

# Der Supersafe - elektrisch doppelt gesicherter Tresor mit Arduino

Von Moritz Lerch und Konrad Götz

## **KURZFASSUNG**

In unserem Projekt haben wir einen Sicherheitstresor entwickelt, der durch elektronische Hilfsmittel gesichert wird. Dies geschieht zuerst mittels eines RFID-Tags, das man an einen Sensor halten muss. Anschließend muss ein passender PIN eingegeben werden. Bei korrekter Eingabe wird ein elektromagnetisch verriegeltes Schloss geöffnet. Die Ansteuerung der Elektronik erfolgt über einen Mikrokontroller (Arduino), der mit von uns selbstentwickelter Software programmiert wurde.

## **Gliederung**

### 1. EINLEITUNG

### 2. BAU

#### 2.1. ERSTBESCHÄFTIGUNG MIT ARDUINO

#### 2.2. BAU DES PROTOTYPEN / MATERIALBESCHAFFUNG (ELEKTRONIK)

#### 2.3. PROGRAMMIERUNG

#### 2.4. PLANUNG DES FINALEN TRESORS

#### 2.5. BAU DES EIGENTLICHEN TRESORS

### 3. ERGEBNISSE

### 4. ZUSAMMENFASSUNG / AUSBLICK

### 5. LITERATURVERZEICHNIS / QUELLEN

### 6. UNTERSTÜTZUNGSLEISTUNGEN

## 1. EINLEITUNG

Ziemlich früh war uns klar, dass unser Projekt etwas mit Technik und Informatik zu tun haben sollte. Moritz Lerch war bereits mit Arduino und dessen Programmiersprache „C++“ vertraut, weshalb es nahe lag, den soeben genannten Microcontroller in unserem Projekt zu nutzen. Anfangs planten wir eventuell Bluetooth- Kopfhörer zu bauen, aber da es diese im großen Umfang zu kaufen gibt, wollten wir uns etwas Neues ausdenken. So kamen wir auf die Idee, einen Tresor mit mehreren Verifizierungsstufen zu bauen. Als erstes sollte man einen RFID-Tag (in Form einer Kreditkarte oder eines kleinen Tags, den man am Schlüsselbund trägt) an einen Sensor halten und anschließend auch noch einen vierstelligen Code eingeben. Nur so sollte man den Tresor öffnen können.

Der Schwerpunkt unserer Arbeit liegt hierbei auf der Programmierung.

