

Erstellt von

Tomislav Dodig

Emin Mehic

Hannah Fehringer

Überarbeitet von

Hannah Mörth

Lena Pötzl

Janine Feyrer

Angefertigt am

Institut für

Wirtschaftsinformatik –

Communications

Engineering

Betreuer / Betreuerin

o.Univ.-Prof. Dipl.-Ing. Dr.

Christian Stary

Claudia Kaar, BSc MSc

Dezember 2021

SICHERHEIT MIT DEM INTERNET-OF-THINGS

Eine M5Stack- und Blockly-Anwendung



Diese Anleitung zeigt den Einsatz von Internet-of-Things-Technologien zur Kontrolle von Zugängen. Ziel dabei ist die anschauliche, be-greifbare Wissensvermittlung im Rahmen handlungsgeleiteter Lernprozesse. Die Ausführungen richten sich an Verantwortliche in Aus- und Weiterbildungseinrichtungen zur Integration von IoT-Szenarien in ihre Bildungsarbeit.

INHALTSVERZEICHNIS

1. Einführung	4
2. Tangibles Internet-of-Things Starter Kit und Komponenten.....	5
3. Vorbereitung der Entwicklungsumgebung.....	11
4. Internet-of-Things-System – Einrichtung und Betrieb	12
4.1. M5Burner	12
4.2. UIFlow	15
4.3. Variablen	19
4.4. Labels.....	19
4.5. Units	20
5. Realisierung Anwendungsfall.....	21
5.1. Verwendung der Tastaturkomponente.....	22
5.1.1. Hardware.....	22
5.1.2. Units	23
5.1.3. Benutzeroberfläche	23
5.1.4. Programmierung.....	26
Vollständiger Code	32
5.2. Verwendung der RFID Komponente.....	33
5.2.1. Hardware.....	34
5.2.2. Benutzeroberfläche	34
5.2.3. Programmcode.....	35
Vollständiger Code	38
5.3. Verwendung der Finger Komponente	41
5.3.1. Benutzeroberfläche	41
5.3.2. Programmcode.....	42
Vollständiger Code	45
5.4. Bewegungssensor und RGB Light.....	49
5.4.1. Hardware.....	49
5.4.2. Benutzeroberfläche	49
5.4.3. Programmcode.....	50
Vollständiger Code	51
5.5. Lichtschranke	54
5.5.1. Hardware.....	54
5.5.2. Benutzeroberfläche	54
5.5.3. Programmcode.....	55

6. Abbildungsverzeichnis	56
--------------------------------	----

1. Einführung

Die Entwicklungen rund um Internet-Technologien halten in allen Bereichen der Gesellschaft Einzug. Dieser Anwendungsfall setzt sich mit dem sogenannten Internet-of-Things (IoT) und einigen damit verbundenen Methoden und Werkzeugen auseinander. Die in der Folge vorgestellten und genutzten Komponenten und Software können eingesetzt werden, um die Sicherheit im Haus zu verbessern. Die in der Anleitung gezeigten Schritte erlauben die Konstruktion eines Sicherheitssystems, das beispielsweise Räume im Haus vor unbefugtem Betreten schützt. Das System besitzt dabei die Fähigkeiten,

- Bewegung im Raum zu erkennen
- Fingerabdrücke zu erkennen und mit anderen Fingerabdrücken zu vergleichen
- RFID (Radio-Frequency-IDentification)-Karten zu lesen und diese mit anderen Karten zu vergleichen
- Zugangsdaten wie Passwörter mittels Tastatur zu erfassen und zu kontrollieren
- mittels Lichtschranke zu erkennen, ob eine Begrenzung durchbrochen wurde oder nicht

Zur Umsetzung der Sicherheit mit dem Internet-of-Things Anwendungsfalles wird das IoT-System **M5Stack** verwendet. Es besteht aus angreifbaren Elementen, und zwar Sensoren, Kabelverbindungen und dem Steuerelement M5 Stack mit kleinem Bildschirm und einfachen Druckknöpfen. Zur Umsetzung praxistauglicher Anwendungen ist zusätzlich eine Programmierumgebung erforderlich. In unserem Fall wird die Programmierumgebung UIFlow mit der graphischen Programmiersprache Blockly eingesetzt. Der Vorteil dieser Entwicklungsumgebung ist, dass zum Arbeiten mit dem M5Stack und Blockly von UIFlow kein Vorwissen im Bereich digitaler Systeme notwendig ist.

Der M5Stack ist ein Microcontroller-Entwicklungsboard, ausgestattet mit einem Gehäuse, Grafikdisplay, microSD Steckplatz, USB-C und Lautsprechern sowie Erweiterungssteckern. An einen M5Stack können mehrere zusätzliche Komponenten mit unterschiedlichen Fähigkeiten, wie z.B. dem Erkennen von Bewegung, angeschlossen werden.

- Diese Komponenten werden als **Units** bezeichnet.
- Die Anschlüsse am M5Stack für diese Units werden als **Ports** bezeichnet.

Mit Hilfe dieser Elemente kann ein einfaches Programm entwickelt werden. Der M5Stack ist darauf ausgelegt, Units zu verbinden. Durch stufenweise Erweiterung lassen sich an- und begreifbar, und damit in überschaubaren Schritten, komplexe Systeme entwickeln. Da der M5Stack auch mit Lego-Bausteinen kompatibel ist, bietet das System auch für Kinder und Jugendliche die Möglichkeit, Hemmschwellen zur Beherrschbarkeit von digitalen Systemen, und somit auch zur Programmierung, zu überwinden.

UIFlow ist gezielt darauf ausgelegt, den M5Stack zu programmieren, und zwar mittels der visuellen Programmiersprache **Blockly** oder textbasiert in Python zu machen. Blockly ist

ein visuelles Drag-and-Drop Block-Stecksystem, mit dem die Funktionalität des M5Stack bearbeitet werden kann. Mit Blockly ist für das Programmieren wenig Schreibaufwand verbunden, sodass die Entwicklung durch einfaches Verschieben und sachorientiert erfolgen kann.

Diese Anleitung ermöglicht einen einfachen Einstieg in die Verwendung des M5Stack und seiner Komponenten, sowie der Programmierumgebung UIFlow. Sie hilft, einen Anwendungsfall mit den erwähnten Bestandteilen M5Stack und UIFlow zu erstellen. Die Anleitung umfasst folgende Kapitel neben dieser Einführung in das Thema:

- Kapitel 2 stellt das Starter-Kit und seine Komponenten vor, sowie jene zusätzlichen Units, die für die Realisierung des Anwendungsfalls benötigt werden.
- Kapitel 3 fasst die Vorbereitung der gesamten Entwicklungsumgebung in Form einer Checkliste zusammen.
- Kapitel 4 zeigt die Vorbereitung des M5Stack und seiner wichtigsten Komponenten zur Entwicklung des Anwendungsfalls.
- Kapitel 5 beschreibt die Erstellung des Anwendungsfalls. Sowohl textuell als auch grafisch wird Schritt-für-Schritt die Entwicklung eines funktionstüchtigen IoT-basierten Sicherungssystems gezeigt.

Das Sicherungssystem wird somit nicht nur benutzungsfreundlich entwickelt, sondern weist alle Eigenschaften auf, wie IoT-Systeme praktisch genutzt und im individuellen Umfeld eingesetzt werden können.

2. Tangibles Internet-of-Things Starter Kit und Komponenten

In diesem Abschnitt werden das zur Realisierung des Anwendungsfalls erforderliche M5Go IoT Starter Kit ESP32 sowie zusätzliche Komponenten erläutert und beschrieben.

Das Starter Kit von M5 enthält 6 verschiedene Units, einen M5Stack sowie Lego Bausteine. Dieses Starter Kit erleichtert, mit dem Konzept des Internet-of-Things (IoT) vertraut zu werden und wesentliche Komponenten kennenzulernen. Jede Komponente kann separat angeschafft werden, sodass der Einsatz von Komponenten in mehreren Etappen erfolgen kann.

Im Starter Kit enthalten ist:

M5Stack:

Diese Komponente ist die zentrale physische Komponente jedes IoT-Systems. An den M5Stack lassen sich alle Units anschließen. Dies ermöglicht die Kombination von mehreren Units.

Motion Sensor:

Diese Unit kann Bewegung erkennen.

Grove Hub:

Diese Unit erlaubt es, bis zu drei Units an einen Port anzuschließen.

.

RGB LED:

Diese Unit besitzt drei Lampen, die jeweils sämtliche Farben der RGB Skala annehmen können.

Abbildung 1 gibt einen Überblick über die beschriebenen Komponenten (Units) des Starter



Abbildung 1: M5Go IoT Starter Kit ESP32

Abbildung 2: Grove Hub Unit
Abbildung 3: M5Go IoT Starter Kit ESP32

Kit.

Für den Anwendungsfall werden jedoch noch weitere Units benötigt:

Grove Hub:

Mit diesem Hub lassen sich bis zu drei Units an einen Port anschließen.



4 Grove Hub

pb.Hub:

An diesen Hub lassen sich bis zu sechs Units anschließen, jedoch mit der Einschränkung, dass diese nur für den Port B gedacht sind. Eine Besonderheit von diesem Hub ist es, dass er an Port A angesteckt werden muss.



5 Pb.Hub

Weiteres RGB LED:

Diese Unit besitzt drei Lampen, die jeweils sämtliche Farben der RGB Skala annehmen können.



6 RGB LED
Komponente

Weiterer Motion Sensor:

Diese Unit kann Bewegung erkennen.



7 Motion Sensor

Card Kb:

Diese Unit ist eine Tastatur und erlaubt die interaktive Eingabe von Zeichen.



8 CardKB

RFID:

Diese Unit ermöglicht die Erkennung von RFID-Karten.



9 RFID

Laser RX:

Diese Unit funktioniert nur in Kombination mit der Unit Laser TX. Sie empfängt den Infrarotstrahl von Laser TX.



10 Laser RX

Laser TX:

Diese Unit funktioniert nur in Kombination mit der Unit Laser RX. Diese Unit sendet einen Infrarotstrahl, der von der Laser RX Unit empfangen werden kann.



11 Laser TX

Finger:

Diese Unit repräsentiert eine Fingerscan, der einen Abdruck einscannen und mit vorab gespeicherten Fingerabdrücken abgleichen kann.



12 Finger Unit

Sobald alle Komponenten für den Sicherheit mit dem Internet-of-Things Anwendungsfall richtig zusammengesteckt wurden, ergibt sich mit dem M5Stack und den notwendigen

3. Vorbereitung der Entwicklungsumgebung

Die folgende Checkliste erleichtert den Einstieg in die IoT-Anwendungsentwicklung. Die Liste enthält sämtliche notwendigen Vorkehrungen, um den Anwendungsfall mit dem M5Stack zu realisieren.

- ☐ Installation von M5Burner auf Computersystem - M5Burner ist unter <https://m5stack.com/pages/download> zu finden
- ☐ Verwendung des M5Burner um auf dem M5Stack die aktuellste Firmware zu installieren (mehr Info unter [M5Burner](#))
- ☐ Installieren von Treiber CP2104 Driver des M5Stack – dieser ist unter <https://m5stack.com/pages/download> zu finden
- ☐ Installation von lokaler UIFlow-Programmierungsumgebung auf Computersystem (Mehr Info unter [UIFlow](#))
- ☐ Verwendung der Sicherheit mit dem Internet-of-Things Anwendungsfall-Anleitung, um mit den Units, UIFlow und M5Stack vertraut zu werden

4. Internet-of-Things-System – Einrichtung und Betrieb

Der M5Stack bildet gemeinsam mit den in diesem Abschnitt vorgestellten Software-Entwicklungskomponenten das Internet-of-Things-System. Zur Entwicklung und zum Betrieb wird der M5Stack mit dem Computersystem über ein USB-C Kabel angeschlossen. Sobald der M5Stack eingeschaltet wird, ist es notwendig, den richtigen Modus auszuwählen, um eine Verbindung mit dem Computersystem herzustellen. Dafür drückt man Knopf C, um in die Setup Kategorie zu gelangen. Von dort wählt man „Switch mode“ um das Gerät richtig einzustellen. In Abbildung 12 wird der M5stack mit seinen verschiedenen Ports und Knöpfen gezeigt.

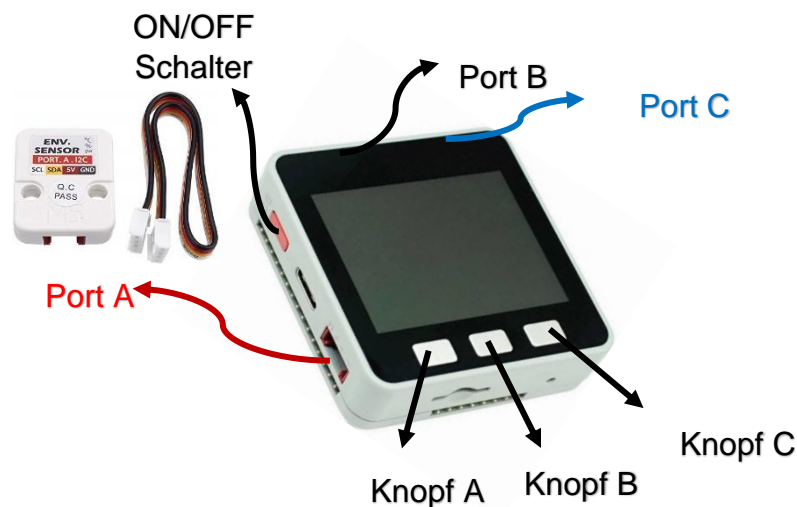


Abbildung 14: Beschriftung vom M5Stack

Es kann passieren, dass der M5Stack in einem anderen Modus hängt und auch nach erneutem Einschalten sofort in diesen Modus wieder zurückgeht. Um diesen Modus zu verlassen, muss sehr schnell innerhalb der ersten paar Sekunden nach dem Einschalten mit Taste C in den Setup Modus kommen. Dort kann dann ein anderer Modus vorzugsweise USB Modus ausgewählt werden. Dann kann man den gewünschten Modus einstellen.

4.1. M5Burner

Bei der allerersten Verwendung des M5Stack muss man den M5Burner verwenden. Der M5Burner kann von der Seite <https://m5stack.com/pages/download> heruntergeladen und durch ihn die Firmware auf dem M5Core installiert werden. Dieser muss dieselbe Version haben wie UIFlow (wenn man damit arbeitet).

Um den M5Burner heruntergeladen zu können, muss auf den obigen Link geklickt werden. Anschließend wird man zu einer Seite geleitet, auf der verschiedene Software heruntergeladen werden kann. In dieser Liste muss der Punkt „M5Burner“ gesucht und den sich daneben befindlichen „Download“-Button geklickt werden und das für das eigene Betriebssystem richtige Package ausgewählt werden. Dieses wird dann heruntergeladen. Sobald der Download abgeschlossen ist, kann die .zip Datei entpackt werden.

Den M5Burner kann man nach dem Download über die Datei „M5Burner.exe“ ausführen, wie in der nächsten Abbildung gezeigt.

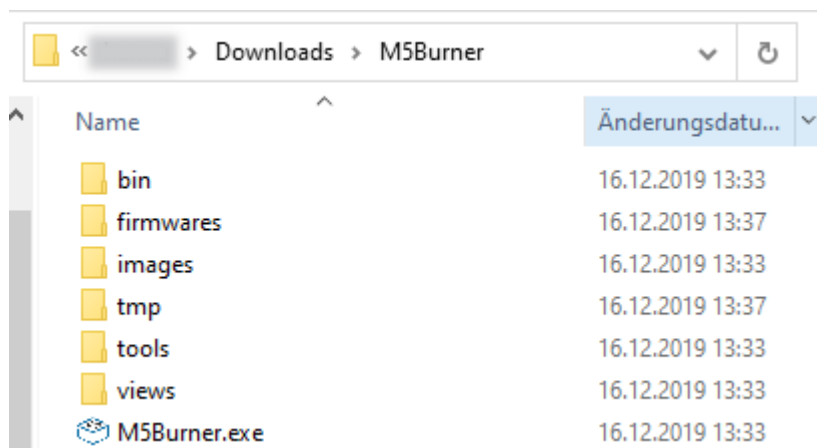


Abbildung 19: Ordnerstruktur vom M5Burner

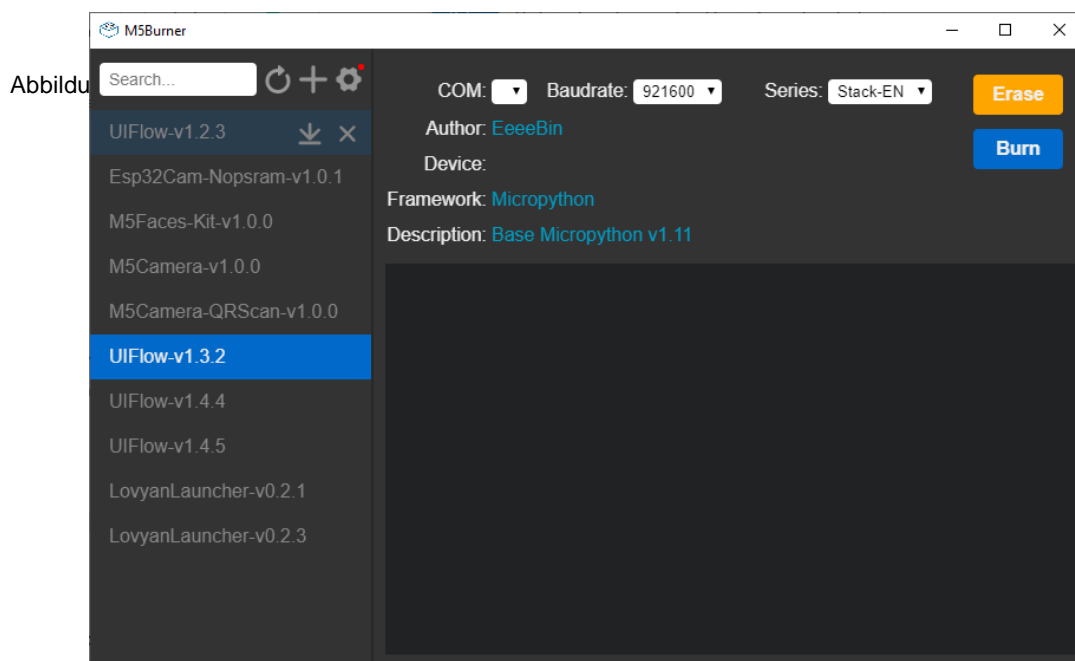



Abbildung 16: Benutzeroberfläche vom M5Burner

Abbildung 17: Ordnerstruktur vom M5BurnerAbbildung 18: Benutzeroberfläche vom M5Burner

Sobald man den M5Burner geöffnet hat, muss man die richtige UIFlow-Version laden. Dafür muss man auf das Download-Symbol  neben der ausgewählten UIFlow-Version klicken. Danach sind der M5Stack über das USB-C Kabel an den Laptop/PC anzuschließen und der richtige COM-Anschluss auszuwählen. Anschließend kann man auf Burn klicken und die Firmware wird auf dem M5Stack installiert. Empfehlenswert ist es hier die Version „UIFlow-v1.4.5“ zu installieren. Die Darstellung vom M5Burner wird in Abbildung 5 (**Fehler! Verweisquelle konnte nicht gefunden werden.**) gezeigt. In dieser Abbildung wurde die UIFlow-Version „UIFlow-v1.3.2“ ausgewählt.

