

Projekthandbuch:  
**„X-Safe – Safe to Go“**

Modul: Hands-on Innovation  
SoSe 2019

Vorgelegt bei: Prof. Oliver Mauroner  
David Zakoth

# Inhaltsverzeichnis

I. Abbildungsverzeichnis.....	2
1. Geschäftsidee.....	3
1.1 Mission Statement.....	3
1.2 Vision Statement.....	3
2. Business Model.....	3
2.1 Value Proposition.....	4
2.2 Customer Segments.....	5
2.3 Revenue Streams.....	8
3. Bauanleitung X-Safe.....	10
3.1 Materialliste:.....	10
3.2 Vorbereitung für den Bau des Prototyps.....	11
3.3 Bauanleitung X-Safe.....	11
3.3.1 Zusammensetzen der einzelnen Holzplatten des Tresorgehäuses.....	11
3.3.2 Einsetzen der Tür mit Scharnieren.....	11
3.3.3 Bearbeitung Türinnenseite und Anbringung des Arduino.....	12
3.3.4 Verkabelung der Sensorik und Elektronik.....	17
3.3.5 Einbau Tragefunktion.....	18
3.3.6 Anbringen des Teamnamens auf der Seite des X-Safe.....	19
4. Programmierung.....	21
4.1 Zusatzprogrammierung Fingerprint.....	24
II. Anhang.....	29
III. Literaturverzeichnis.....	32

## I. Abbildungsverzeichnis

Abbildung 1: Anteil der Kriminalitätsbereiche an Straftaten gesamt.....	4
Abbildung 2: Wo schlafen Sie meistens auf Urlaubsreisen? .....	6
Abbildung 3: Geschäftsreisen nehmen zu .....	7
Abbildung 4: Kostenrechnung X-Safe.....	8
Abbildung 5: Boden des Tresors .....	11
Abbildung 6: Tür mit Schanieren .....	12
Abbildung 7: Anbringung Solenoid Door Lock .....	12
Abbildung 8: Kegelbohrer für Loch des Touchbuttons .....	13
Abbildung 9: Anbringung Touchsensor und LED .....	14
Abbildung 10: Fertige Vorderseite des Tresors.....	14
Abbildung 11: Befestigung Arduino.....	15
Abbildung 12: Anbringung der Batterien mit Lochbändern.....	16
Abbildung 13: Presszange und Verkleidung des Kabels mit Metallschuh .....	16
Abbildung 14: Innenseite der Tür.....	17
Abbildung 15: Anbringung Tragegurte .....	18
Abbildung 16: Anbringung Teamname .....	19
Abbildung 17: Fertiger Safe .....	20
Abbildung 18: Schematischer Aufbau Arduino.....	21
Abbildung 19: Gesamter Programmcode.....	22
Abbildung 20: Programmzeilen 1 bis 4 des Programmcodes.....	23
Abbildung 21: Programmzeilen 5 bis 10 des Programmcodes .....	23
Abbildung 22: Programmzeile 12 bis 15 des Programmcodes .....	24
Abbildung 23: Programmzeile 16 bis 20 des Programmcodes .....	24
Abbildung 24: Herunterladen der Bibliothek für den Fingerprintsensor .....	25
Abbildung 25: Bibliotheksverwalter .....	25
Abbildung 26: Enroll-Funktion.....	26
Abbildung 27: Einspeichern Finger .....	27
Abbildung 28: Übertragung an Arduino.....	28