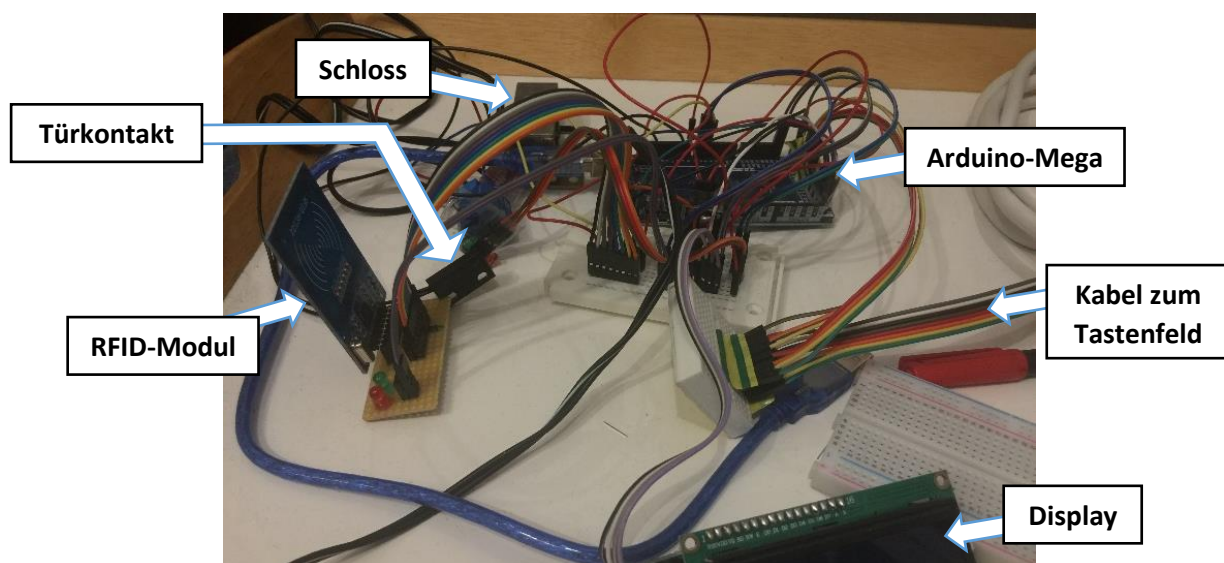
## 2. BAU

### 2.1. ERSTBESCHÄFTIGUNG MIT ARDUINO

Als nächstes mussten wir uns erst mit Arduino und C++ vertraut machen, dazu verwendeten wir Material der Schule. Programmiert haben wir auf unseren Tablets, welche wir wegen der Tablet-Klasse haben. Anfangs bauten wir zum Test einen Schaltkreis, der durch Knopfdruck eine kleine Lampe zum Aufleuchten brachte und später steigerten wir uns in immer schwerere Bereiche hinein. Wir kauften uns einen eigenen Arduino Baukasten und begannen auch die weiteren Bauteile für den Tresor zu kaufen. Darunter ein elektronisches Schloss (Solenooid), ein Display (I<sup>2</sup>C – LC Display; kurz LCD), ein Tastenfeld (4x4) und ein Relais.

### 2.2. BAU DES PROTOTYPEN / MATERIALBESCHAFFUNG (ELEKTRONIK)

Nachdem alle benötigten Materialien beschafft waren, die Gesamt-Kosten für die Elektronikbauteile betrugen hierbei circa 50 Euro, begannen wir mit dem Bau des ersten Prototypen. (Siehe Abb. 1) Der Bau war nicht besonders problematisch. Es mussten nur die Bauteile einzeln auf einem Breadboard platziert und mit anwendungsspezifischen Kabeln verkabelt werden. Die Programmierung hingegen stellte uns vor eine große Herausforderung, die wir erst nach intensiver Arbeit lösen konnten.



**Abb. 1, Prototyp Supersafe**

## 2.3. PROGRAMMIERUNG

Durch die Programmierung wird die Kommunikation zwischen den Sensoren und Aktoren, in unserm Fall angesteuert durch den Arduino-Mega, realisiert. Programmiert wurde das Ganze in der Arduino typischen Programmiersprache „C++“.

Grundlegend arbeitet das Programm mit zwei Modi. Je nachdem ob eine sogenannte Boolean-Variable, die Wahrheitswerte („true“; „false“) speichern kann, den Wert „WAHR“ oder „FALSCH“ besitzt, wird entschieden, ob nun der RFID-Sensor auf einen Chip „wartet“ oder das Tastenfeld auf eine Codeeingabe. So können einige Probleme, die beim Entwickeln des Programmcodes aufgetreten sind, vorgebeugt werden. Parallel werden alle wesentlichen Schritte mit einer zweifarbigen LED begleitet und mit einem kleinen Piezo-Lautsprecher positiv bzw. negativ quittiert. Das LCD zeigt außerdem Anweisungen oder Hinweise an.

Hier haben wir uns dafür entschieden, dass der Mikrocontroller mit dem Auslesen des RFID-Modules startet und anschließend der Code auf eine Tasteneingabe wartet.