Es kann vorkommen, dass das Computersystem den M5Stack nicht erkennt. Sollte dies geschehen, kann der Treiber „CP2104 Driver“ für den M5Stack, ebenfalls von der Seite

<https://m5stack.com/pages/download> geladen werden. Sollte es dennoch nicht funktioniereb, empfiehlt es sich im Gerätemanager nachzuschauen. Es ist möglich, dass dann bei demPunkt „Anschlüsse“ ein kleines,gelbes Warndreieck ist. Durch Klick auf die rechte Maustaste kann man dann oft einen neuen Treiber auswählen. Hier muss der zuvor heruntergeladene Treiber gewählt werden.

4.2. UIFlow

Die lokale Programmierumgebung von UIFlow kann von <https://m5stack.com/pages/download> geladen werden. Innerhalb der UIFlow-Oberfläche wird beim Starten der lokalen Programmierumgebung nach einem verbundenen M5Stack gefragt, welcher in der COM: Schaltfläche ausgewählt wird, wie in Abbildung 15 zu sehen ist.

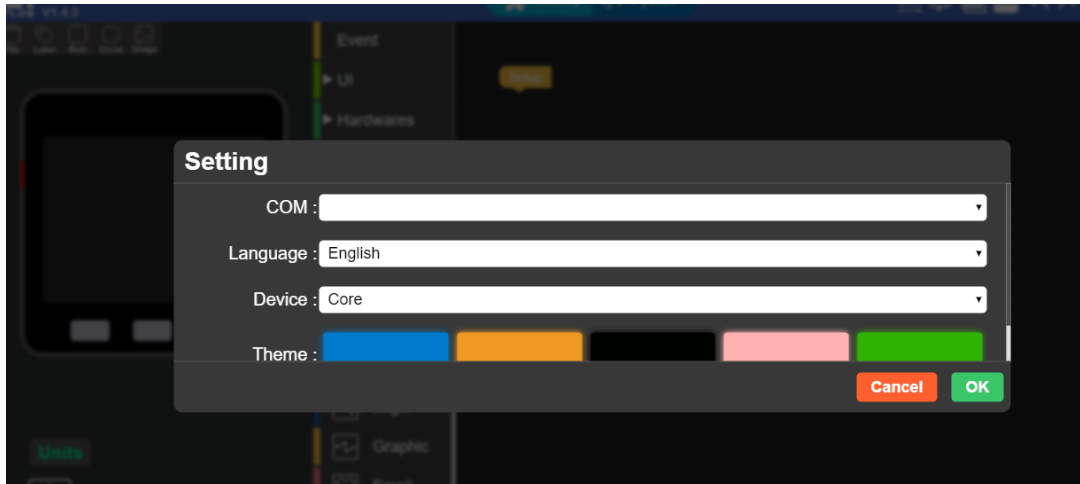


Abbildung 22: Setting Auswahl vom UIFlow

Abbildung 23: Die Ansicht, nachdem man auf das Windows Symbol einen Rechtsklick ausführte
Abbildung 24: Setting Auswahl vom UIFlow

Üblicherweise wird der korrekte COM-Anschluss automatisch angezeigt. Sollte dies jedoch nicht der Fall sein, so kann man über den Gerätemanager des Computersystems herausfinden, welcher COM-Anschluss in UIFlow zu verwenden ist. Dafür klickt man mit der rechten Maustaste auf das Windows Symbol im unteren linken Bereich des Desktops und anschließend wählt man „Geräte-Manger“ aus, wie in der folgenden Abbildung zu sehen ist. Die weiteren Einstellungen braucht man nicht zu beachten.



Abbildung 25: Die Ansicht, nachdem man auf das Windows Symbol einen Rechtsklick ausführte

Unter „Anschlüsse (COM & LPT)“ kann man dann herausfinden, über welchen COM der M5Core angesprochen werden kann. Diese Information findet man, sobald „Silicon Labs CP210x USB to UART Bridge (...)“ unter den Anschlüssen aufscheint.

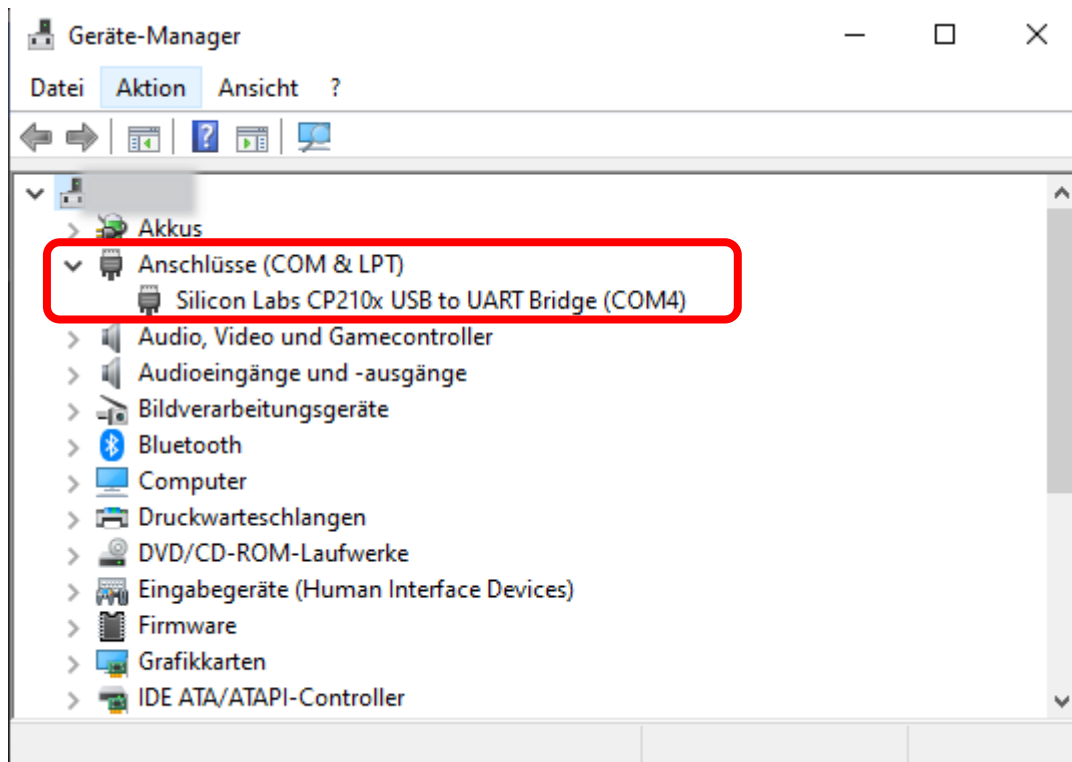


Abbildung 26: Anzeige der verschiedenen Anschlüsse am Computer

Dieser COM-Anschluss muss dann auf der Startseite von UI-Flow eingegeben werden.

Nach einer erfolgreichen Verbindung ist die Nutzung vom M5Stack und UIFlow vorbereitet. Dies erkennt man, wenn am unteren linken Rand des Bildschirms der Status „Connected“ sichtbar ist. Diese Anzeige ist in der nächsten Abbildung zu sehen.



Abbildung 27: M5Stack-Verbindung wird erfolgreich angezeigt.

Danach ist die Nutzung der UIFlow-Plattform möglich – die ersten Schritte für die Entwicklung einer eigenen Anwendung sind gesetzt. In Abbildung 19 werden die Benutzungsschnittstelle dargestellt und die wichtigsten Komponenten hervorgehoben. Diese Komponenten werden unterhalb des Bildes noch genauer erklärt.

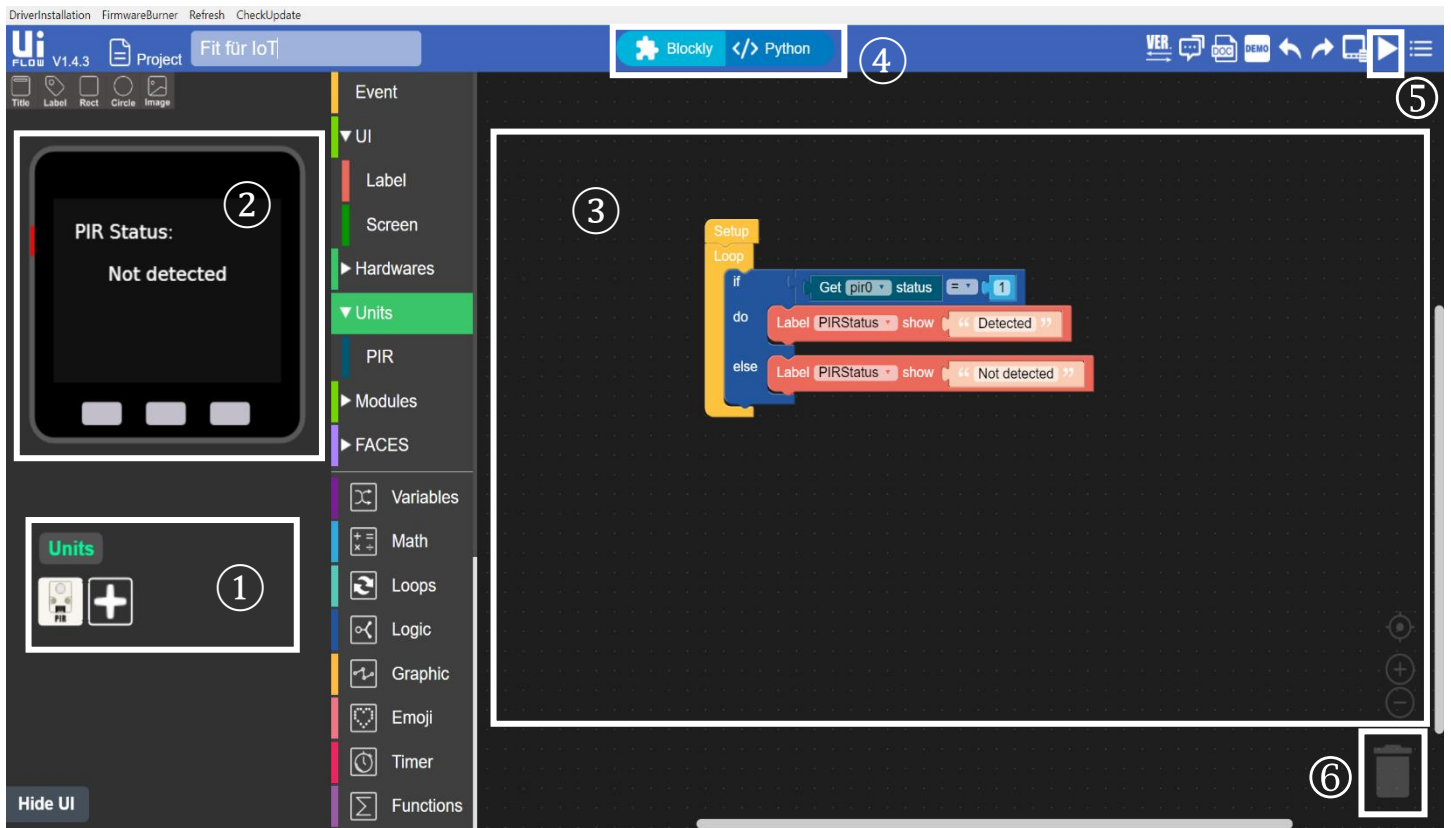


Abbildung 28: Erklärung zur UIFlow Oberfläche (anhand eines PIR Beispiel).

1. Hier können jene Units hinzugefügt werden, die man programmieren möchte. Nähere Informationen zu Units siehe Abschnitt [Units](#).
2. Die Oberfläche des M5Core kann hier kreiert werden. Kreise, Texte, Quadrate und Bilder können per Drag and Drop eingefügt werden.
3. Dies ist der Programmierbereich (Coding Area). Hier erstellt man per Drag and Drop seinen Code, den man später ausführen möchte, zusammen.
 - a. Blöcke sind Bausteine in verschiedensten Farben mit verschiedensten Funktionen, die benutzt werden, um eine Ablauflogik zu programmieren. Abhängig davon, welche Funktion man benötigt, kann man einen Block per Drag und Drop auf der Oberfläche einfügen. Im Beispiel werden Blöcke aus dem Event-, PIR- und Label-Block verwendet.
4. Hier kann man zwischen den Programmiersprachen Blockly und Python wechseln.
5. Damit spielt man den Code temporär auf den M5Core, welcher diesen ausführt, sofern er korrekt zusammengestellt wurde.
6. Per Drag and Drop kann man zu diesem Papierkorb Blöcke hinziehen, die man nicht mehr benötigt, damit sie gelöscht werden.

Die Blöcke, die man zum Programmieren verwendet, befinden sich in dem Panel auf der linken Hälfte der UIFlow-Oberfläche. Dort kann zuerst eine Metakategorie ausgewählt werden und anschließend der Block, den man benötigt, per Drag and Drop in die Coding Area gezogen werden. Zur Veranschaulichung wird der Event-Block in Abbildung 20 verwendet, welcher besonders wichtig für den Start eines Programms ist.



Abbildung 29: Event-Block in UIFlow

Um mit dem Programmieren sinnvoll starten zu können, sollte man jedoch davor noch ein die in der Folge beschriebenen Konzepte kennen.

4.3. Variablen

Variablen stellen sozusagen das Gedächtnis eines Programms dar. Sie können Werte speichern, die im Zuge der Ausführung eines Programms geändert bzw. verwendet werden können, und zwar durch den Aufruf des Namens der Variable. Beispiele unterschiedlicher Variablen sehen wir in der nächsten Abbildung.

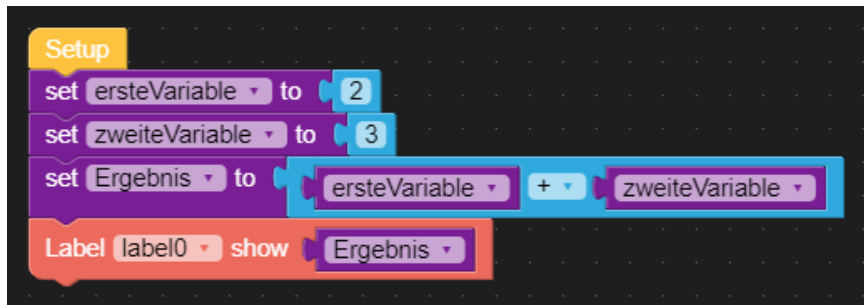


Abbildung 30: Beispiel für Variablen – Berechnung von $2 + 3$ inklusive Ausgabe

4.4. Labels

Als Label werden die Textplatzhalter auf dem M5Core bezeichnet, mit denen man gewisse Werte oder Strings auf dem LCD-Display des M5Core ausgeben kann.

Um ein Label auf dem M5Core anzuzeigen, muss man dieses von der linken oberen Ecke in UIFlow per Drag and Drop auf die Oberfläche des M5Core ziehen, wie in der Abbildung 22 zu sehen ist.



Abbildung 31: Drag und Drop von einem Label auf den M5Core

Danach erscheint in der Auswahlleiste für die Blöcke unter UI eine Subkategorie Label. Somit kann man jetzt die Blöcke auswählen, mit denen man die Ausgabe auf dem M5Core verändern kann. Ein solcher Block wird in der Abbildung 23 gezeigt.

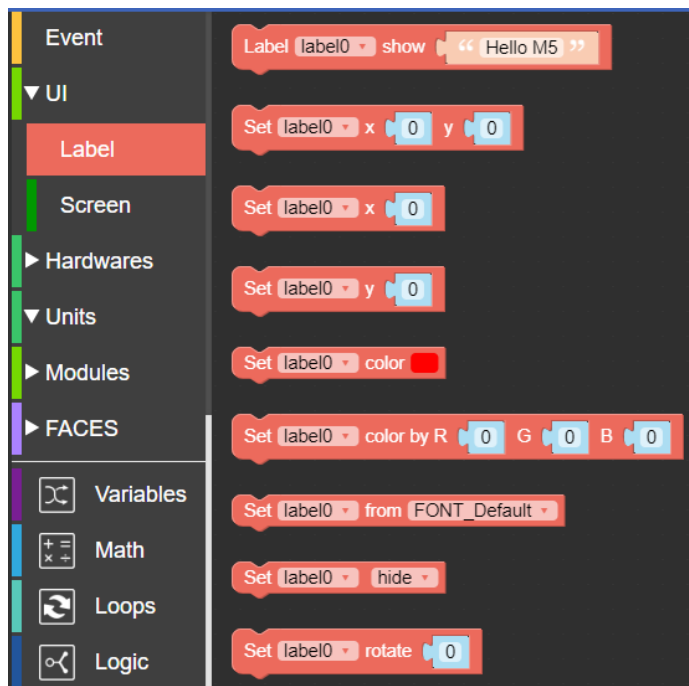


Abbildung 32: Block Ansicht für "Label"

Das Prinzip für Title, Rect, Circle und Image ist das gleiche wie für Label, mit Ausnahme der Subkategorien unter „UI“, die dann Title, Rect, Circle oder Image heißen, abhängig davon, welche Felder per Drag and Drop auf den Screen des M5Stack gezogen wurden.

4.5. Units

Units sind Sensoren, die man an den M5Core anschließen kann. Diese liefern Daten, die weiterverarbeitet werden können. Ein Beispiel ist der Environment-Sensor, der die Umgebungstemperatur misst und weiterleiten kann.

Um eine Unit auf einfache Art und Weise in UIFlow zu verwenden, muss sie zuallererst unter Units über das „+“ Symbol, im linken unteren Eck von UIFlow hinzugefügt werden. Hierfür einfach auf das „+“-Symbol klicken, welches in der Abbildung 24 abgebildet wird.

