückgabewert ist der entsprechende Adminbefehl.

setUserMode(id): Entscheidet ob die ID ein User oder Admin oder Garnichts von beiden ist. Gibt entsprechend den Modus für das Programm zurück.

motor(action): Spricht die GPIOs an und aktiviert die Pins an denen der Motor angeschlossen ist. Je nach Übergabeparameter wird der Pin zum hoch oder runterfahren auf 1 gesetzt. Als Rückgabewert wird der Status des Motors gegeben.

checkMotorState(): Überprüft in welcher Stellung sich der Motor befindet (geschlossen oder offen) und gibt das als Rückgabewert zurück.

ledSwitch(led,on): schaltet die zu spezifizierende Led an oder aus.

ledNotification(state): Hier sind zwei verschiedenen Mittelungsmodi definiert. Einer für „nicht ok“ und einer für „ok“. Entsprechend dem Übergabeparameter blinkt dann die rote oder grüne Led.

Übergabeparameter:

finger : Variable in der alle nötigen Infos des Fingersensors abgespeichert sind. Wird benötigt um ihn anzusprechen

id : Datenbankadresse auf dem Fingersensor

state : verschiedene Zustände die eingenommen werden können. Sie sind in der Datei `states.py` definiert

adminCommand: die Adminbefehle. Sie sind in der Datei `states.py` definiert.

action: Aktionen die ausgeführt werden können. Sie sind in der Datei states.py definiert.

modes: Modi die eingenommen werden können. Sie sind in der Datei states.py definiert.

Anzeigen und Bedienung der Box

Beim Anschließen der Kiste, bootet der Raspberry ein paar Sekunden. Sobald die rote Led leuchtet kann der Fingersensor aktiviert werden.

Öffnen der Kiste als User

1. Grünen Knopf drücken, der Sensor leuchtet nun blau.
2. Finger auf den Sensor legen
3. Wenn die grüne Led blinkt wurde der Finger erfolgreich erkannt und die Kiste öffnet sich und grüne Led leuchtet.
4. Zeug aus der Kiste oder in die Kiste legen.
5. Grünen Knopf drücken zum Schließen der Kiste oder einfach warten. Die Kiste schließt sich nach 5 Minuten automatisch

Adminoperationen

1. Grünen Knopf drücken, der Sensor leuchtet nun blau.
2. Finger auf den Sensor legen
3. Wenn die grüne Led blinkt wurde der Finger erfolgreich erkannt es wird in den Admin-Modus gewechselt. Beide Leds leuchten nun.
4. Entsprechend den vorher festgelegten Admin-Fingern einen Befehl ausführen. Wenn der Befehl erkannt wurde blinkt die grüne Led.
5. Ausloggen mit dem Finger für einloggen und ausloggen.
6. Falls der Fingerabdruck nicht erkannt wird, wird der Admin-Modus verlassen und man landet wieder im Default-Modus

Neuen Finger in die Datenbank einpflegen (Userfinger)

1. In den Admin-modus wechseln und Befehl „Neuen User einfügen“ ausführen.
2. Der Sensor leuchtet nun blau. Der User soll nun seinen Finger auf den Sensor legen.
3. Wenn der Finger korrekt eingescannt wurde blinkt die grüne Led
4. Der User muss diesen Vorgang insgesamt 3-mal wiederholen.
5. Wenn jedes Mal die grüne Led blinkt ist der User nun in der Datenbank eingespeichert und kann die Kiste öffnen.

Neuen Admin anlegen

Dazu muss der komplette „Normal-Mode“ des Programms, auskommentiert, und der entsprechende Admin-Teil einkommentiert werden. Das ist nur möglich wenn auf den Sourcecode zugegriffen werden kann. Eine andere Möglichkeit besteht nicht einen Admin zu erstellen.

Es werden insgesamt 4 Finger abgespeichert. Für jeden Befehl ein Finger. Jeder dieser Finger muss wie im Punkt „Neuen Finger in die Datenbank einpflegen behandelt“ werden.

Bei einem Fehler muss die komplette Prozedur wiederholt werden.

Led-Anzeigen

Rote Led:

Leuchtet: Man befindet sich im Default-Modus

Blinkt: Ein Fingerabdruck wurde nicht erkannt.

Grüne Led:

Leuchtet: Die Kiste ist offen

Blinkt: Ein Fingerabdruck wurde erkannt

Beide Leds leuchten: Programm befindet sich im Admin-Modus