# 1. Geschäftsidee

X-Safe ist ein Kleinunternehmen der drei Gründer Airidas Bertulis, Dario Dorsano und Nikolas Kunkel. Das Unternehmen stellt Safes in verschiedenen Größen her. X-Safe bedient sich an den typischen Elementen eines Safes, eines Rucksacks und dem Arduino. Wir kombinieren die Vorteile und Vorzüge der drei genannten Komponenten und verknüpfen diese. Das bedeutet, dass die Nachteile eines herkömmlichen Safes, nämlich die Immobilität und das Gewicht, durch den Einsatz von Leichtmetall und Tragegurten kompensiert werden. Dadurch wird der X-Safe handlich, mobil und flexibel einsetzbar. Der X-Safe wird aus dem Leichtmetall Titan gefertigt und ist durch einen Fingerscan individuell geschützt. Es bedarf somit weder einer Geheimzahl noch einem Schlüssel zum Öffnen des Safes. Zur Flexibilität und weiteren Nutzensteigerung des Tresors kann dieser durch eine einfache Verschraubung an einer Wand, Schrank oder Ähnlichem befestigt werden. Außerdem können Tragegurte angebracht werden, die den Safe transportabel machen. Diese Tragegurte kann man über die Befestigung mit Karabinerhaken bequem ein- und aushängen. Ziel ist es die Produktion outzusourcen und über eine Massenmarktstrategie das Produkt über mehrere Distributionskanäle an die Kunden zu bringen.

## 1.1 Mission Statement

Wir revolutionieren Safes, um die Sicherheitsbedürfnisse unserer Kunden flexibler und alltags-tauglicher zu gestalten. Wir erschaffen „Sicherheit zum Mitnehmen“.

## 1.2 Vision Statement

Wir streben danach ein führender Anbieter für Safes zu werden. Durch Innovation und eine in dieser Form nie dagewesene Flexibilität in diesem Markt soll ein positives und nachhaltiges Image aufgebaut werden.

# 2. Business Model

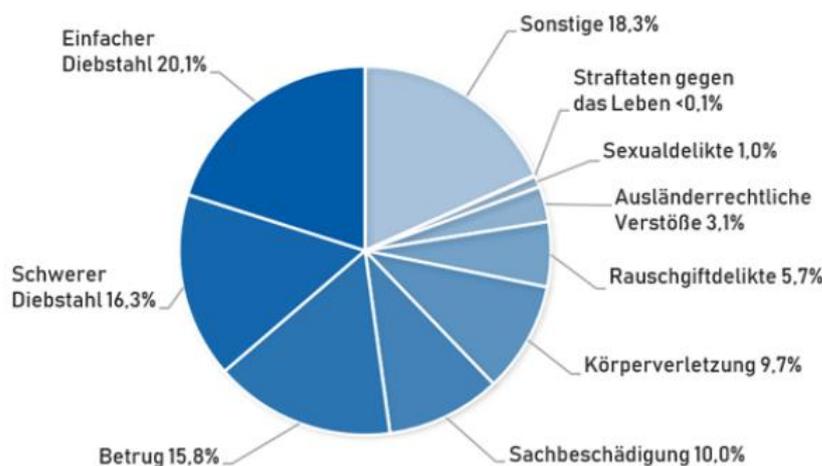
Um eine Geschäftsidee zu realisieren ist es wichtig das gesamte Umfeld eines Unternehmens zu betrachten. Um eine gute Übersicht über Unternehmensfelder zu bekommen, wurde das Business Model Canvas als Hilfe herangezogen. Dieses Modell dient als gute Hilfestellung, um den eigenen Wert des Unternehmens einschätzen zu können und daraus Handlungsstrategien zu entwickeln. Das Business Modell unterscheidet zwischen den Bereichen Key Partners, Key Activities, Key Resources, Value Propositions, Customer Relationships, Channels,

Customer Segments, Cost Structure und Revenue Streams. Jedes einzelne Feld hat eine für die Unternehmung wichtige Rolle und sollte beachtet werden. In unserer Projektarbeit im Rahmen des Moduls „Hands on Innovation“ haben wir uns auf folgende drei Aspekte konzentriert: Value Proposition, Customer Segments und Revenue Streams. In den einzelnen Feldern wird zur Erklärung auch eine kurze Übersicht über die Vertriebskanäle sowie die Kostenstruktur gegeben. Durch Analyse der Felder ist ein Use Case entstanden, welcher für das Unternehmen sehr wichtig ist, um den Vertrieb Kundengerecht zu gestalten.