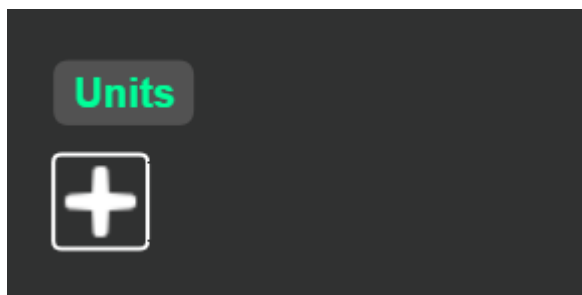
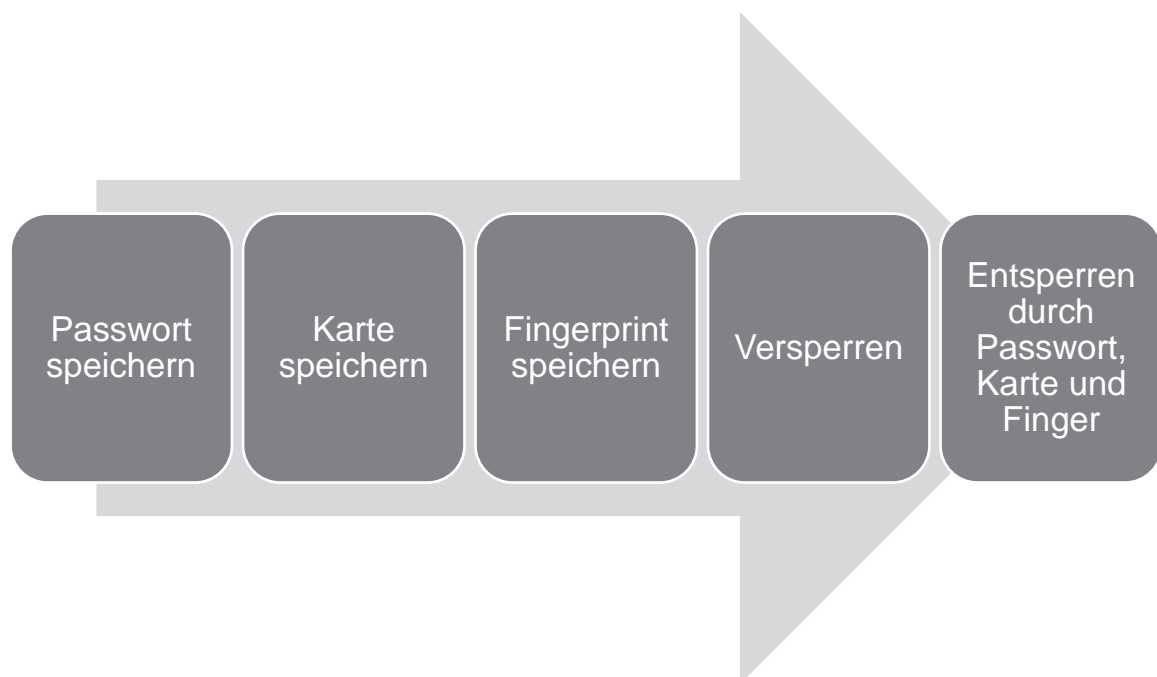


Abbildung 33: Hinzufügen einer Unit in UIFlow

5. Realisierung Anwendungsfall

Der „Sicherheit mit dem Internet-of-Things“-Anwendungsfall betrifft ein Inhouse-Sicherungssystem. Er erlaubt, Räume zu sichern und nur bestimmten Personen Zutritt zu gewähren. Falls sich jemand dem gesicherten Raum nähert, zeigt das System dies an. Kinder und Jugendliche haben oft den Wunsch, dass die Eltern nicht jederzeit das eigene Zimmer betreten und wünschen sich ein entsprechendes Frühwarnsystem. Der Anwendungsfall zeigt die Installation der erforderlichen physischen Geräte und die Programmierung des Sicherungssystems. Die Idee hinter dem Anwendungsfall wird in Abbildung 25 dargestellt.



34 Ablauf Anwendungsfall

In der Folge wird detailliert erklärt, welche Entwicklungsschritte zu setzen sind, um eine funktionstüchtige Anwendung zu erhalten. Jede/r kann entwickeln und das System somit selbst bauen, da die einzelnen Units einfach anzustecken sind und die Programmierung mit einfachen visuellen Symbolen erfolgt.

Da der Use Case doch sehr groß ist und viele verschiedenen Komponenten und Aspekte beinhaltet wird er in kleine, überschaubarere Teile aufgebrochen. Die einzelnen Arbeitspakete sind

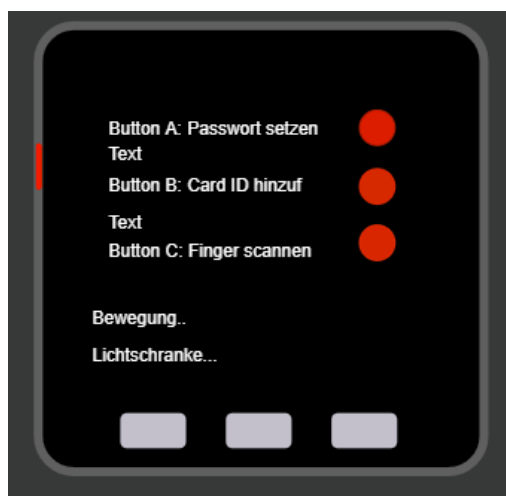
- das Verwenden der Tastatur,
- das Einlesen einer Karte mit Hilfe der RFID Unit,
- der Fingerprint Scanner,
- der Bewegungssensor,

- die Lichtschranke.

Die Arbeitspakete können der Reihe nach Bausteinartig hinzugefügt werden, wobei auch teilweise der bereits bestehende Code modifiziert werden muss. Das Programm ist bereits nach der Implementierung des ersten Arbeitspaketes einsatzbereit, wird aber die stetige Ergänzung der weiteren Arbeitspakete in jedem Schritt sicherer und komplexer.

Es ist anzumerken, dass das Programm regelmäßig manuell gespeichert werden muss, da sonst die Gefahr besteht, den Code zu verlieren.

Die folgende Abbildung zeigt, wie die Benutzeroberfläche des m5Stacks aussehen soll, wenn alle Komponenten implementiert wurden.



35 Benutzeroberfläche

5.1. Verwendung der Tastaturkomponente

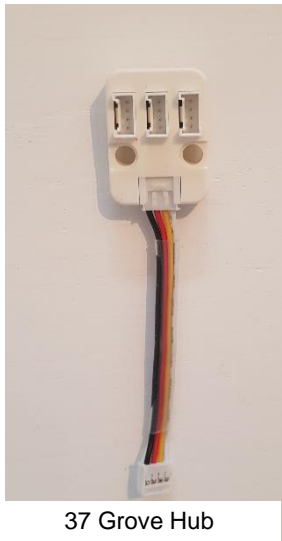
Das erste Arbeitspaket beschäftigt sich mit der Verwendung der CardKB Unit, also der Tastatur sowie mit dem grundlegenden Setup.

Am Ende dieses Arbeitspaketes kann ein neues Passwort durch Eingabe auf der Tastatur angelegt werden, durch Drücken der Taste A gespeichert werden. Außerdem kann das Haus durch Betätigen der Taste C verriegelt werden und anschließend durch Eingabe des richtigen Passworts wieder entriegelt werden. Dabei gibt es außerdem visuelle und akustische Signale.

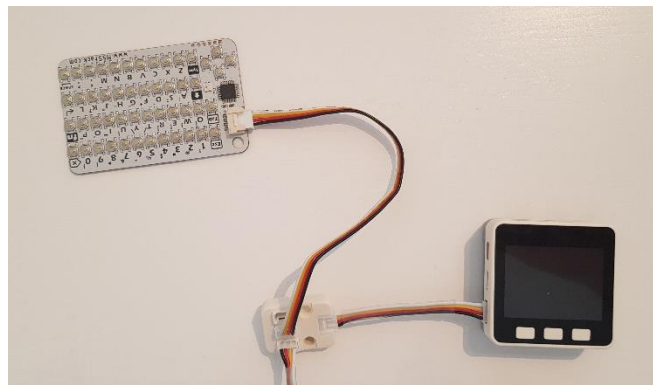
5.1.1. Hardware

Dafür werden folgende Komponenten benötigt:

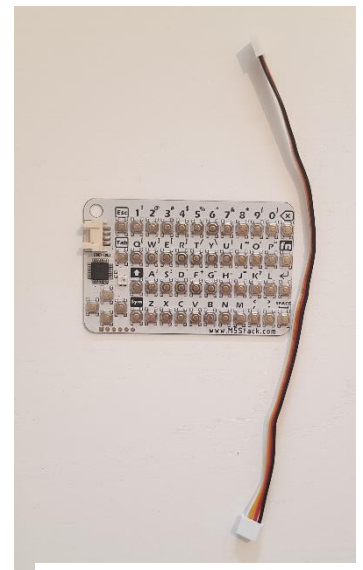
- M5 Stack,
- 2 Grove Kabel,
- Grove Hub



37 Grove Hub



36 M5Stack mit Grove Hub und CardKB Komponente



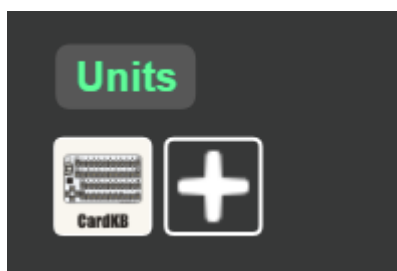
38 CardKB Komponente

Der Grove Hub wird mit Hilfe eines Grove Kabels an Port A des M5Stacks angeschlossen. Anschließend wird an einen Port des Grove Hubs mit dem zweiten Grove Kabel die CardKB Unit angeschlossen. Der Grove Hub fungiert hierbei als Verteiler. Er ermöglicht es also, dass bis zu drei Units an Port A angeschlossen werden können. Zwar wird dies im ersten Arbeitspaket noch nicht benötigt, spart dann aber in den folgenden Abschnitten Arbeit.

5.1.2. Units

Bevor nun mit der eigentlichen Programmierung gestartet werden kann, müssen die benötigten Units in UIFlow hinzugefügt werden.

Wie im Abschnitt Units erwähnt, klickt man auf das „Plus“ Symbol, um eine Auflistung aller vorhandenen Units zu erhalten. In Abbildung 30 findet sich eine Darstellung des „Plus“-Symbols. Anschließend wählt man die gewünschte Unit aus und fügt diese durch einen Klick auf „OK“ hinzu. Für das erste Arbeitspaket wird nur die Unit CardKB gebraucht.



39 Hinzufügen der Komponente in UIFlow

5.1.3. Benutzeroberfläche

Sobald alle benötigten Units hinzugefügt wurden, ist im nächsten Schritt, die Benutzeroberfläche vom M5Stack zu erstellen. Dabei werden wir uns auf „Label“ und „Circle“ konzentrieren. In der Abbildung 31 sehen wir verschiedene Objekte, die der M5Core Oberfläche hinzugefügt werden können.



Abbildung 40: Die verschiedenen Möglichkeiten für die Benutzeroberfläche

Wie bereits im Abschnitt [Labels](#) erklärt, wird ein Label per Drag und Drop auf die Oberfläche gezogen. Danach ist es notwendig, dem neu erstellten Label einen passenden Namen sowie Text zu geben, um es später ansprechen zu können.

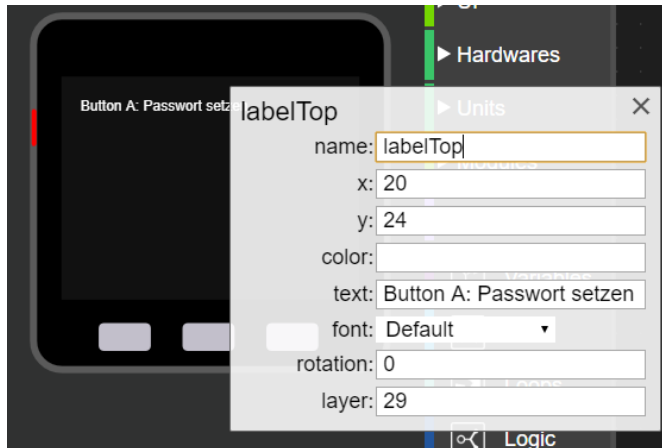


Abbildung 41: Label-Einstellungen

Per Mausklick auf das Label werden die verschiedenen Einstellungen angezeigt, die man vornehmen kann. Diese Einstellung wird in der Abbildung 32 abgebildet. Hier vergeben wir dann den Namen „labelTop“ und als Text verwenden wir „Button A: Passwort setzen“. Als nächstes werden wir Circle verwenden, um später die Farbe auf Grün oder Rot ändern zu können, abhängig davon, ob die Eingabe richtig oder falsch war.

Analog zu Label wird Circle per Drag und Drop positioniert und durch einen Klick auf die linke Maustaste werden die Einstellungen geöffnet. Als nächstes weisen wir Circle einen neuen Namen zu, „circle0“. Für den nächsten Schritt ist zu beachten, dass sowohl „borderColor“ als auch „backgroundColor“ dieselbe Farbe verwenden. Diese Anzeige wird in der nachfolgenden Abbildung gezeigt.

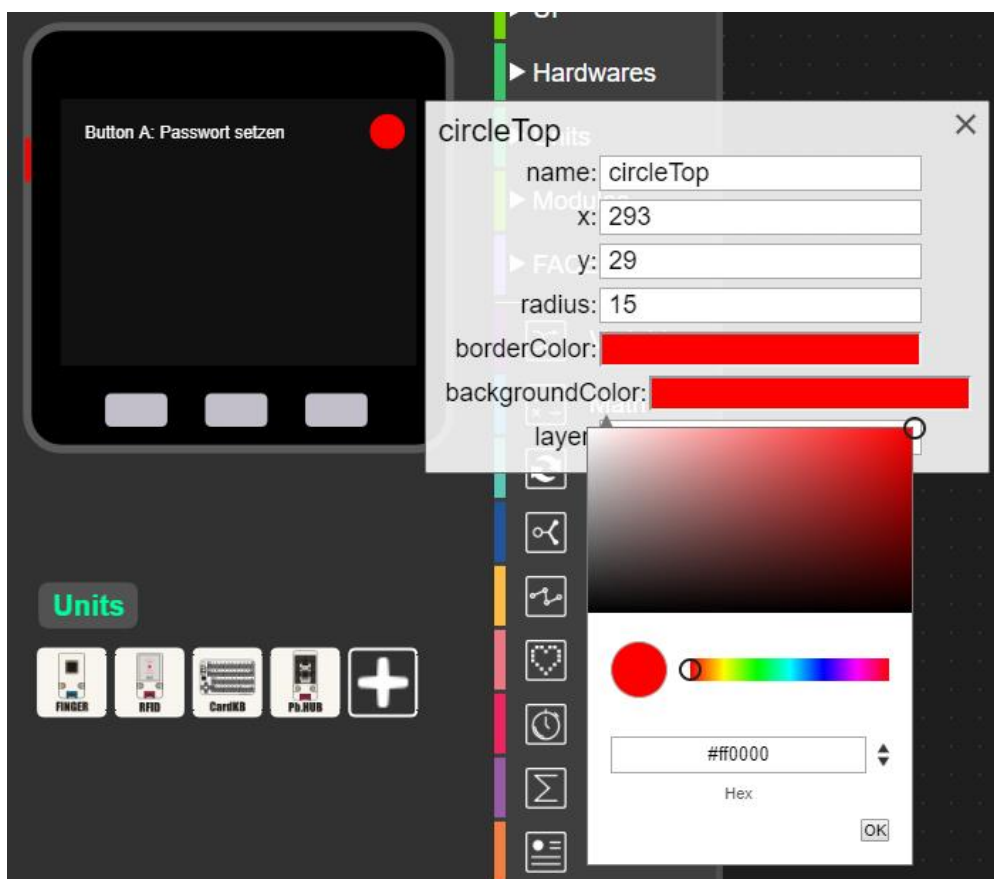


Abbildung 42: Circle-Einstellung

Um die Benutzeroberfläche für den Anwendungsfall fertig zu stellen, benötigen wir einige weitere Labels und Circles. Die genaue Bezeichnung aller Komponenten wird in der Abbildung 33 gezeigt.

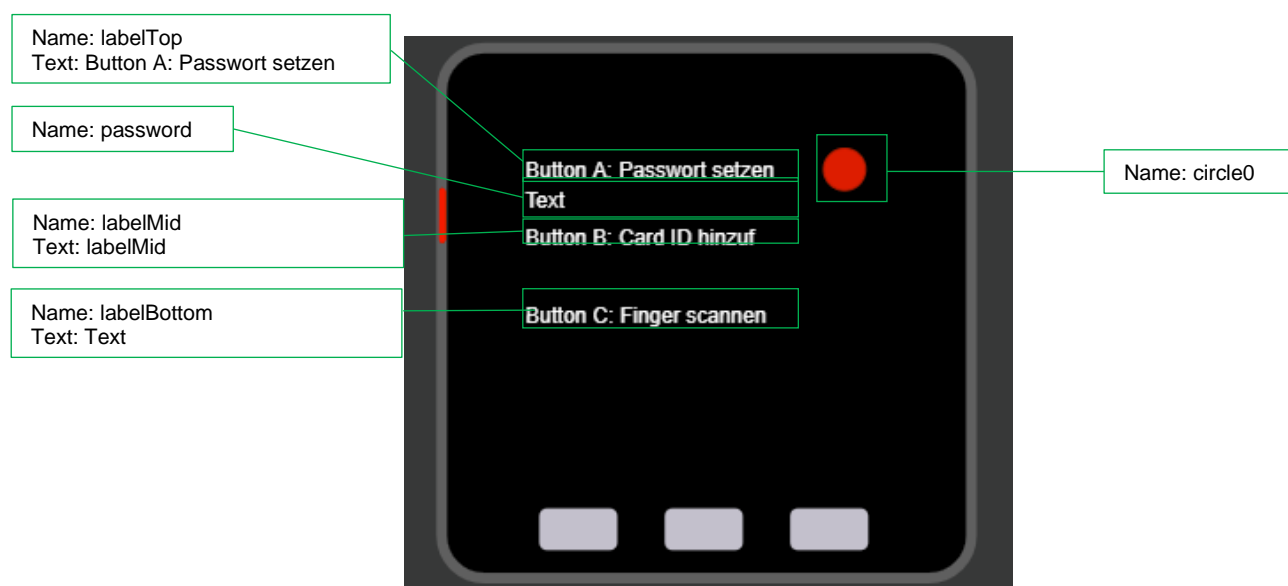
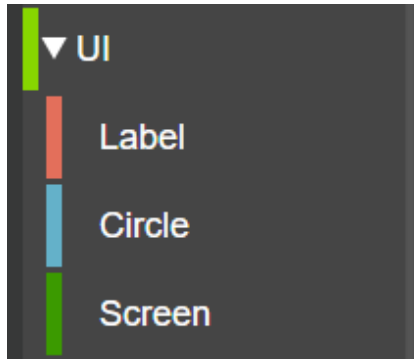


Abbildung 43: M5Stack Benutzeroberfläche

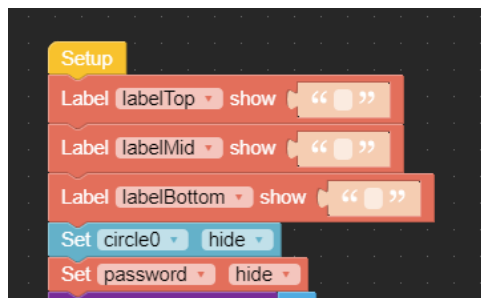
5.1.4. Programmierung

Startpunkt des Programms ist der Setup Block. Hier wird mit der Ausführung des Programms begonnen.