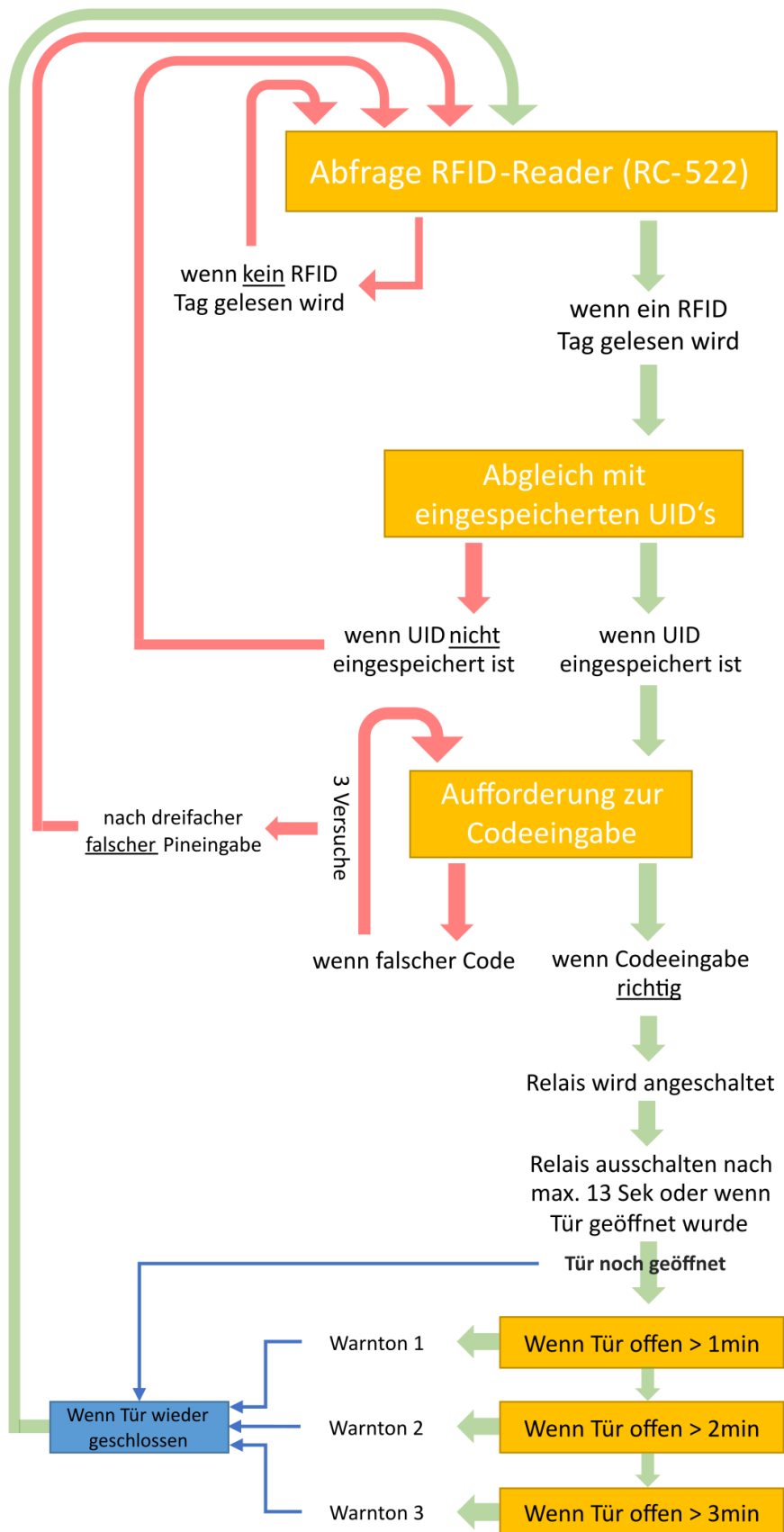
Wenn im ersten Modus (Auslesen des RFID-Sensors) der Sensor einen RFID- oder NFC-Tag erkennt, wird die einzigartige Kartennummer (UID) des Chips zwischengespeichert und anschließend mit einer zuvor eingelesenen und eingespeicherten UID abgeglichen. Falls nun die aktuelle UID mit der zuvor Eingespeicherten übereinstimmt wird sie als richtig „bewertet“ und durch das Ändern der Boolean-Variable auf „false“ zum zweiten Modus gewechselt, andernfalls wird die ausgelesene Chipnummer als falsch bewertet und erneut die UID abgerufen und diese wieder als wahr bzw. falsch eingestuft. Wenn die UID also nicht übereinstimmt, wertet das Programm die Karte immer wieder als falsch.

Sofern im ersten Modus die Karte „richtig“ war, wird der anschließend beschriebene, zweite Modus ausgeführt. Durch das Erstellen eines vierstelligen Platzhalters wird nun der nächste Schritt möglich. Hier wird jetzt auf einen Tastendruck am Keypad gewartet und der Platzhalter ausgefüllt. Nach vier eingegebenen Zeichen wird geprüft, ob der eingegebene Code dem richtigen Code entspricht, wenn ja dann wird der Tresor entsperrt, wenn nein wird in einer Variable plus eins gerechnet, dann wird wieder nach einem Code gefragt. Wenn die „Codeversuche“-Variable größer drei ist und es somit drei fehlgeschlagene Versuche gab, muss man, da der Modus durch die Boolean-Variable geändert wurde, wieder den Chip scannen. So kann „Bruteforcing“ („Erraten“ des Passwortes durch schnelles, direkt aufeinanderfolgendes Ausprobieren verschiedener Kombinationen) vermieden werden.