## 2.1 Value Proposition

Im Bereich Value Proposition oder zu Deutsch „Wertangebot“ soll der Nutzen für den Kunden verdeutlicht werden. Hierbei wird nicht nur das Produkt an sich betrachtet, sondern auch die Dienstleistung, die das Produkt umhüllt.

**Anteil der Kriminalitätsbereiche an Straftaten gesamt**



© Quelle: Bericht zur Polizeilichen Kriminalstatistik 2017

Abbildung 1: Anteil der Kriminalitätsbereiche an Straftaten gesamt

Die Abbildung 1: „Anteil der Kriminalitätsbereiche an Straftaten gesamt“ zeigt auf, dass der Bereich Diebstahl 36,4% aller Kriminalitätsdelikte in Deutschland ausmacht. Gerade bei Wohnungseinbrüchen werden häufig nicht geschützte Wertsachen entwendet. Aber auch in Hotels und vor allem in Hostels besteht immer die Gefahr von Diebstählen. In diesem Bereich stellt X-Safe einen hervorragenden Kundennutzen dar. X-Safe bietet einen Tresor zum Schutz der Wertsachen an, der zum einen über eine einfache Verschraubung einbaubar ist, zum anderen aber auch durch Tragegurte und das erheblich reduzierte Gewicht, im Vergleich zu Mitbewerbern, sehr gut zu transportieren ist. Durch das genutzte Leichtmetall Titan, ist der Safe nicht zu schwer und kann auch für Kurzurlaube, Geschäftsreisen oder ähnliches genutzt werden.

Er wird in verschiedenen Größen angeboten. Der Kunde hat die Möglichkeit zwischen 30x15x15 cm, 45x20x20 cm und 60x30x30 cm zu wählen. Das Produkt wird möglichst günstig gestaltet um es für Studenten, Berufseinsteiger und Reisende attraktiv zu machen. Um den Kunden vor einem lästigen Schlüssel zu schützen, ist ein Fingerscan, bei dem mehrere Finger eingescannt werden können, installiert. Dieser bietet einen zusätzlichen Sicherheitsvorteil und verhindert einen Verlust oder das Vergessen eines Schlüssels oder einer Geheimzahl.

Sicherheit stellt ein Grundbedürfnis eines jeden Menschen dar, welches wir, X-Safe, abdecken wollen. Dieses Grundbedürfnis bietet auch einen nachhaltigen Wert für unsere Unternehmung. Trotz veränderter Prozesse und Wirtschaftsbedingungen wird ein Bedürfnis nach Sicherheit und Schutz vor Diebstählen immer vorhanden sein.

X-Safe geht hierbei speziell auf den Bereich Diebstahl ein und bietet dem Kunden einen Schutz daheim und gleichzeitig auch eine Möglichkeit diesen Schutz an jeden Ort ihrer Wahl mitzunehmen.

## 2.2 Customer Segments

Der Bereich Customer Segments oder zu Deutsch „Kundensegmente“ soll Aufschluss über die genaue Zielgruppe geben. Hierbei soll auch dargestellt werden in welcher Häufigkeit das Produkt nachgefragt bzw. genutzt werden kann.

Die grundsätzliche Verkaufsstrategie der Unternehmung beruht auf einer Massenmarktstrategie. Dies liegt schlicht und ergreifend daran, dass das Produkt grundsätzlich für Jedermann hergestellt wird und auch von jedem genutzt werden kann. Außerdem soll durch einen günstigen Preis eine Vielzahl von Safes verkauft werden. Trotz alledem kristallisieren sich drei Kundengruppen heraus, für die das Produkt besonders geeignet ist. Auf diese Kundengruppen sollen die Marketingmaßnahmen besonders zugeschnitten werden.