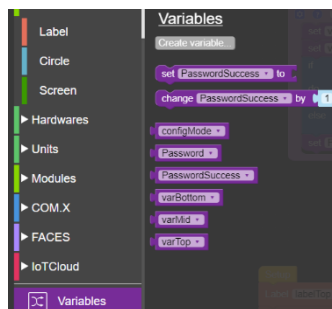
44 Kategorie UI

Im ersten Schritt muss einmal das User Interface erstellt werden. Die zuvor definierten Label sollen zu Beginn noch nichts anzeigen, dafür findet man in der Kategorie UI unter dem Punkt Label den passenden Block. Ebenso in der Kategorie UI findet man den Punkt Circle. Da auch die Kreise erst angezeigt werden sollen, sobald ein Passwort gespeichert und der Raum versperrt wurde, wird der Block gewählt. Somit sind die Kreise zunächst für den Benutzer noch „unsichtbar“



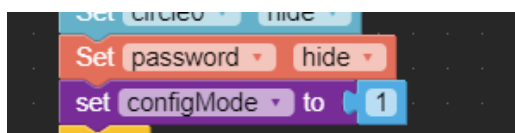
45 Setup des Programms

Nun kann mit der eigentlichen Logik begonnen werden. Die Idee dahinter ist die folgende:
Zuerst muss festgelegt werden, in welchem Modus sich die Applikation befindet. Dazu wird eine Variable configMode angelegt. Je nachdem welche Zahl diese Variable hat wird dann ein anderer Teil des Programms ausgeführt.



46 Erstellen neuer Variablen

Um eine Variable anzulegen, muss man in der Kategorie Variable auf den Button „Create Variable“ klicken. Nun kann diese Variable verwendet werden. Um ihr einen Wert zuzuweisen, wird der entsprechende Block ausgewählt. Zu Beginn soll die Variable den Wert 1 haben. Um diesen Wert zuzuweisen, wird der Block aus der Kategorie Math benötigt.

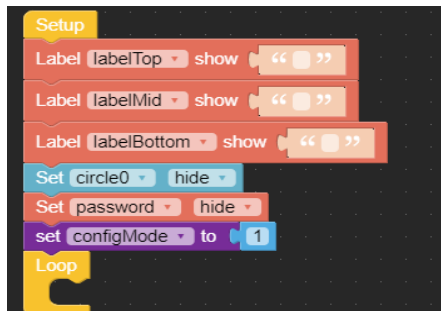


47 Wertzuweisung



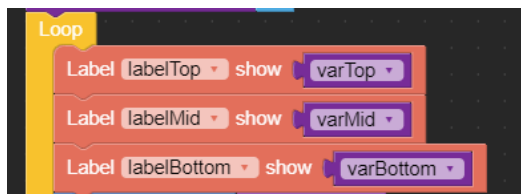
48 Loop

Nun möchten wir festlegen, dass Teile des Programms mehrfach ausgeführt werden. Dafür brauchen wir aus der Kategorie Event den Loop Block. Alles was sich innerhalb dieses Blockes befindet, wird in einer Schleife immer wieder ausgeführt.



49 Verwendung des Loop-Blocks

Natürlich soll jetzt der Bildschirm des M5stacks auch etwas anzeigen. Dazu müssen zuerst in der Kategorie Variablen die entsprechenden Variablen angelegt werden, wie bereits gezeigt, in denen die anzuzeigenden Werte dann gespeichert werden können.



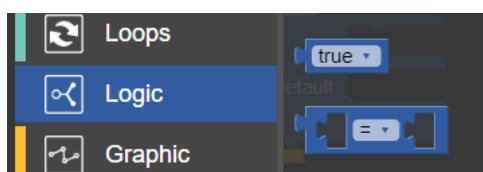
50 Anzeigen der Label

Damit diese Werte aber nun auch wirklich am Bildschirm angezeigt werden, müssen die entsprechenden Label, die zuvor festgelegt wurden, die Variablen anzeigen. Dies muss innerhalb der Schleife passieren, weil sich die Label nicht automatisch mitändern würden, wenn sich der Wert der Variablen ändert. Wird eine Variable also verändert, muss das Label neu angezeigt werden, damit auch am Bildschirm der neue Wert der Variable erscheint.



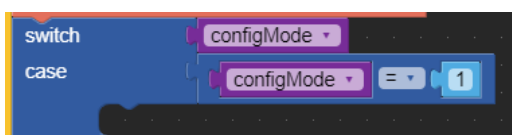
51 Switch-Case-Verzweigung

Im nächsten Schritt soll dann, abhängig von dem Wert der Variable configMode etwas anderes getan werden, wie zum Beispiel das Setzen eines neuen Passwortes oder das Versperren des Raums. Um das Szenario, dass es verschiedene Fälle gibt und in jedem Fall etwas anderes gemacht wird, umzusetzen wird der Switch Case Block benötigt. Diesen findet man in der Kategorie Logic. Beim Switch-Case wird der Wert der Variable innerhalb verschiedener **Cases** (Fälle) gesucht und bei einer Übereinstimmung die zusammenhängende Anweisung ausgeführt. Wenn man einen weiteren Case dazugeben möchte, muss man dafür einfach am Ende des Switch Case Blockes auf das Plus drücken.

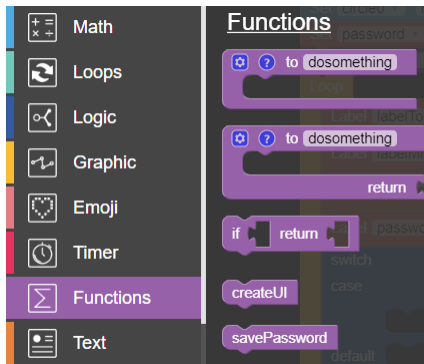


52 If-Bedingung in Kategorie Logic

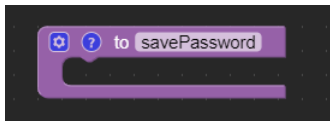
Behandelt keiner der Fälle den Wert, den die Variable hat, wird der Default Teil ausgeführt. In unserem Beispiel möchten wir, dass immer der Wert der Variable configMode untersucht wird, deshalb wird sie beim Switch angehängt. Nun müssen wir die einzelnen Fälle definieren.



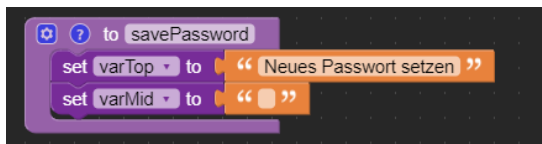
53 If Bedingung



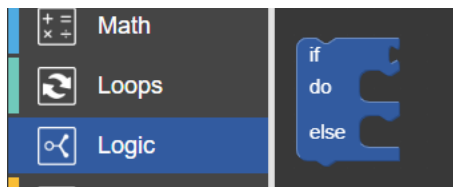
54 Erstellen einer neuen Funktion



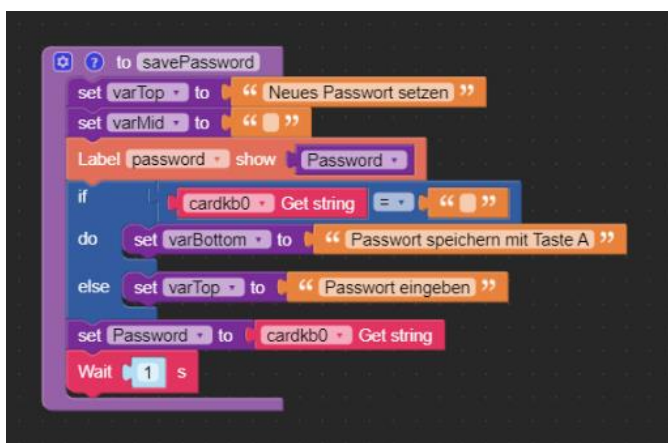
55 Funktion "savePassword" Teil 1



56 Funktion "savePassword" Teil2



57 If-Else-Verzweigung



58 Funktion "save Password" Teil 3

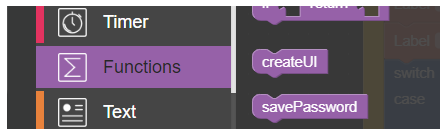
Den ersten Fall, den wir nun brauchen ist der, dass die Variable configMode den Wert 1 hat. Dafür brauchen wir eine Bedingung, die wir zum ersten Case dazugeben. Um diese Bedingung zu schreiben, findet man in der Kategorie Logic die nötigen Blöcke. Zuerst wird ein simpler Vergleichsoperator gebraucht. Dieser Fall soll ausgeführt werden, wenn die Variable configMode den Wert 1 hat.

Im Falle, dass der Wert der Variable configMode 1 ist, soll der Benutzer ein neues Passwort setzen können. Damit er dies kann, müssen wir zunächst eine Funktion mit dem Namen savePassword schreiben. Dazu braucht man aus der Kategorie Function den entsprechenden Block.

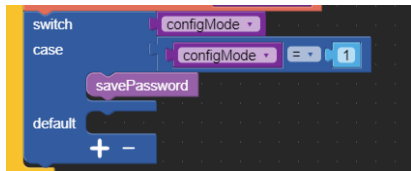
Damit der Benutzer überhaupt erfährt, dass er ein neues Passwort setzen kann, wird der Wert der Variablen entsprechend angepasst.

Um das Passwort zu speichern, muss die Taste A gedruckt werden. Diese Information soll dem Benutzer aber erst angezeigt werden, wenn er bereits mehr als ein Zeichen auf der Tastatur eingegeben hat. Um dies umzusetzen, brauchen wir eine if-else Verzweigung. Ähnlich wie beim Switch Case wird auch hierfür wieder eine Bedingung benötigt, in diesem Fall soll aber geprüft werden, ob bereits auf der Tastatur etwas eingegeben wurde. Je nachdem wird in einen der zwei Zweige gegangen und der Variable ein neuer Wert zugewiesen. Anschließend wird auch der Variable

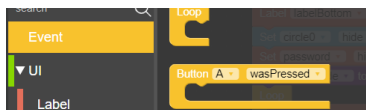
Password das auf der Tastatur eingegebene Zeichenkette zugewiesen.



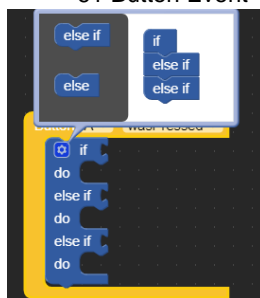
59 Funktionsaufruf



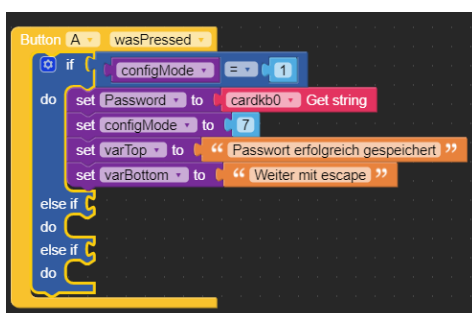
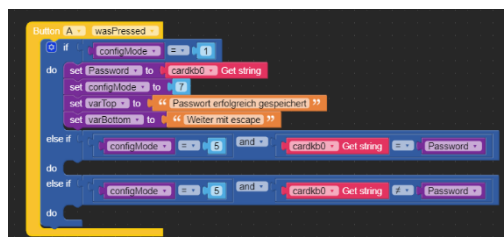
60 Funktionsaufruf Teil 2



61 Button Event



62 If-Else If-Else Verzweigung



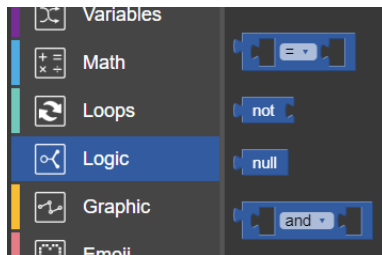
63 Button A Teil 1

Erst wenn dieser Block angelegt und richtig benannt wurde, kann man im Case die Funktion aufrufen. Ist der Wert also nun 1, wird in diesen Fall gegangen und die Funktion savePassword aufgerufen. Dabei wird dann in den entsprechenden Codeblock gesprungen und dieser ausgeführt.

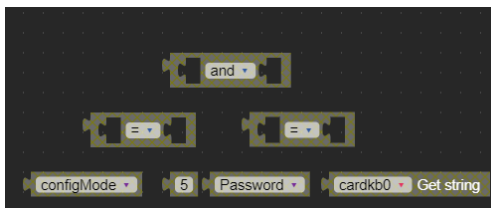
Nun wird dem Benutzer nach Eingabe einer Zeichenkette angezeigt, dass er auf Taste A drücken muss, um das Passwort auch wirklich zu speichern. Wenn der Benutzer also nun auf Taste A drückt, soll etwas passieren. Um dies umzusetzen, brauchen wir aus der Kategorie Event den passenden Block. Beim Betätigen der Taste A wird nun also ausgeführt, was sich innerhalb des Blocks befindet. Button A soll verschiedene Funktionalitäten bekommen, je nachdem in welchen Modus sich das Programm gerade befindet.

Ist der Wert der Variable 1 soll das neue Passwort gespeichert werden, ist der Wert 5 und die eingegeben Zeichenkette richtig, soll das Passwort auf erfolgreich gesetzt werden. Dafür wird eine if Verzweigung gebraucht. Zuerst müssen wir die Bedingung zusammenbauen. Ist diese Bedingung wahr, wird alles ausgeführt, was sich im do Teil des Ifs befindet. Zuerst soll die eingegeben Zeichenkette in der Variable Passwort gespeichert werden und der Wert von configMode auf 7 gesetzt werden. Außerdem werden auch den Variablen varTop und varBottom neue Werte zugewiesen. Da das Passwort Label nicht angezeigt werden soll, wird es mit hide auf unsichtbar gesetzt. Für den zweiten Teil der Funktionalität des Buttons wird wieder eine Verzweigung benötigt, in diesem Fall aber eine if-elseif Verzweigung. Diese findet man im Block Logic und man muss auf das kleine Zahnrad klicken damit man weitere else if hinzufügen kann. Nun wird die erste Bedingung für das if

64 Button A Teil 2



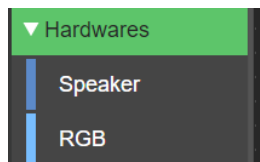
65 geschachtelte Bedingung



66 Komponenten für Bedingung



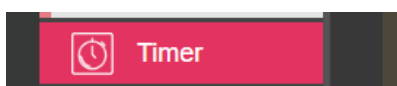
67 RGB Light und Speaker



68 Kategorie Hardware



69 Kreis anzeigen



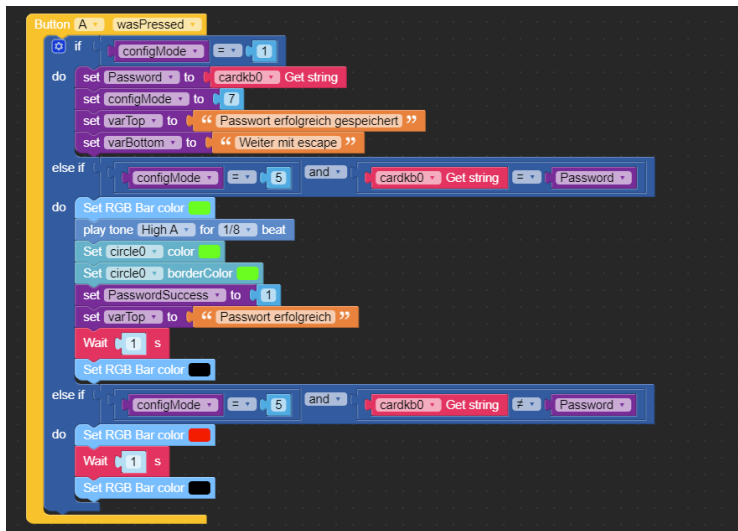
70 Timer Kategorie

zusammgebaut. Dabei soll der do Block ausgeführt werden, wenn configMode den Wert 5 hat und der auf der Tastatur eingegeben String jenem entspricht der in der Variable Passwort gespeichert wurde. Dafür braucht man eine geschachtelte if Bedingung.

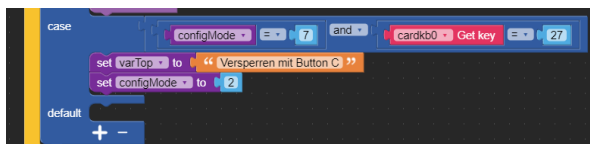
Ist sie wahr, soll der M5Stack grün leuchten und einen Ton abspielen. Dafür benötigen wir aus der Kategorie Hardware jeweils aus RGB und Speaker den passenden Block. Außerdem soll der Kreis am Bildschirm grün werden und angezeigt werden, dass das Passwort erfolgreich ist. Nach einer Sekunde Wartezeit, wofür aus der Kategorie Timer der

Block benötigt wird, soll das grüne Leuchten des M5Stacks wieder erlöschen, wofür es ausreicht die Farbe auf Schwarz zu setzen.

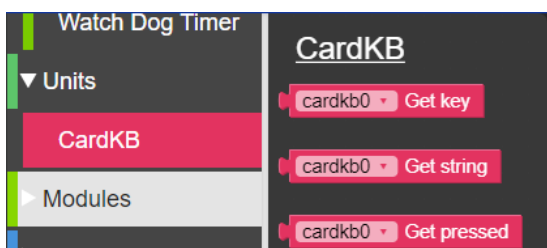
Befindet sich das Programm im ConfigMode 5, stimmt jedoch das eingegebene Passwort nicht mit dem gespeicherten Passwort überein, dann soll der M5Stack rot aufleuchten. Dafür wird für den else IF Zweig die passende Bedingung gebraucht. Auch diese wird mit dem AND Block und zwei einzelnen Bedingungen zusammengebaut.