Durch Schalten eines Relais, das beim „Versorgen“ mit einem bestimmten Signal Stromfluss zwischen zwei Pins erlaubt, werden 12V auf ein sogenanntes „Solenoid“-Schloss geschaltet, das zum Öffnen der Tür mithilfe eines Magnetfeldes bei Stromfluss einen Bolzen zurückzieht. Durch einen Reed-Kontakt, der bei Kontakt mit einem Magneten, den Stromfluss unterbricht, kann durch eine Kombination mit einem Magneten zusätzlich noch überprüft werden, ob die Tür geöffnet ist. Wenn die Zeit, in der die Türe geöffnet ist, 1, 2 oder 3 Minuten überschreitet, wird eine akustische Warnung ausgegeben, die sich bei zunehmender Dauer ändert. So kann vermieden werden, dass die Tür zu lange offen steht.

Wenn der Tresor wieder geschlossen wurde, wird die Boolean-Variable wieder auf „true“ gesetzt, wodurch der erste Modus wieder von neuem beginnt.

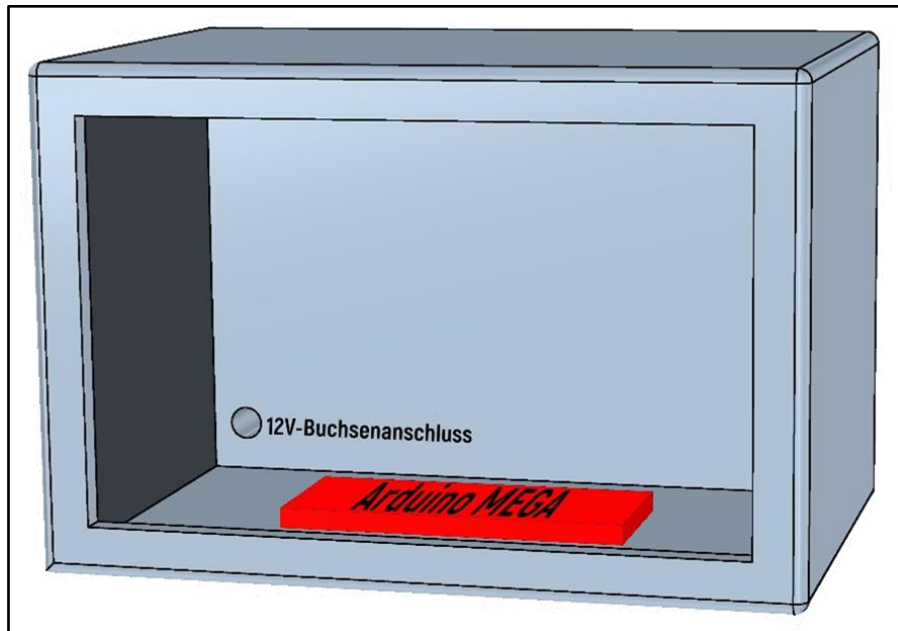
Auf dem nachfolgenden Bild sieht man ein vereinfachtes Verlaufsdiagramm, dass den Ablauf des Programmcodes noch einmal bildlich darstellen soll:



**Abb. 2, Verlaufsdiagramm Programmcodes**

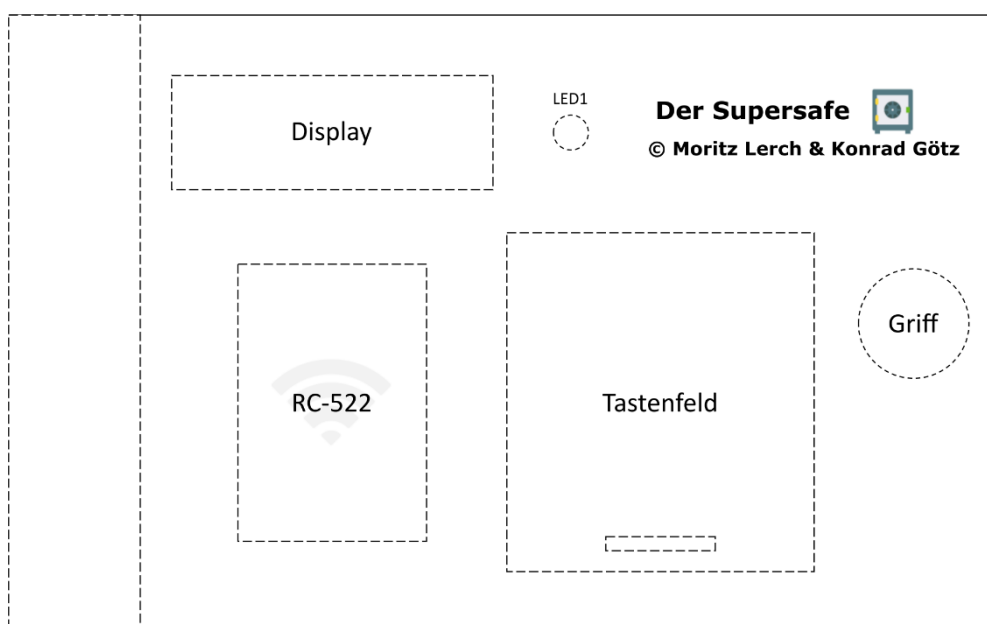
## 2.4. PLANUNG DES FINALEN TRESORS

Nach Fertigstellung der Programmierung wurde das Erstdesign des Safes auf Papier gezeichnet und dieses anschließend in ein 3D-CAD-Programm eingefügt. Dort wurden dann noch weitere Änderungen vorgenommen und so die finalen Maße bestimmt. Die Maße betragen 30cm · 20cm · 20cm (B · H · T). Diese wurden noch in das 3D-Programm eingepflegt. Schlussendlich ergab sich folgendes: (Wesentliche Bauteile beschriftet)



**Abb. 3, Design Tresorgehäuse**

Nun wurde noch ein Design für die Tür, an der später die Sensoren usw. sitzen sollten, angefertigt. Im nachfolgenden Bild sieht man die Türfront schematisch mit allen anzubringenden Bauteilen. Später wurden allerdings noch Kleinteile sowie das Solenoid-Schloss ergänzt.



**Abb. 4, Tresortür**

## 2.5. BAU DES EIGENTLICHEN TRESORS

Das Gehäuse des Tresors wurde selbstgebaut, da ein vorgefertigter (Massiv-)Tresor zum einen zu teuer gewesen wäre (wir wollten darauf achten das „Gesamtpaket“ in einem recht preisgünstigen Rahmen zu halten) und zum anderen nicht modular genug wäre, um eigene Bauteile hineinzubauen.

Hierzu wurde eine stabile, 12mm dicke Holzplatte aus dem Baumarkt verwendet, die maßgerecht zugesägt und verschraubt wurde. Die Frontplatte bekam einen Ausschnitt für die Tür, die somit den Rahmen von der Tür zum Rest bildet. Anschließend wurde die Tür noch mit den nötigen Ausschnitten für die Elektronikbauteile versehen. (analog zu Abb. 4)

Mit Holz und Eisenwaren erhöht sich der Preis von 50€ auf 60€.

Nun wurde der Prototyp aufgelöst und die elektronischen Bauteile in das fertige Gehäuse und an die Tür gebaut. Hierfür mussten noch einige Lötarbeiten vonstattengehen, wofür wir viel Zeit und großes Fingerspitzengefühl benötigten. So wurden einige sog. Breakout-Boards für einzelne Komponenten hergestellt, damit diese bei Schadensfall leicht austauschbar sind. Zum Schutz der Elektronik wurde im Innenraum über jener noch eine Acrylglas-Scheibe eingefügt, sodass der Tresorraum zweigeteilt ist.

Außerdem wurde noch ein Schlüsselschalter für die einfache Wartung eingebaut, mit dem man bei Bedarf mit einer einfachen Parallelschaltung das Relais umgehen kann.