Kundengruppe 1 stellen Studenten dar. Zu Beginn ihres Studiums stehen junge Leute häufig vor einem Wohnortswechsel. Gerade in Wohnheimen aber auch in WGs besteht ein freier Zugang zu Wertsachen der Mitbewohner. X-Safe bietet genau dieser Kundengruppe die Möglichkeit wichtiges Hab und Gut sicher zu verstauen. Durch eine einfache Verschraubung ist der Safe leicht an der Wand anzubringen, kann aber gleichzeitig für eine Fahrt nach Hause abmontiert werden und auch an einem anderen Ort genutzt werden.

Eine weitere Kundengruppe stellen Reisende dar. Grundsätzlich sind hier sämtliche Urlauber als Zielgruppe zu erfassen. Allerdings wollen wir mit unserem Geschäftsmodell besonders jüngere Reisende und sogenannte Backpacker ansprechen.

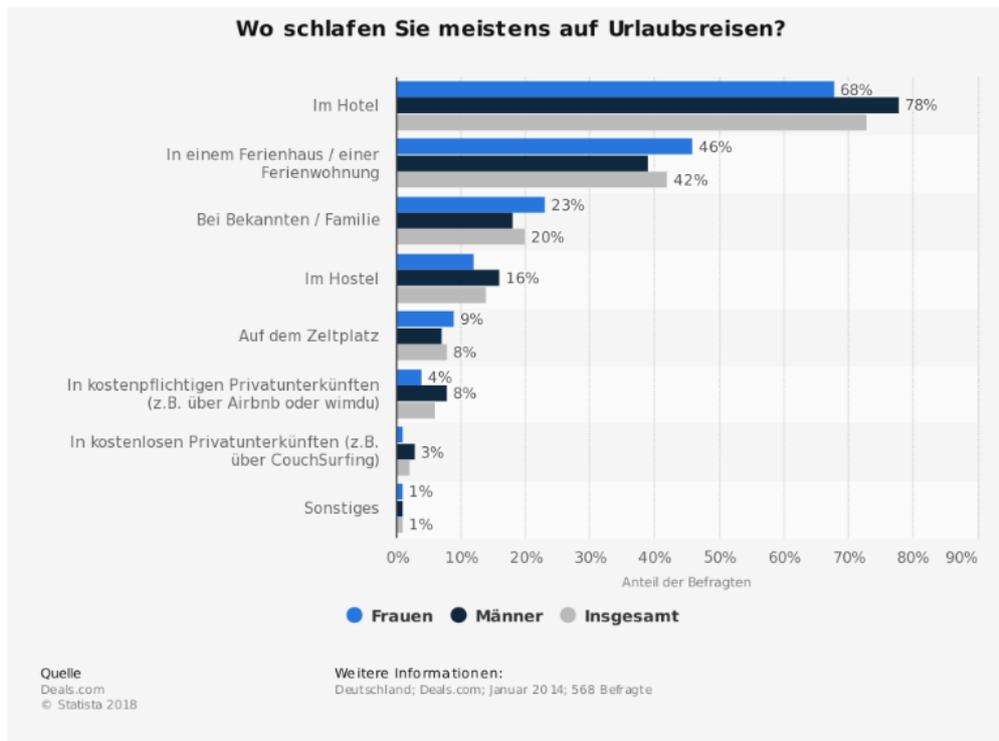


Abbildung 2: Wo schlafen Sie meistens auf Urlaubsreisen?

Betrachtet man die oben aufgeführte Abbildung 2: „Wo schlafen Sie meistens auf Urlaubsreisen?“ fällt auf, dass das Nächtigen in Hostels, Privatunterkünften sowohl kostenpflichtig als auch kostenlos und auf Zeltlagern einen Anteil von 34% bei Männern und 28% bei Frauen ausmacht. In all diesen Bereichen herrscht eine recht große Problematik bei der sicheren Verwahrung von Wertsachen. In Hostels teilen sich Reisende Zimmer mit Fremden. Häufig stehen hier keine Schließfächer zur Verfügung. Hier bietet X-Safe für genau diese Kundengruppe eine ideale Lösung. Wichtige Reisedokumente sowie weitere Wertgegenstände können einfach in einem Safe, vorzugsweise das kleinere Modell von 30x15x15 cm, verwahrt und auch einfach transportiert werden.