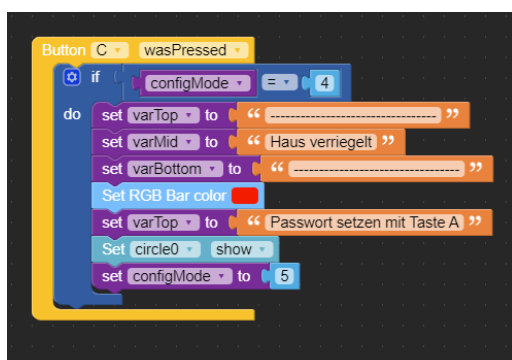
71 Implementierung Button A



72 Implementierung der Escape - Taste



73 CardKB Blocks



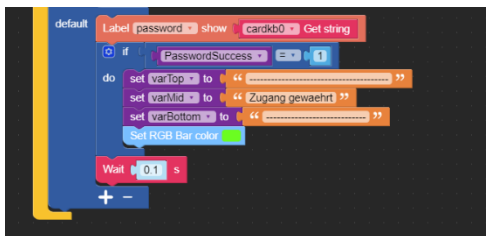
74 Button C

Danach geht es weiter im Hauptblock. Für den nächsten Case wird überprüft ob der Wert der configMode Variable ist und ob auf der Tastatur zuletzt das Escape Zeichen gedrückt wurde. Um dies zu überprüfen, brauchen wir wieder eine zusammengesetzte Bedingung. Damit festgestellt werden kann, ob zuletzt das Escapezeichen gedrückt wurde, brauchen wir aus dem Block CardKB den Block getKey. Dieser gibt uns den Wert der zuletzt gedrückten Taste. Die Escapetaste hat den Wert 27 demnach vergleichen wir, ob zuletzt die Taste mit dem Wert 27 gedrückt wurde. Ist beides der Fall gehen wir in den Block rein und dieser wird ausgeführt. Die Variable varTop soll dem Benutzer nun anzeigen, dass mit Button C das Haus verriegelt werden kann. Zusätzlich soll der Wert von configMode auf 4 erhöht werden. Dies wird für den Versperrmechanismus in den folgenden Arbeitspaketen benötigt.

Damit bei Drücken des Button C auch wirklich etwas geschieht, müssen wir den entsprechenden Block anlegen.

Dies funktioniert genauso wie bei Button A. Innerhalb des Blocks benötigen wir eine if-else Verzweigung. Momentan soll nur etwas passieren, wenn der Wert der ConfigMode Variable genau 4 ist. Dies ist also unsere If Bedingung. Da am Bildschirm auch angezeigt werden soll, dass das Haus verriegelt wurde, müssen die entsprechenden variablen angepasst werden. Außerdem soll der Benutzer ein visuelles Signal bekommen. Um den M5Stack rot leuchten zulassen,

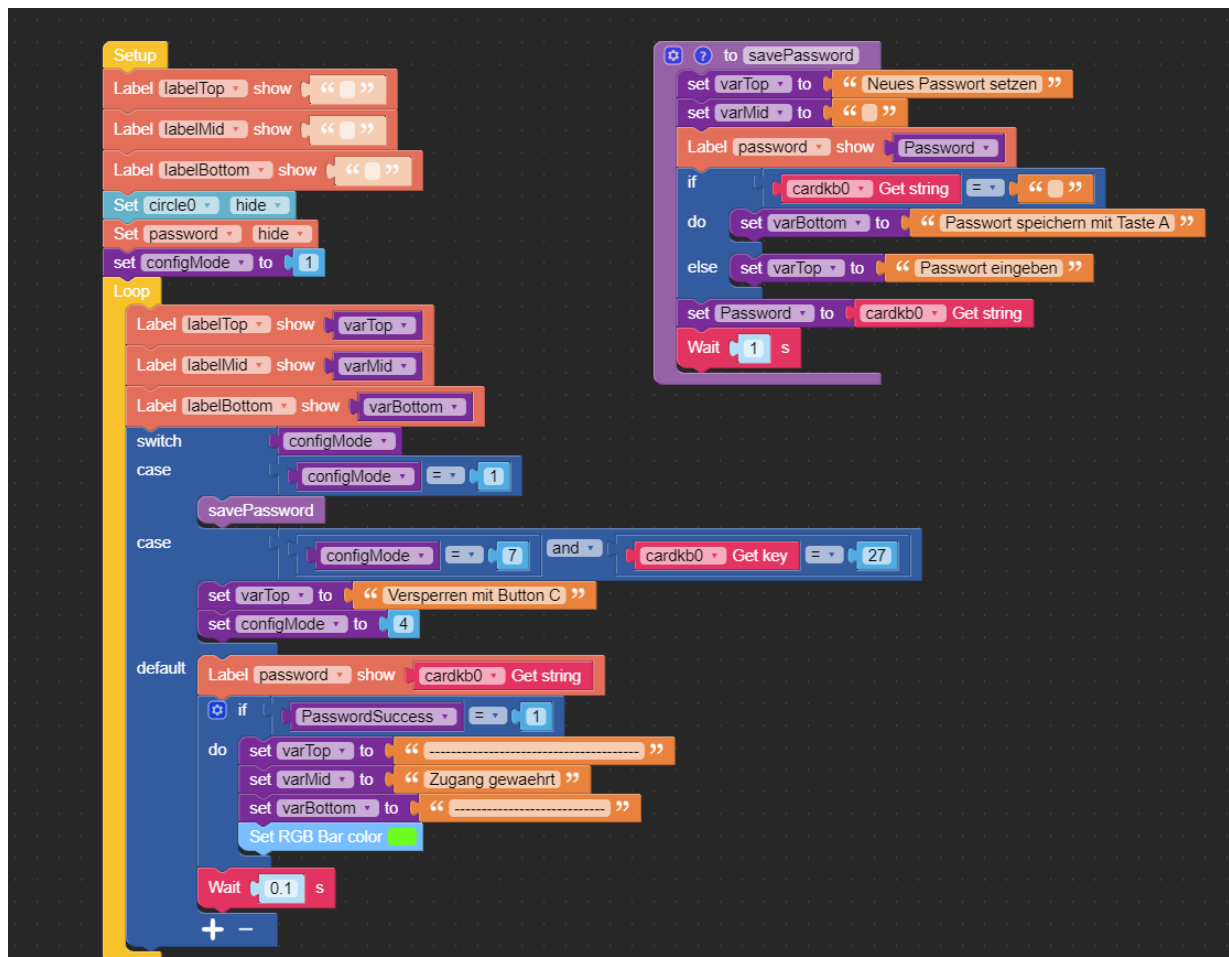
brauchen wir wieder den RGB Block aus der Kategorie Hardwares. Nach einer Sekunde soll das rote Leuchten zum Erlöschen kommen, deswegen wird mit Hilfe des Timer Blocks eine Sekunde gewartet und dann RGB auf schwarz eingestellt. Nun soll der Benutzer die Möglichkeit haben das Haus zu entsperren. Dafür soll ihm die Möglichkeit Passwort setzen angezeigt werden. Außerdem soll der zuvor angelegte Kreis rot angezeigt werden, da das Haus ja noch verriegelt ist. Dies kann mit dem Block Circle in der Kategorie UI gemacht werden. Schließlich soll der Modus auf 5 geändert werden.

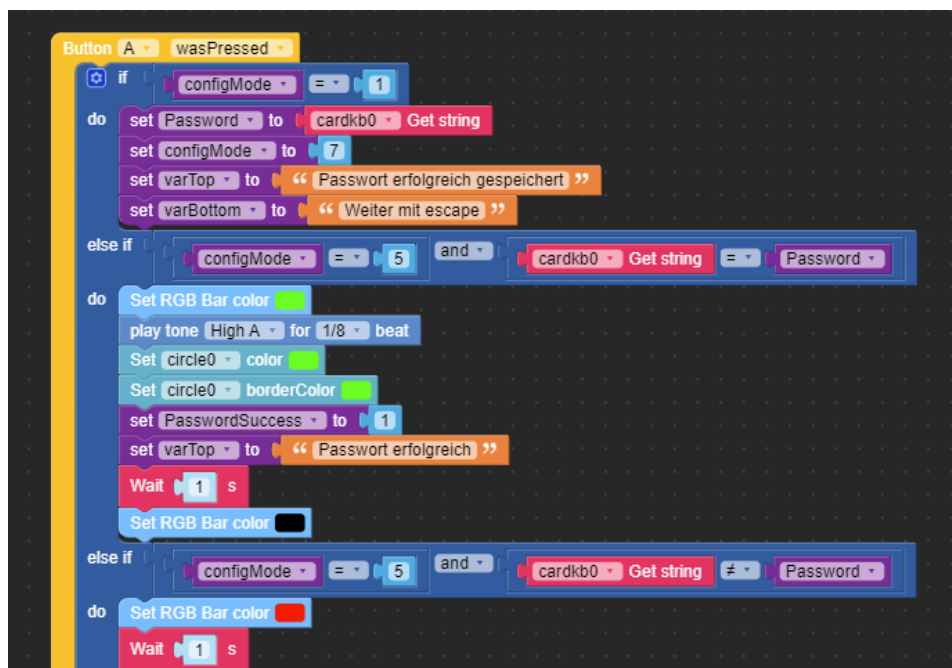


75 Default Block

Weiter geht es im Default Block der Switch-Case Verzweigung. Dieser Block wird ausgeführt, wenn keiner der davor vorhandenen Cases eintritt. Hier wird geprüft, ob der Wert der Variable passwordSuccess wirklich 1 ist. Dies ist nur der Fall, wenn das eingegeben Passwort mit dem gespeicherten übereinstimmt. Dann soll das Haus entriegelt werden und der M5Stack grün aufleuchten.

Vollständiger Code





76 Hauptteil



77 Button A und C

5.2. Verwendung der RFID Komponente

Am Ende des zweiten Arbeitspaketes soll es zusätzlich die Möglichkeit geben, eine Karte einzulesen. Um dies zu verwirklichen, benötigt man eine weitere Unit, nämlich die RFID Komponente. Außerdem werden zwei neue Funktionen angelegt werden müssen, Button B wird in Verwendung genommen und es wird zu Anpassungen im restlichen Code kommen.

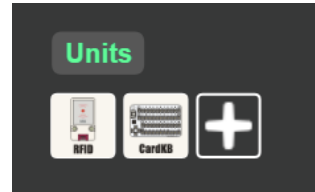
5.2.1. Hardware



79 RFID Komponente
und Grove Hub



78 RFID
Komponente

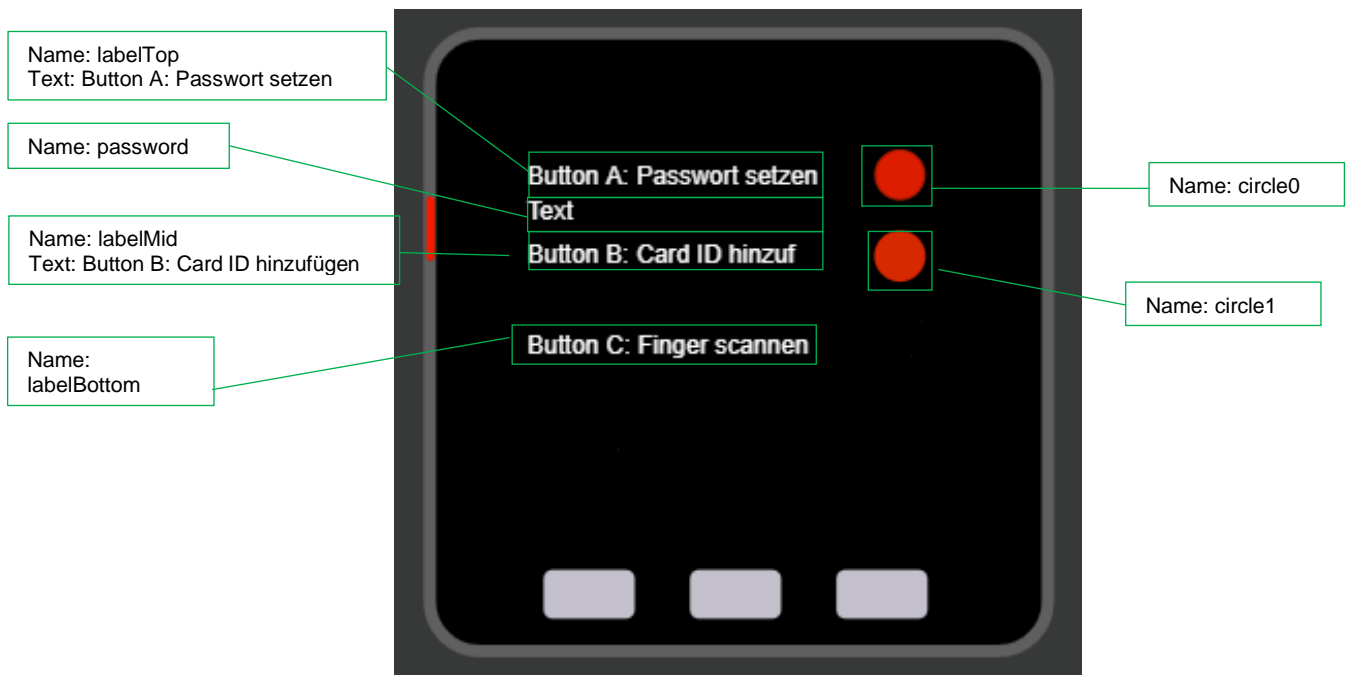


80 Units in UIFlow

Dazu muss als erstes die RFID Komponente mit Hilfe eines Grove Kabels an einen freien Steckplatz am Grove Hub angeschlossen werden und in UI Flow unter Units hinzugefügt werden.

5.2.2. Benutzeroberfläche

Nun muss ein weiterer Kreis auf Höhe des Labels labelMid hinzugefügt werden. Außerdem kann der Text des labelMids angepasst werden.



5.2.3. Programmcode

Zum Speichern der Karte benötigen wir eine eigene Funktion. Diese kann in der Kategorie Function erstellt werden und wir geben ihr den Namen saveCard.

Innerhalb der Funktion möchten wir, wenn eine Karte an den Sensor gelegt wurde, ihre Nummer speichern, ansonsten dem Benutzer mitteilen, dass er eine Karte an den Scanner legen muss.

Um diesen Sachverhalt umzusetzen, brauchen wir wieder eine if-else Verzweigung.

In der If Bedingung prüfen wir, ob eine Karte an den Sensor gelegt wurde und ob die UID dieser Karte nicht leer ist.

Da wir zwei Sachen auf prüfen möchten, brauchen wir wieder eine geschachtelte Bedingung, die mit AND zwei einzelne verbindet.

Dieses Konstrukt findet man im Block Logic.

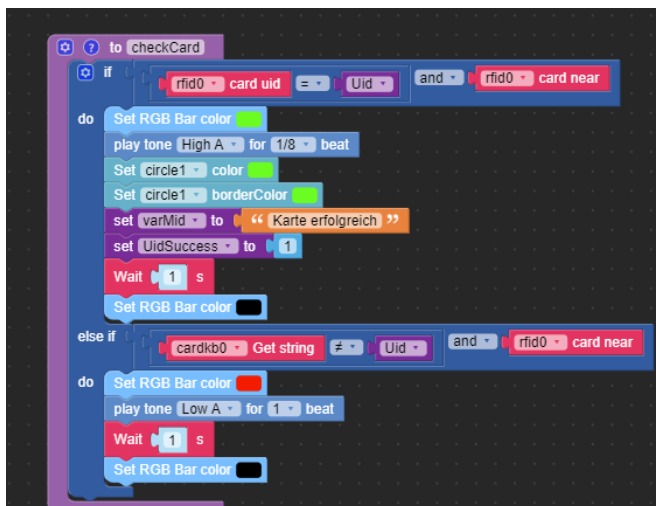


82 Funktion "saveCard"

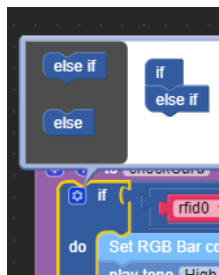


83 RFID Blocks

Für die erste Bedingung brauchen wir die Karten UID. Diesen Block findet man in der Kategorie Units unter RFID. Da wir kontrollieren möchten, dass die Nummer nicht einfach leer ist, holen wir aus der Kategorie text eine leere Zeichenkette. Für den zweiten Teil der Bedingung brauchen wir aus der Kategorie Unit den Block card near. Ist eine Karte am Scanner angelegt wird dieser Block zu true ausgewertet. Sind beide Bedingungen wahr, wird in den do Teil der IF Verzweigung reingegangen. Dort wird der Modus auf 3 gesetzt, sowie die Kartennummer in die Variable uid gespeichert. Zuvor muss diese variable natürlich angelegt werden. Dies kann in der Kategorie Variables gemacht werden.



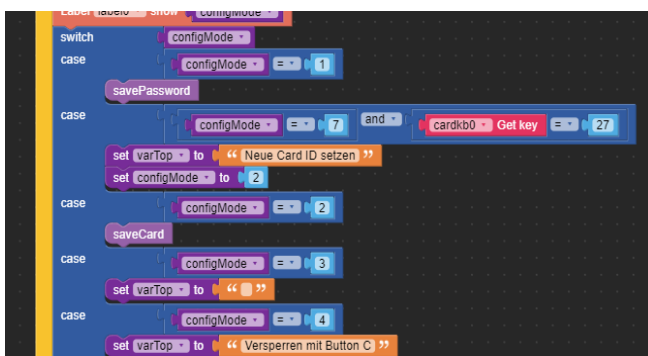
84 Funktion "checkCard"



85 If-Else If-Verzweigung



86 Button B

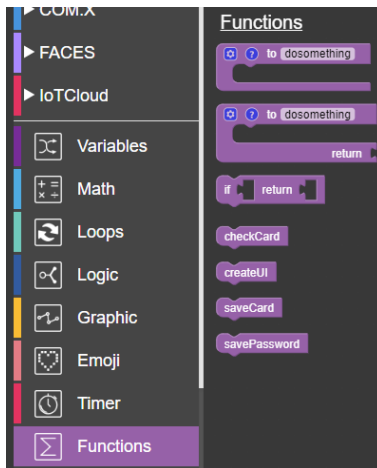


87 Erweiterung Switch-Case-Verzweigung

Zusätzlich zu dieser Funktion brauchen wir auch eine Funktion, die überprüft, ob eine an den Scanner gelegte Karte mit der gespeicherten Karte übereinstimmt. Dafür legen wir eine neue Funktion mit dem Namen checkCard an. Innerhalb dieser Funktion brauchen wir eine If-Else If- Verzweigung. Diese kann in der Kategorie Logic nach Auswahl des If-else Blocks mit dem kleinen blauen Zahnrad erstellt werden. Für die If Bedingung benötigen wir eine zusammengesetzte if Bedingung. Denn es soll überprüft werden, dass einerseits eine Karte in der Nähe ist und andererseits, dass die UID der Karte mit der gespeicherten UID übereinstimmt. Ist dies der Fall, soll der M5Stack grün aufleuchten und einen Ton abspielen. Außerdem soll der Kreis für die Karte am Bildschirm grün werden und Karte erfolgreich ausgegeben werden. Nach einer Sekunde soll das Leuchten wieder erlöschen. Im Else-if Zweig soll der Fall abgedeckt werden, dass zwar eine Karte angelegt wurde, diese jedoch nicht mit der gespeicherten Karte übereinstimmt. Dafür brauchen wir auch wieder eine zusammengesetzte Bedingung.