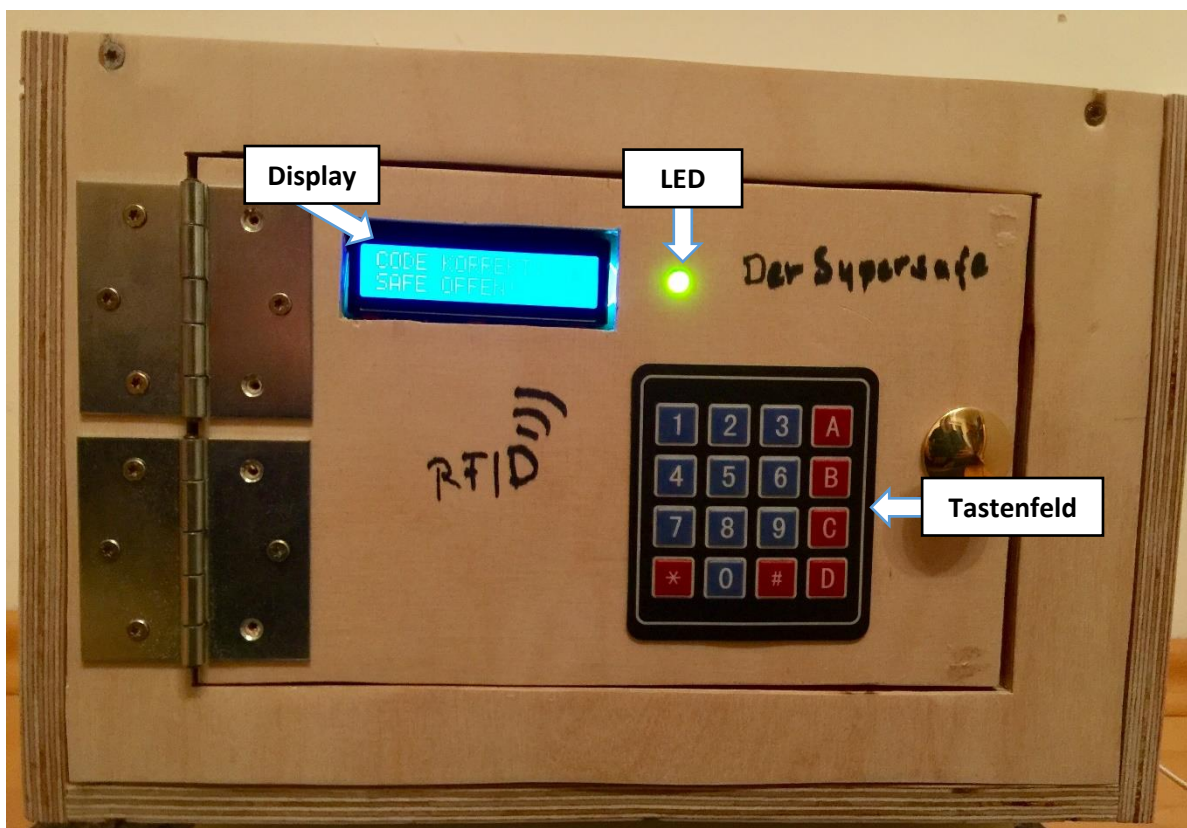
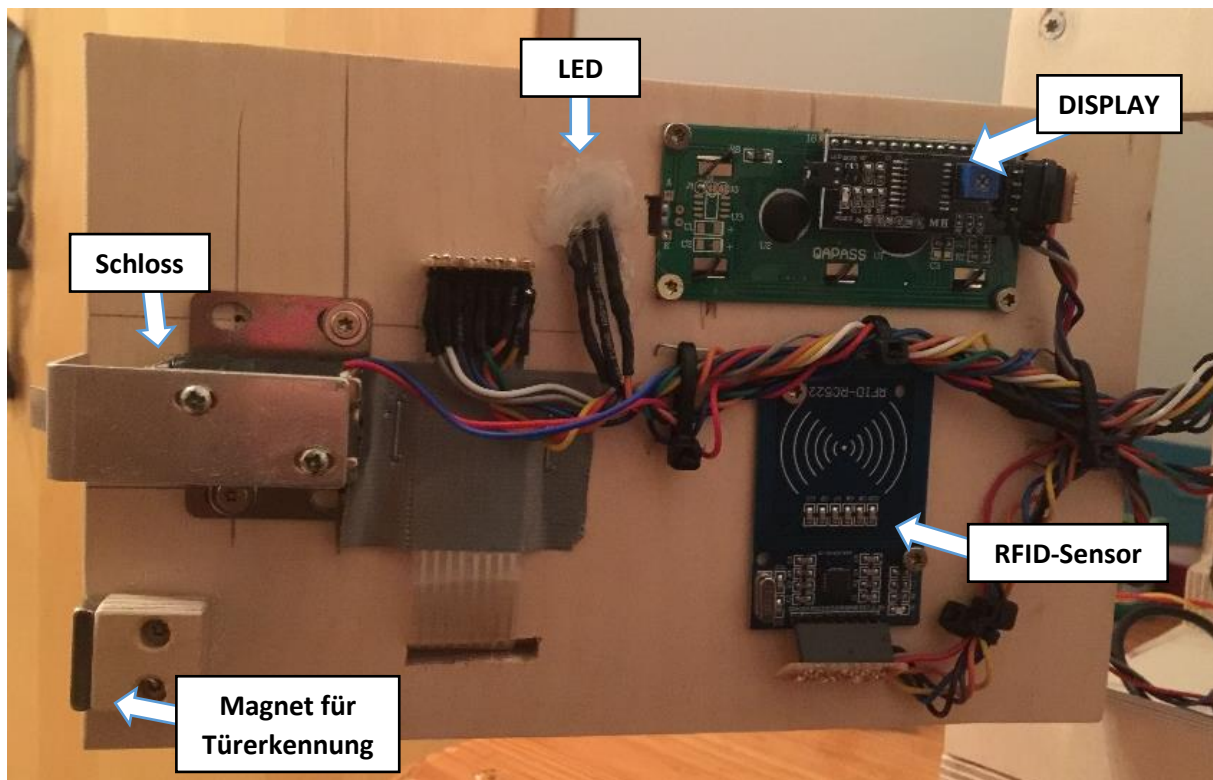