Weil die angelegte Karte erst bei Betätigung der Taste B gespeichert werden soll, brauchen wir auch Funktionalität im Button B. Dafür müssen wir aus der Kategorie Events den benötigten Block holen. Im Moment soll bei Button B nur der Modus auf 4 geändert werden, wenn er im Moment des Drückens 3 ist. In den folgenden Arbeitspaketen wird hier noch mehr Funktionalität dazukommen.

Nun muss der Hauptteil des Codes entsprechend angepasst werden. Nachdem das Passwort gespeichert und Escape gedrückt wurde, soll man nun nicht direkt das Haus verriegeln können, sondern es soll angezeigt werden, dass man eine neue Karte scannen kann. Dafür muss die Zeichenkette, die der Variable varTop zugewiesen wird, geändert werden.



88 neue Funktion erstellen

Im Fall, dass der Modus 2 ist, soll nun die neue Funktion saveCard aufgerufen werden. Den Methodenaufruf findet man in der Kategorie Function. Auch der Fall, dass der Modus 4 ist, muss angepasst werden. Hier soll nun angezeigt werden, dass das Haus mit Button C verriegelt werden kann.



89 Erweiterung If Bedingung

Auch die IF Bedingung im Default Block der Switch-Case-Verzweigung muss erweitert werden. Das Haus soll nun nicht mehr bereits nach der korrekten Eingabe des Passwortes entriegelt werden, sondern erst, wenn auch die gespeicherte Karte gescannt wurde.



90 Erweiterung Button C

Auch im Block der Taste C wird eine Variable hinzugefügt sowie der Kreis angezeigt.

Vollständiger Code

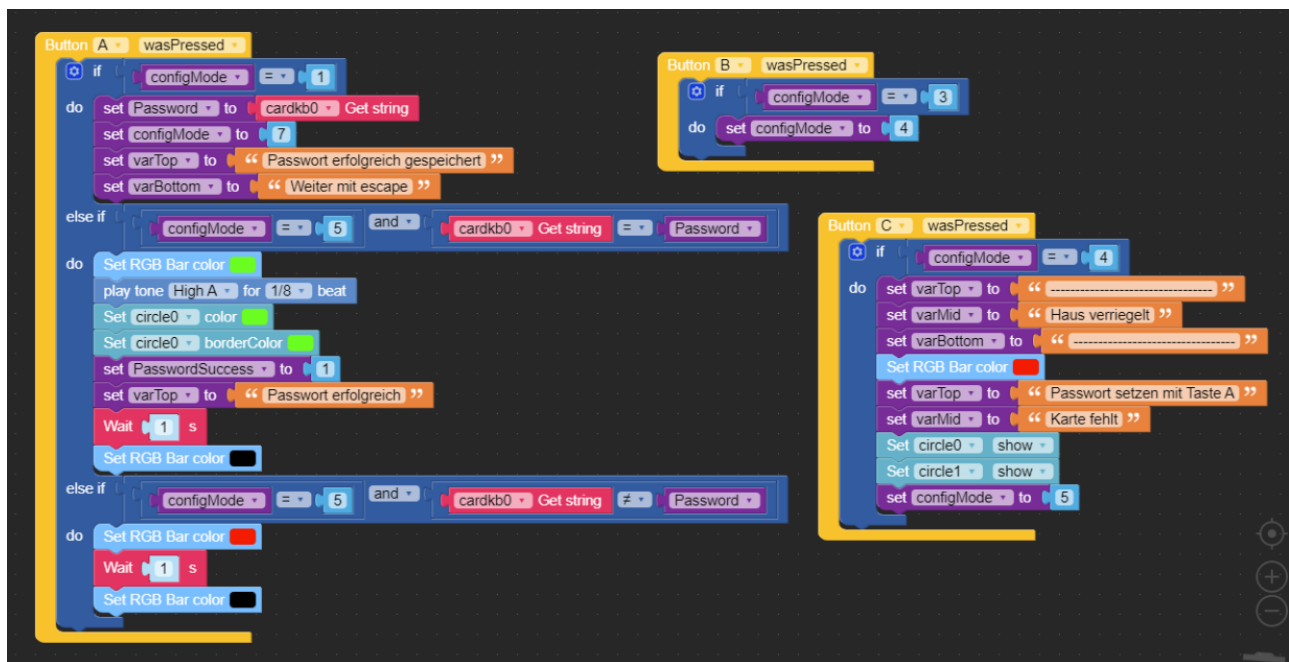
```

to saveCard
  if
    rfid0 card uid = " " and rfid0 card near
  do
    set configMode to 3
    set uid to rfid0 card uid
    set varTop to "Karte scannen"
    set varMid to " "
    set varBottom to "Scan starten mit Taste B"
  else
    set varMid to " "
    set varBottom to "Karte bitte beim Scanner anlegen"

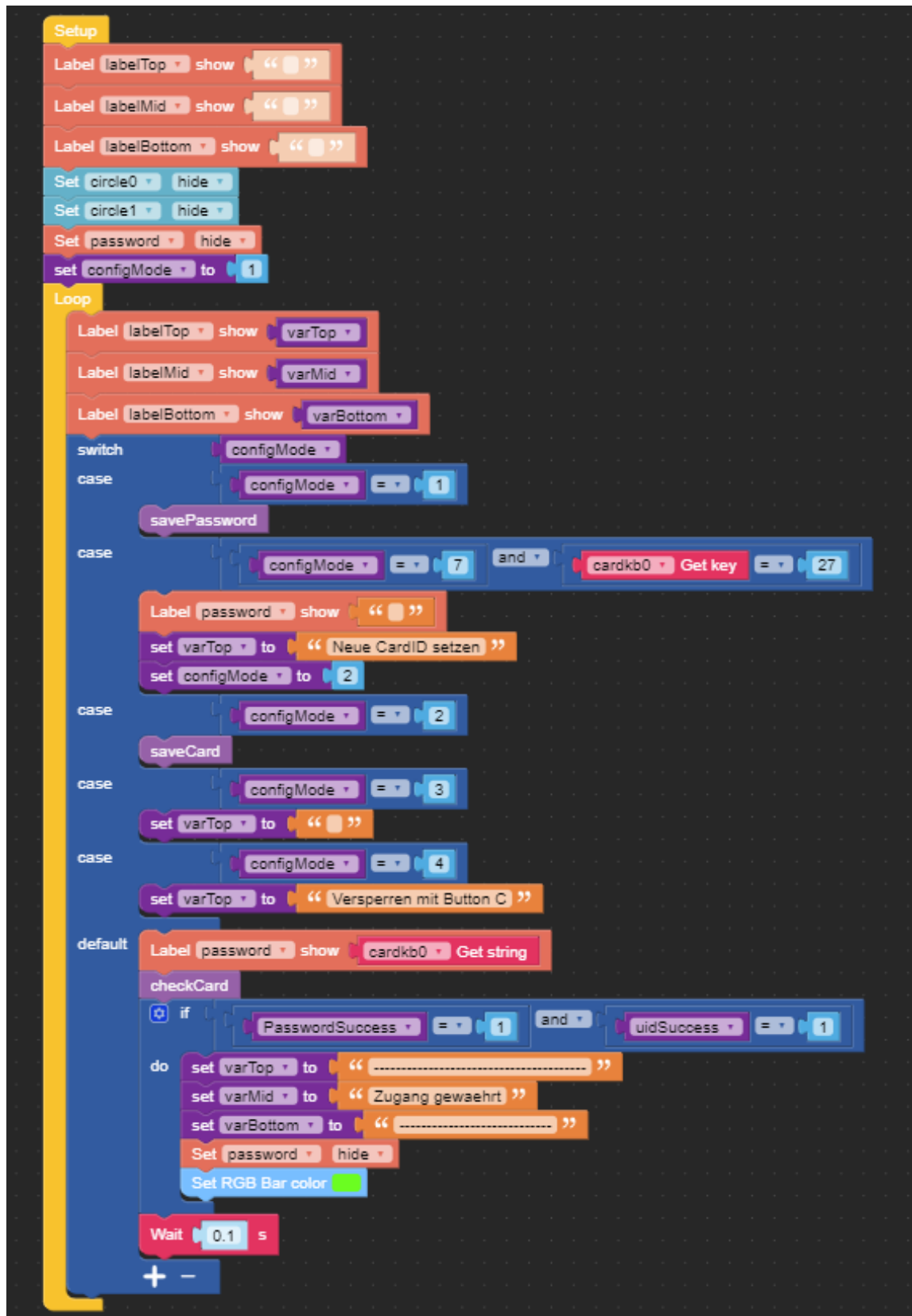
to savePassword
  set varTop to "Neues Passwort setzen"
  set varMid to " "
  Label password show Password
  if
    cardkb0 Get string = " "
  do
    set varBottom to "Passwort speichern mit Taste A"
  else
    set varTop to "Passwort eingeben"
  set Password to cardkb0 Get string
  Wait 1 s

to checkCard
  if
    rfid0 card uid = uid and rfid0 card near
  do
    Set RGB Bar color
    Set circle1 color
    Set circle1 borderColor
    play tone High A for 1/8 beat
    set varMid to "Karte erfolgreich"
    set uidSuccess to 1
    Wait 1 s
    Set RGB Bar color
  else if
    rfid0 card uid ≠ uid and rfid0 card near
  do
    Set RGB Bar color
    play tone Low A for 1/8 beat
    Wait 1 s
    Set RGB Bar color
  
```

91 benötigte Funktionen

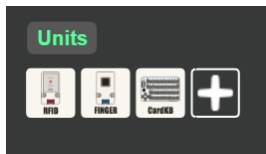


92 Button A, B und C



93 Hauptteil des Programms

5.3. Verwendung der Finger Komponente



94 Units in UIFlow

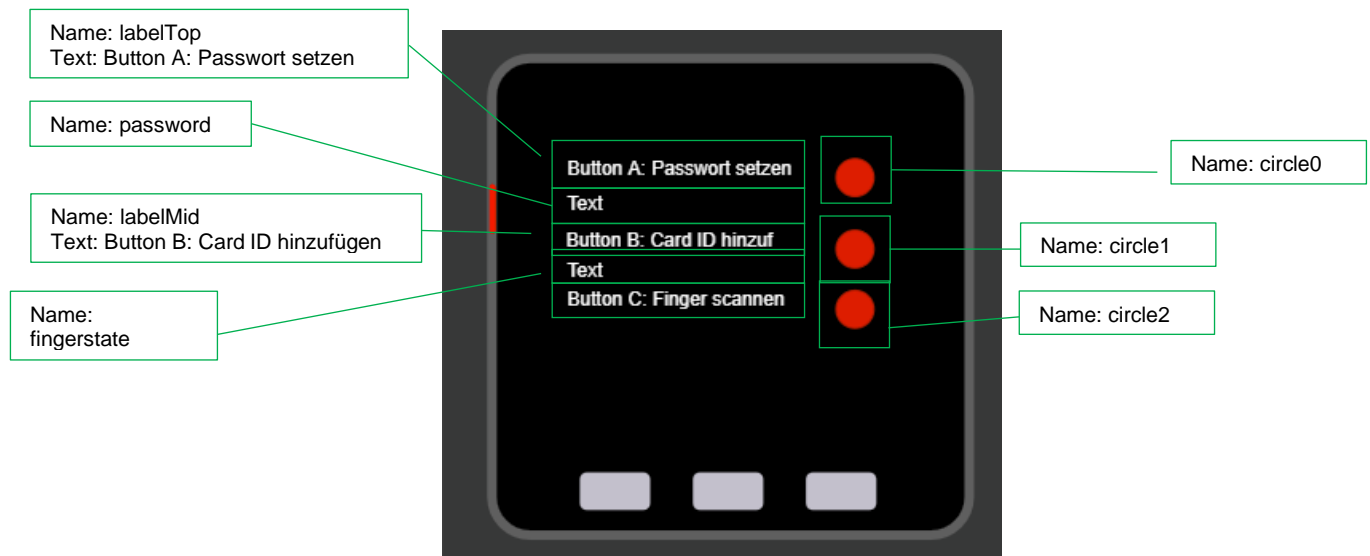


95 Finger Komponente

Nach dem dritten Arbeitspaket soll der Benutzer auch die Möglichkeit haben, seinen Finger zu scannen. Dafür müssen wir die Finger Unit mit einem Grove Kabel an Port C des M5Stacks anschließen und dann in UIFlow hinzufügen.

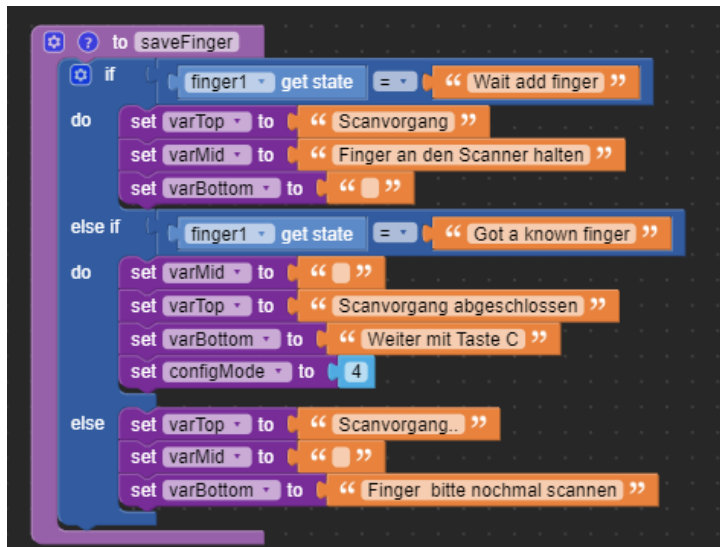
5.3.1. Benutzeroberfläche

Danach muss der Bildschirm entsprechend angepasst werden.

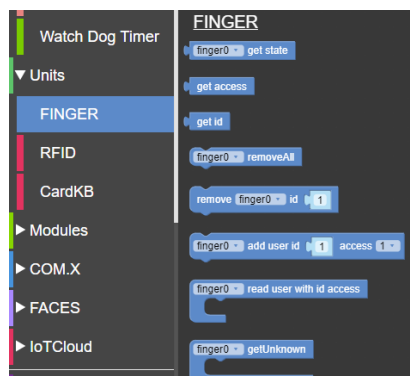


96 Benutzeroberfläche

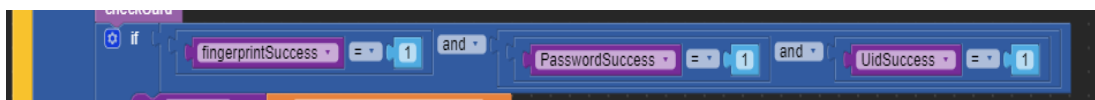
5.3.2. Programmcode



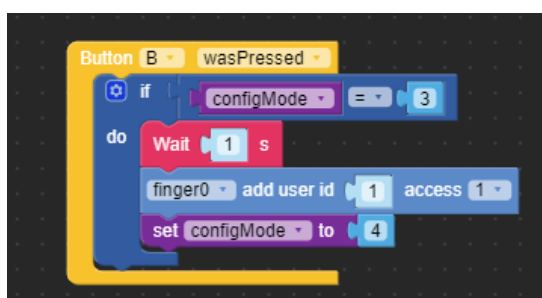
97 Funktion "saveFinger"



98 Fingerblöcke



99 erweiterte Bedingung



100 Erweiterung Button B

Zuerst muss eine Funktion mit dem Namen saveFinger erstellt werden. Hier wird eine If-Else Verzweigung mit einer weiteren else if Bedingung gebraucht. Die Blöcke für den Fingerscan findet man in der Kategorie Units unter Finger. Im Hauptblock des Programms muss nun der Falls, dass configMode den Wert 4 hat, angepasst werden. Hier wird die neu erstellte Funktion saveFinger aufgerufen.

Auch im Default Block müssen wir nun die Bedingung der If Verzweigung anpassen. Das Haus soll erst dann entriegelt werden, wenn alle drei Mechanismen funktioniert haben. Deshalb muss die Bedingung weiter verschachtelt werden, damit auch überprüft wird, ob die Variable fingerprintSuccess den Wert 1 hat.

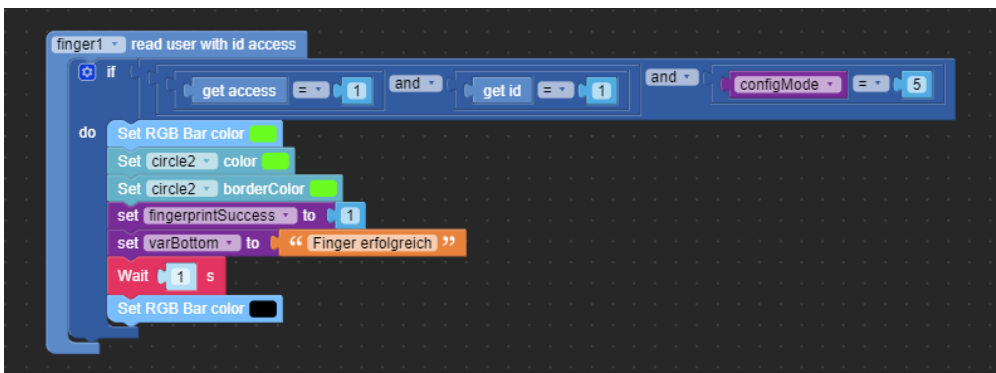
Auch der Block von Button B muss angepasst werden. Hier gehört eine weitere Zeile hinzugefügt.

Zwei weitere Code Blöcke müssen noch hinzugefügt werden.



101 Funktion "getUnknown"

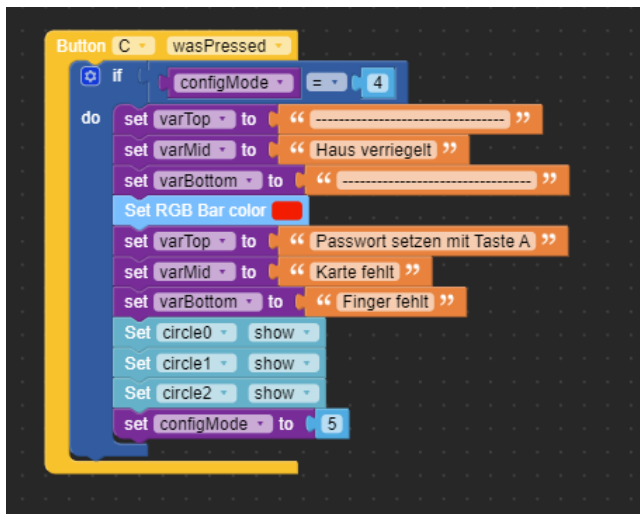
Der Block getUnknown, in dem das Verhalten bei Scan eines unbekannten Fingers festgelegt wird, kann in der Kategorie Unit gefunden werden.



102 Funktion "read User with id access"

Auch der zweite Block kann in dieser Kategorie gefunden werden. Hier wird programmiert, was passieren soll, wenn ein bereits gespeicherter Finger, erkannt wird. Wird ein gespeicherter Finger auf die Komponente gelegt und ist der Wert der configMode

Variable 5, also der Modus, in welchem ein Entsperren möglich sein soll, leuchtet der m5Stack grün und es wird angezeigt, dass der Finger erfolgreich war



Außerdem muss der Block von Button C noch um das Anzeigen des Kreises sowie des Textes, dass der Finger fehlt, erweitert werden.

103 Erweiterung Button C

Vollständiger Code

```

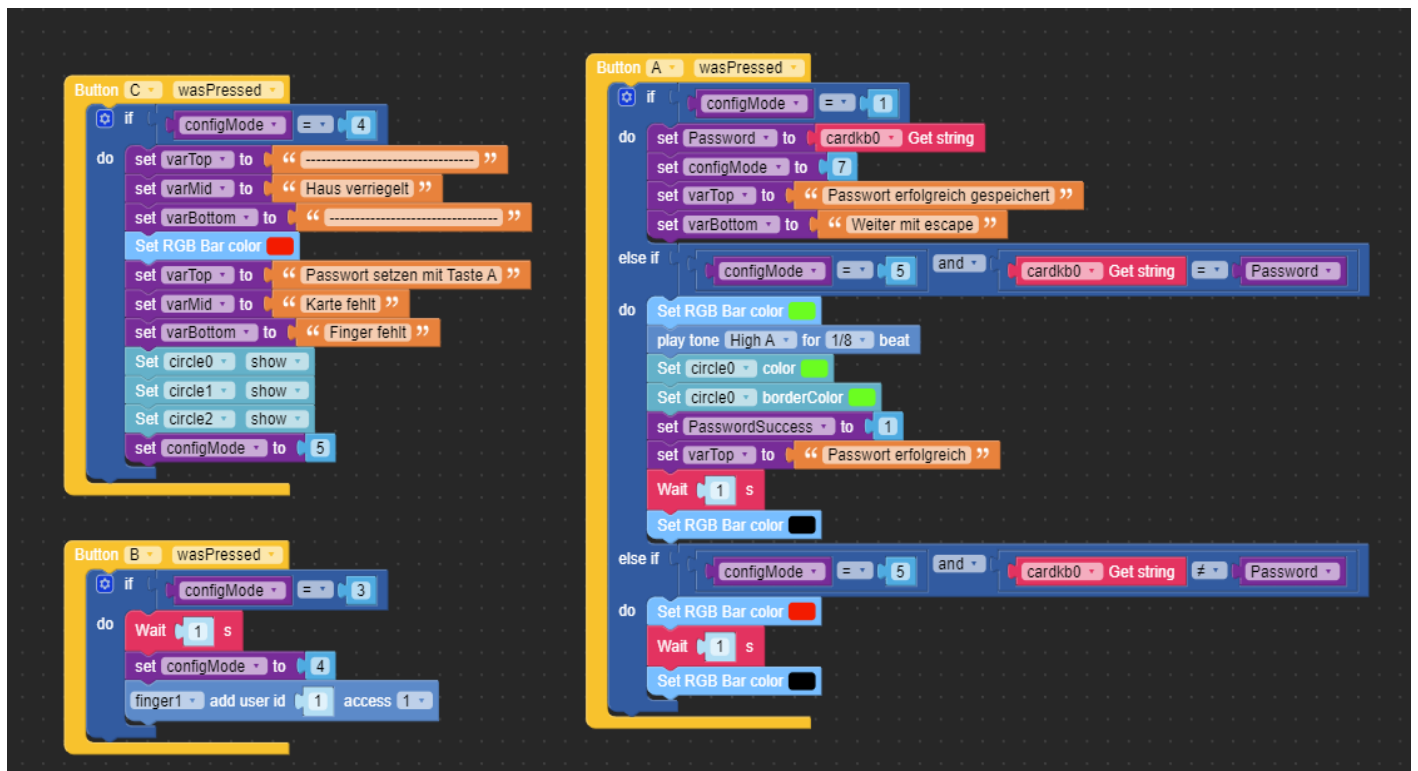
Setup
  Label labelTop show " "
  Label labelMid show " "
  Label labelBottom show " "
  Set circle0 hide
  Set circle1 hide
  Set circle2 hide
  Set password hide
  set configMode to 1

Loop
  Label labelTop show varTop
  Label labelMid show varMid
  Label labelBottom show varBottom
  switch configMode
  case configMode = 1
    savePassword
  case configMode = 7 and cardkb0 Get key = 27
    Label password show " "
    set varTop to "Neue CardID setzen"
    set configMode to 2
  case configMode = 2
    saveCard
  case configMode = 3
    saveFinger
    set varTop to " "
  case configMode = 4
    set varTop to "Versperren mit Button C"
  default
    Label password show cardkb0 Get string
    checkCard
    if PasswordSuccess = 1 and uidSuccess = 1 and fingerprintSuccess = 1
    do
      set varTop to " "
      set varMid to "Zugang gewaehrt"
      set varBottom to " "
      Set password hide
      Set RGB Bar color
    Wait 0.1 s
  
```

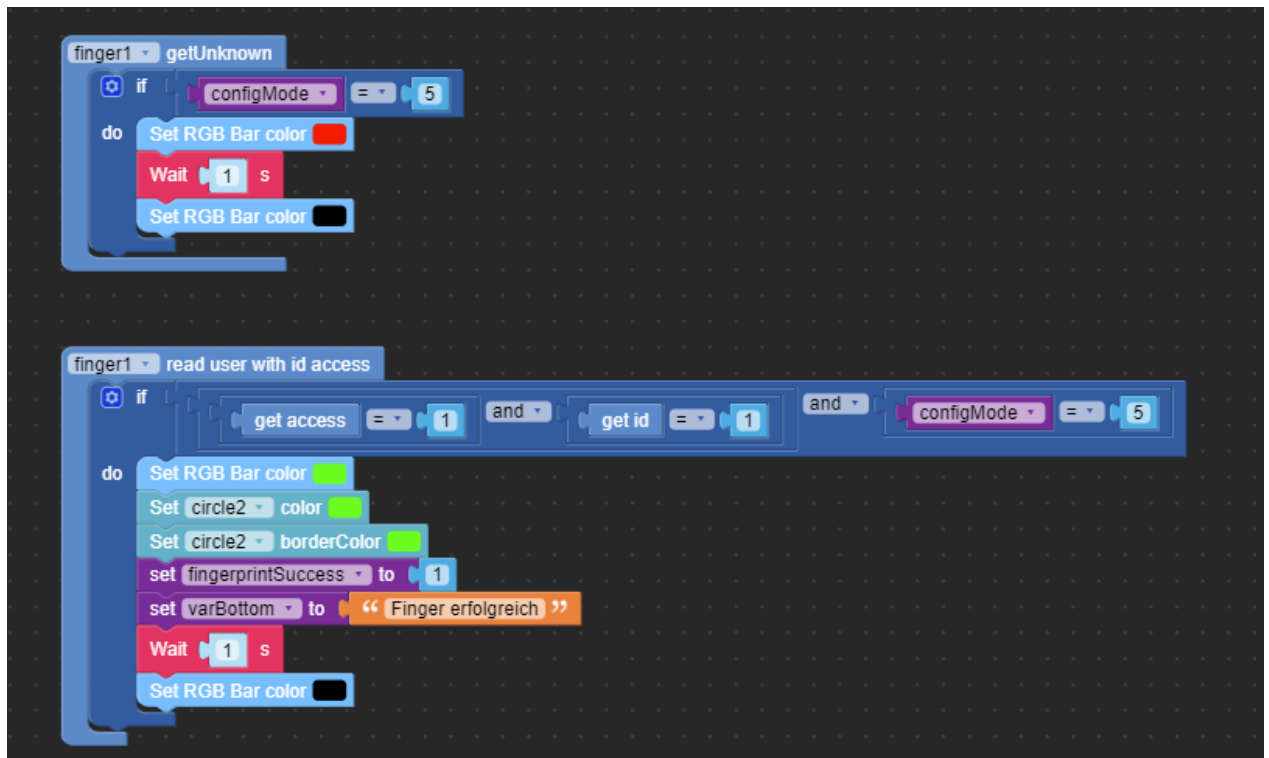
104 Hauptteil des Programms



105 verwendete Funktionen



106 Implementierung der Tasten



107 Implementierung der Finger-Unit

5.4. Bewegungssensor und RGB Light

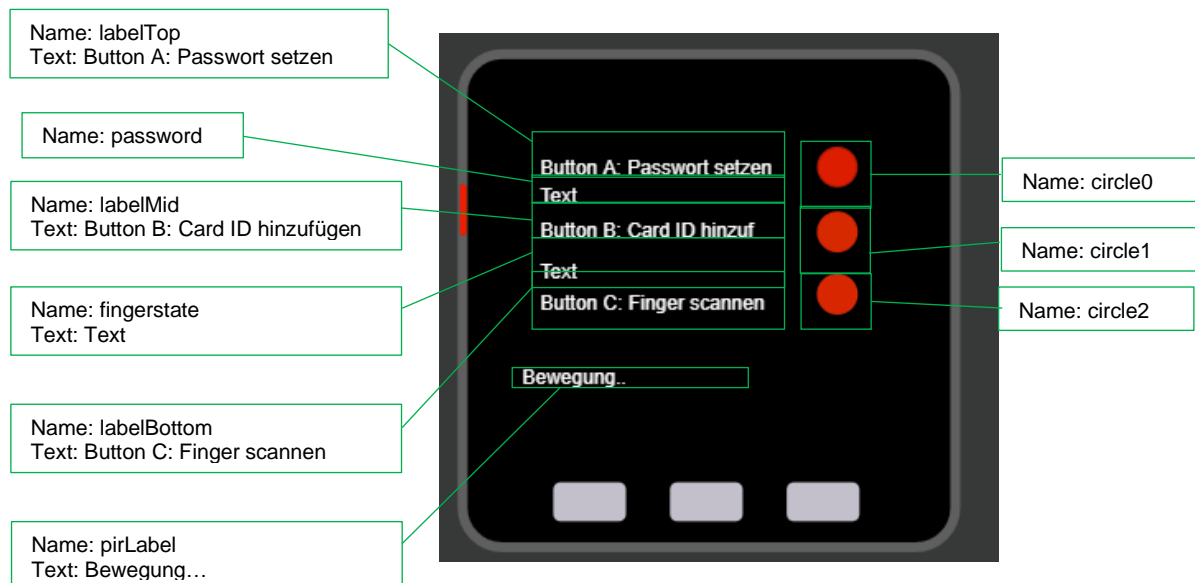
5.4.1. Hardware



108 Motion Sensor
Komponente

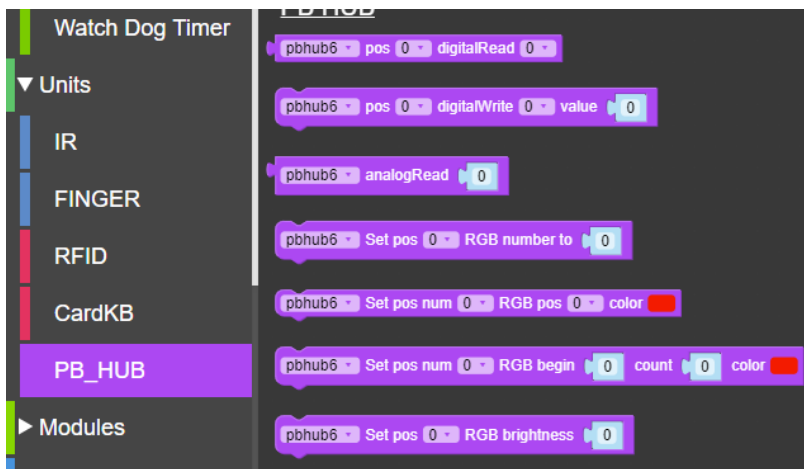
Im nächsten Schritt sollen auch Bewegung erkannt werden. Zusätzlich soll der bestehende Code so erweitert werden, dass bei erfolgreicher Eingabe einer der Mechanismen auch die RGB Light Komponente leuchtet. Dafür benötigen wir den Pb.Hub sowie den Motion Sensor und die RGB Light Komponente und drei Grove Kabel. Wichtig bei der Verwendung des Pb.Hub ist, Die Motion-Sensor Unit mit dem angeschlossenen Grove-Kabel an den Pb.Hub anschließen. (Achtung: Es ist unbedingt notwendig, dass diese Unit an den Steckplatz 0 angeschlossen wird. Die Steckplätze sind alle auf der Unit nummeriert.) Die RGB Light Komponente wird an Steckplatz Nummer 1 angeschlossen.

5.4.2. Benutzeroberfläche

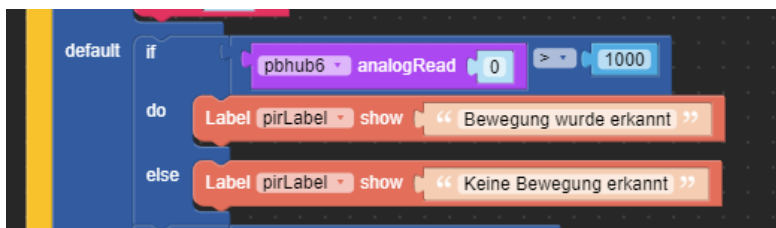


Auch die Benutzeroberfläche muss angepasst werden. Für dieses Arbeitspaket benötigen wir ein weiteres Label unterhalb von LabelBottom mit dem Namen pirLabel.

5.4.3. Programmcode



110 Pb.hub Blöcke



109 Erweiterung um Bewegungserkennung

Um die Funktionalität auch im Code zu implementieren, brauchen wir eine weitere if-else Verzweigung im Default Block. Die Zeile findet man in der Kategorie Units bei Pb.hub. Durch diese „if-Abfrage“ kontrollieren wir, ob der Wert, der vom Steckplatz 0 am Pb.Hub (in unserem Fall vom Motion-Sensor) erhalten wird, größer als 1000 ist oder nicht. Wenn er größer als 1000 ist, so wurde ein Bewegung erkannt und wir wollen dementsprechend eine Meldung anzeigen lassen. Sollte der Wert unter 1000 liegen, so wissen wir, dass es zu keiner Bewegung kam – auch hier soll eine passende Meldung sichtbar werden.

Vollständiger Code

```

to saveFinger
  if finger1 get state = "Wait add finger"
  do
    set varTop to "Scanvorgang"
    set varMid to "Finger an den Scanner halten"
    set varBottom to ""
  else if finger1 get state = "Got a known finger"
  do
    set varMid to ""
    set varTop to "Scanvorgang abgeschlossen"
    set varBottom to "Weiter mit Taste C"
    set configMode to 4
  else
    set varTop to "Scanvorgang"
    set varMid to ""
    set varBottom to "Finger bitte nochmal scannen"

to saveCard
  if rfid0 card uid = "" and rfid0 card near
  do
    set configMode to 3
    set uid to rfid0 card uid
    set varTop to "Karte scannen"
    set varMid to ""
    set varBottom to "Scan starten mit Taste B"
  else
    set varMid to ""
    set varBottom to "Karte bitte beim Scanner anlegen"

to savePassword
  set varTop to "Neues Passwort setzen"
  set varMid to ""
  Label password show Password
  if cardkb0 Get string = ""
  do
    set varBottom to "Passwort speichern mit Taste A"
  else
    set varTop to "Passwort eingeben"
  set Password to cardkb0 Get string
  Wait 1 s

to checkCard
  if rfid0 card uid = uid and rfid0 card near
  do
    Set RGB Bar color
    Set circle1 color
    Set circle1 borderColor
    pbhub0 Set pos num 1 RGB begin 1 count 1 color
    play tone High A for 1/8 beat
    set varMid to "Karte erfolgreich"
    set uidSuccess to 1
    Wait 1 s
    Set RGB Bar color
  else if rfid0 card uid != uid and rfid0 card near
  do
    Set RGB Bar color
    play tone Low A for 1/8 beat
    Wait 1 s
    Set RGB Bar color
  