Abb. 5, Frontansicht Supersafe





**Abb. 6, Elektronik Innenansicht**



**Abb. 7, Gesamtansicht Safe**



### **3. ERGEBNISSE**

Durch einige Tests hat sich uns bestätigt, dass unsere Programmierarbeit in Kombination mit dem Arduino reibungslos funktioniert und erfolgreich war.

Es sind uns (auch in den verschiedenen Tests) keine Umgehungsmöglichkeiten zu irgendeinem Punkt in unserem Code aufgefallen, was bei diesem Einsatzgebiet wirklich relevant ist. Nach unseren Maßstäben ist der Tresor sicher!

### **4. ZUSAMMENFASSUNG / AUSBLICK**

Im Großen und Ganzen war unser Projekt erfolgreich und wir haben durch das Projekt viele weitere Arduino-Kenntnisse erlangt und Erfahrungen gesammelt. Insgesamt haben wir für unser Projekt ca. 1 Schuljahr gebraucht (bei 1h pro Woche), wovon die Programmierung die meiste Zeit einnahm. Auch die Lötarbeiten waren wie oben beschrieben sehr zeitintensiv und erforderten handwerkliches Geschick und große Sorgfalt.

Die Elektronik des Tresors, auf die es uns ankam, könnte für (semi-)professionelle Zwecke genutzt werden. Man könnte mit den Komponenten eine Tür zu einem Raum ausstatten, bei dem es zum einen wichtig ist, dass nur befugte Personen den Zutritt erhalten und dass sich die Elektronik zum anderen durch optische und akustische Signale „meldet“, wenn die Tür nicht wieder geschlossen wurde, z.B. beim Zugang zu einem Kühlraum / Medikamentenlager. Uns ist bewusst, dass es solche Bauelemente schon als ganzes Paket für die Industrie gibt, allerdings zu höheren Preisen. Wir wissen um die Schwierigkeiten bei der Programmierung und Produktion.

Durch die Beschäftigung mit der Thematik sind wir nun in der Lage solche Teile kostengünstig mithilfe einer Programmierung zu verwenden und für den Eigen-/Hobbybedarf zu nutzen.

## **5. LITERATURVERZEICHNIS / QUELLEN**

- 1) <https://funduino.de/anleitungen>
- 2) <https://starthardware.org/arduino-programmieren/>
- 3) <https://www.arduino.cc/reference/de/>

Alle Fotos/Grafiken/Abbildungen stammen aus eigener Anfertigung.

## **6. UNTERSTÜTZUNGSLEISTUNGEN**

- Martin Perleth, Physiklehrer, Betreuer Jugend Forscht: Organisatorisches zum Wettbewerb und Hilfe bei der Langfassung