```

111 RGB Komponente in Funktionen

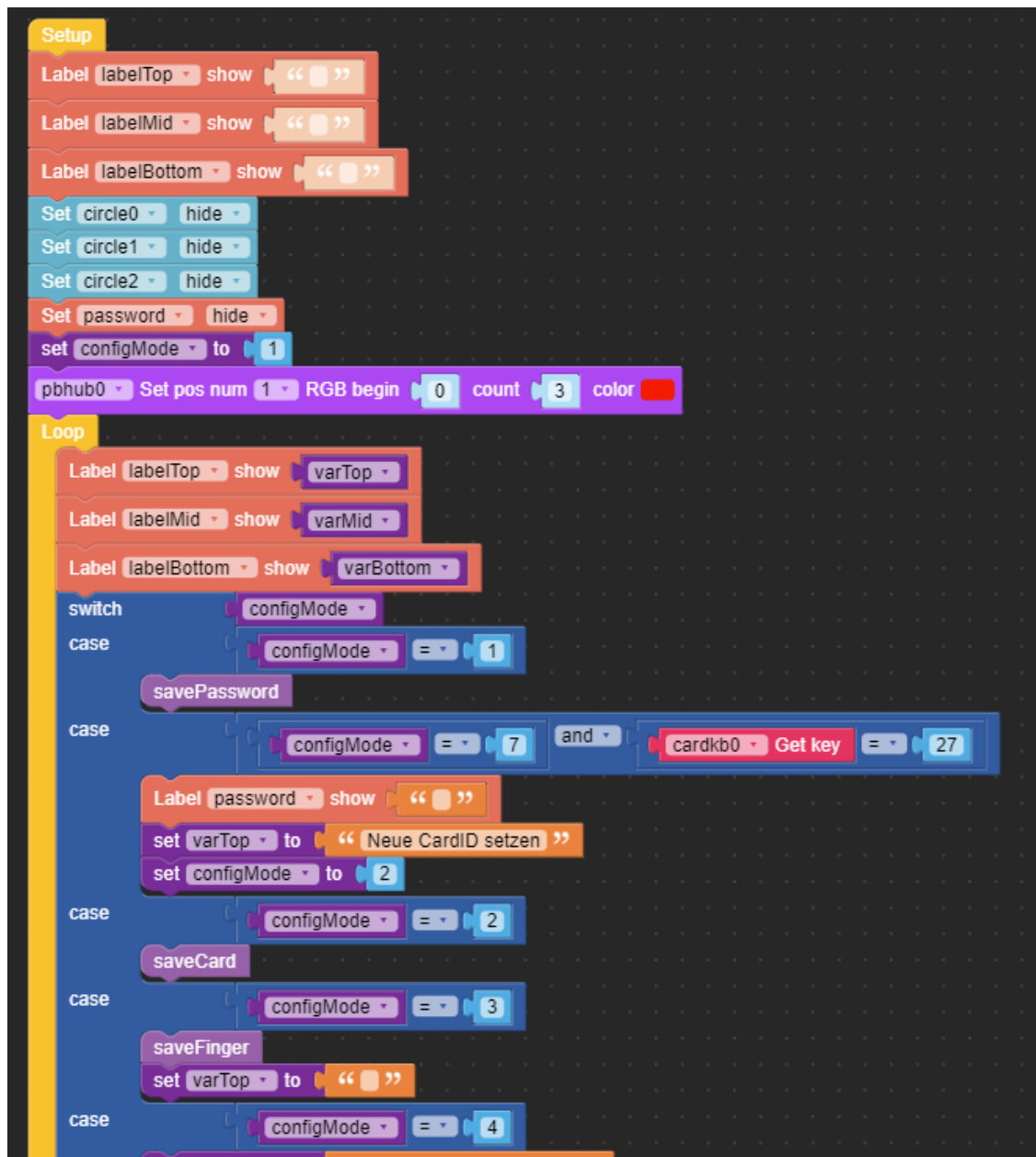
```

Button C wasPressed
  if configMode = 4
  do
    set varTop to ""
    set varMid to "Haus verriegelt"
    set varBottom to ""
    pbhub0 Set pos num 1 RGB begin 0 count 3 color
    Set RGB Bar color
    set varTop to "Passwort setzen mit Taste A"
    set varMid to "Karte fehlt"
    set varBottom to "Finger fehlt"
    Set circle0 show
    Set circle1 show
    Set circle2 show
    set configMode to 5

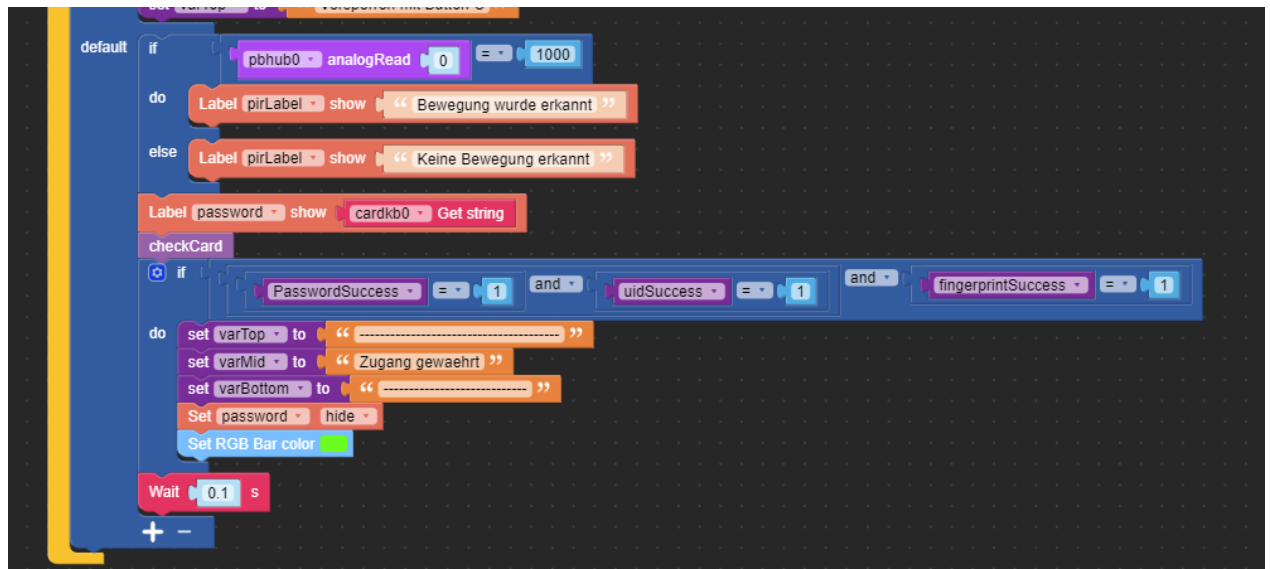
Button B wasPressed
  if configMode = 3
  do
    Wait 1 s
    set configMode to 4
    finger1 add user id 1 access 1

Button A wasPressed
  if configMode = 1
  do
    set Password to cardkb0 Get string
    set configMode to 7
    set varTop to "Passwort erfolgreich gespeichert"
    set varBottom to "Weiter mit escape"
  else if configMode = 5 and cardkb0 Get string = Password
  do
    pbhub0 Set pos num 1 RGB begin 0 count 1 color
    Set RGB Bar color
    play tone High A for 1/8 beat
    Set circle0 color
    Set circle0 borderColor
    set PasswordSuccess to 1
    set varTop to "Passwort erfolgreich"
    Wait 1 s
    Set RGB Bar color
  else if configMode = 5 and cardkb0 Get string != Password
  do
    Set RGB Bar color
    Wait 1 s
    Set RGB Bar color
  
```

112 RGB Komponente in Button



113 Beginn des Hauptteils

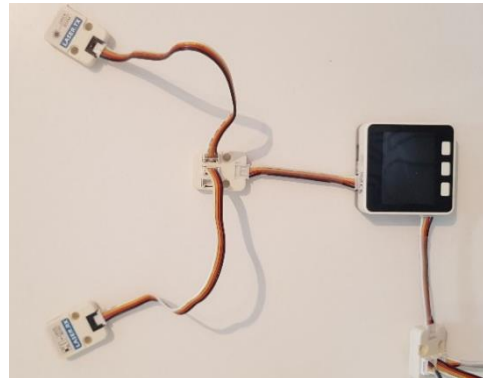


114 Ende des Hauptteils

5.5. Lichtschranke

5.5.1. Hardware

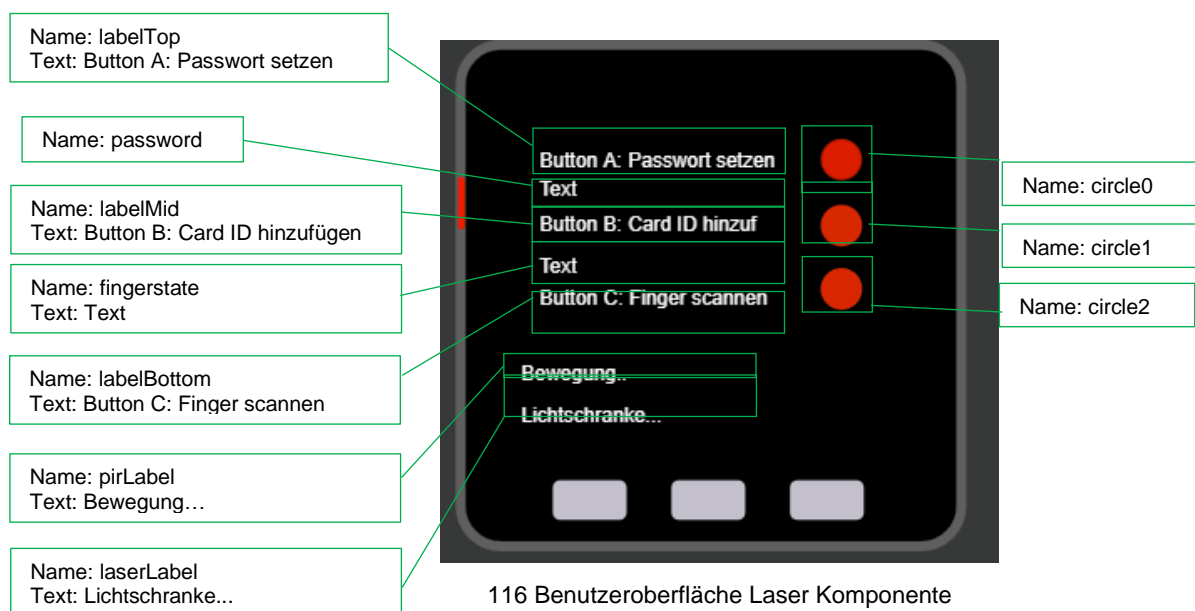
Im letzten Schritt möchten wir noch eine Lichtschranke hinzufügen. Dafür benötigen wir die Komponenten Laser Tx und Laser Rx sowie einen weiteren Grove Hub und drei Grove Kabel. Diese werden miteinander verbunden und dann Port B des M5Stacks angeschlossen.



115 M5Stack und LaserTX und LaserRX Komponenten

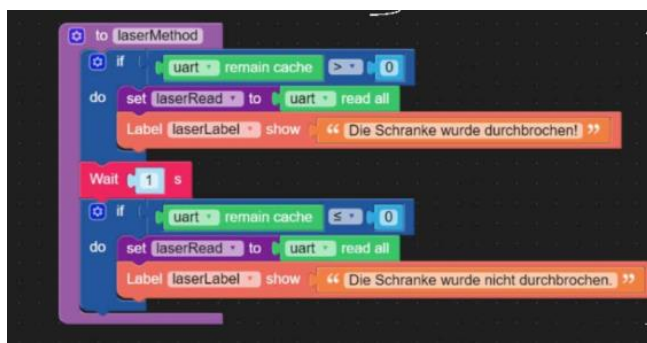
5.5.2. Benutzeroberfläche

Am Bildschirm gehört nun das entsprechende Label hinzugefügt. Dafür ziehen wir per Drag and Drop ein Label unter das pirLabel. Dieses benennen wir als laserLabel.



116 Benutzeroberfläche Laser Komponente

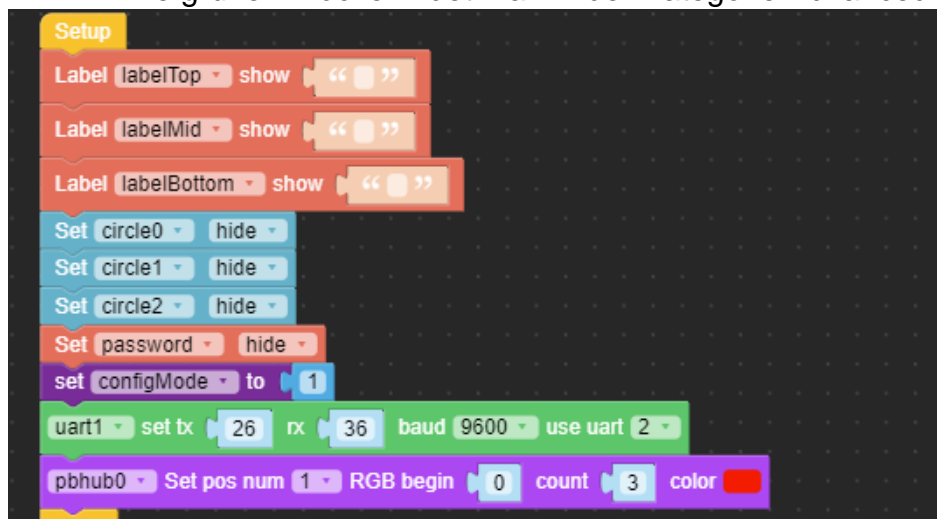
5.5.3. Programmcode



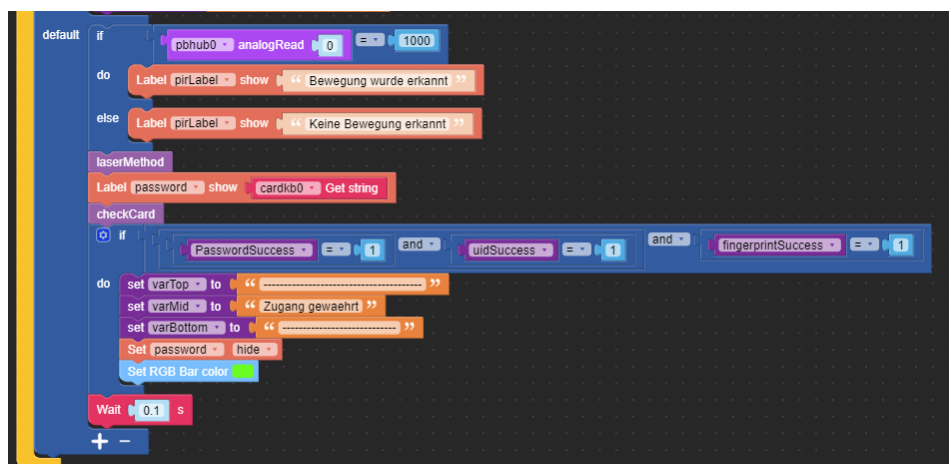
117 Funktion "laserMethod"

Bei der Implementierung beginnen wir mit der Erstellung einer eigenen Funktion. Anschließend wird die laserMethod-Funktion im Defaultblock des Switch-Case-Statements aufgerufen.

Die grünen Blöcke findet man in der Kategorie Advanced unter UART.



119 UART Block



120 Aufruf laserMethod

6. Abbildungsverzeichnis

Abbildung 1: M5Go IoT Starter Kit ESP32	6
Abbildung 2: Grove Hub Unit	6
Abbildung 3: M5Go IoT Starter Kit ESP32	6
4 Grove Hub	6
5 Pb.Hub.....	7
6 RGB LED Komponente	7
7 Motion Sensor.....	7
8 CardKB	8
9 RFID	8
10 Laser RX.....	9
11 Laser TX	9
12 Finger Unit	9
Abbildung 13: M5Stack und die dazugehörigen Units für das Sicherheit mit dem Internet-of-Things System	10
Abbildung 14: Beschriftung vom M5Stack.....	12
Abbildung 15: Benutzeroberfläche vom M5BurnerON/OFF Schalter.....	12
Abbildung 16: Benutzeroberfläche vom M5Burner	13
Abbildung 17: Ordnerstruktur vom M5Burner.....	13
Abbildung 18: Benutzeroberfläche vom M5Burner.....	13
Abbildung 19: Ordnerstruktur vom M5Burner	13
Abbildung 20: Setting Auswahl vom UIFlow.....	13
Abbildung 21: Ordnerstruktur vom M5Burner	13
Abbildung 22: Setting Auswahl vom UIFlow	15
Abbildung 23: Die Ansicht, nachdem man auf das Windows Symbol einen Rechtsklick ausführte.....	15
Abbildung 24: Setting Auswahl vom UIFlow	15
Abbildung 25: Die Ansicht, nachdem man auf das Windows Symbol einen Rechtsklick ausführte	15
Abbildung 26: Anzeige der verschiedenen Anschlüsse am Computer.....	16
Abbildung 27: M5Stack-Verbindung wird erfolgreich angezeigt.....	16
Abbildung 28: Erklärung zur UIFlow Oberfläche (anhand eines PIR Beispiel).	17
Abbildung 29: Event-Block in UIFlow	18
Abbildung 30: Beispiel für Variablen – Berechnung von 2 + 3 inklusive Ausgabe	19
Abbildung 31: Drag und Drop von einem Label auf den M5Core	19
Abbildung 32: Block Ansicht für "Label"	20
Abbildung 33: Hinzufügen einer Unit in UIFlow	20
34 Ablauf Anwendungsfall	21
35 Benutzeroberfläche	22
36 M5Stack mit Grove Hub und CardKB Komponente.....	23
37 Grove Hub	23
38 CardKB Komponente	23
39 Hinzufügen der Komponente in UIFlow.....	23
Abbildung 40: Die verschiedenen Möglichkeiten für die Benutzeroberfläche.....	24
Abbildung 41: Label-Einstellungen.....	24
Abbildung 42: Circle-Einstellung	25
Abbildung 43: M5Stack Benutzeroberfläche	25
44 Kategorie UI.....	26
45 Setup des Programms	26

46 Erstellen neuer Variablen.....	26
47 Wertzuweisung	26
48 Loop.....	26
49 Verwendung des Loop-Blocks	27
50 Anzeigen der Label	27
51 Switch-Case-Verzweigung	27
52 If-Bedingung in Kategorie Logic	27
53 If Bedingung.....	28
54 Erstellen einer neuen Funktion.....	28
55 Funktion "savePassword" Teil 1	28
56 Funktion "savePassword" Teil2.....	28
57 If-Else-Verzweigung.....	28
58 Funktion "save Password" Teil 3.....	28
59 Funktionsaufruf	29
60 Funktionsaufruf Teil 2	29
61 Button Event	29
62 If-Else If-Else Verzweigung	29
63 Button A Teil 1	29
64 Button A Teil 2	30
65 geschachtelte Bedingung.....	30
66 Komponenten für Bedingung	30
67 RGB Light und Speaker	30
68 Kategorie Hardwares	30
69 Kreis anzeigen	30
70 Timer Kategorie	30
71 Implementierung Button A.....	31
72 Implementierung der Escape - Taste	31
73 CardKB Blocks.....	31
74 Button C.....	31
75 Default Block.....	32
76 Hauptteil	33
77 Button A und C	33
78 RFID Komponente	34
79 RFID Komponente und Grove Hub	34
80 Units in UIFlow.....	34
81 Benutzeroberfläche.....	35
82 Funktion "saveCard"	35
83 RFID Blocks.....	35
84 Funktion "checkCard"	36
85 If-Else If-Verzweigung.....	36
86 Button B.....	36
87 Erweiterung Switch-Case-Verzweigung	36
88 neue Funktion erstellen.....	37
89 Erweiterung If Bedingung.....	37
90 Erweiterung Button C.....	37
91 benötigte Funktionen	38
92 Button A, B und C	39
93 Hauptteil des Programms.....	40

94 Units in UIFlow.....	41
95 Finger Komponente	41
96 Benutzeroberfläche.....	41
97 Funktion "saveFinger"	42
98 Fingerblöcke	42
99 erweiterte Bedingung	42
100 Erweiterung Button B	42
101 Funktion "getUnknown"	43
102 Funktion "read User with id access"	43
103 Erweiterung Button C.....	44
104 Hauptteil des Programms.....	45
105 verwendete Funktionen.....	46
106 Implementierung der Tasten	47
107 Implementierung der Finger-Unit.....	48
108 Motion Sensor Komponente.....	49
109 Erweiterung um Bewegungserkennung.....	50
110 Pb.hub Blöcke.....	50
111 RGB Komponente in Funktionen.....	51
112 RGB Komponente in Button	51
113 Beginn des Hauptteils	52
114 Ende des Hauptteils.....	53
115 M5Stack und LaserTX und LaserRX Komponenten	54
116 Benutzeroberfläche Laser Komponente	54
117 Funktion "laserMethod"	55
118 Aufruf laserMethod.....	55
119 UART Block	55
120 Aufruf laserMethod.....	55