Dieselbe Problematik löst X-Safe natürlich auch für Reisende auf Zeltplätzen oder in kostenfreien Privatwohnungen. Auch hier sind wichtige Dokumente und teure Wertgegenstände nicht ohne weiteres zu verstauen. Auch hier bietet der flexible Safe eine Lösung.

Die dritte hervorgehobene Kundengruppe stellen Geschäftsreisende dar.

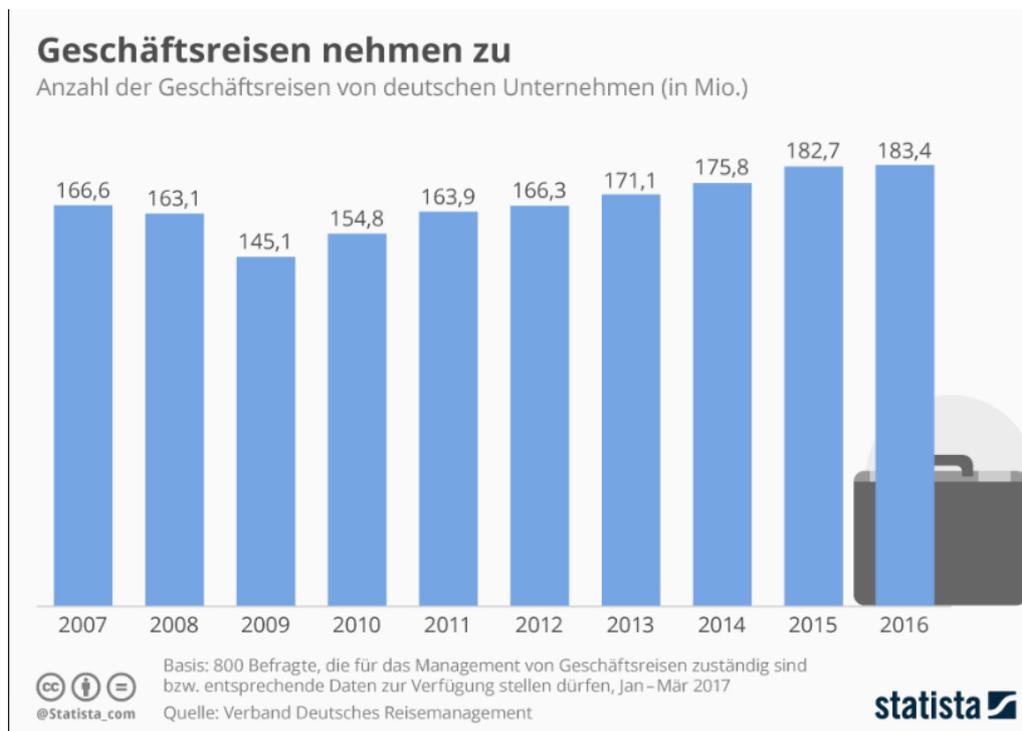


Abbildung 3: Geschäftsreisen nehmen zu

Wie die oben aufgeführte Statistik 3: „Geschäftsreisen nehmen zu“ belegt, nimmt die Anzahl der Geschäftsreisenden zu. Die globale Vernetzung der heutigen Welt wird immer stärker. Überseeaufenthalte sind zur Gewohnheit geworden.

Bei diesen Auslandsaufenthalten stehen die Arbeitstätigen häufig in Stresssituationen. Lange Flüge, Zeitverschiebungen und wenig Kenntnis über das bereiste Land, macht es Taschen-diebstahl einfach sich Wertsachen zu beschaffen. Auch hier bietet X-Safe eine tolle Lösung. Die Flexibilität und Sicherheit des Tresors schützen den Kunden genau in dieser Situation.

Natürlich gilt dies nicht nur für Berufstätige die Übersee reisen, sondern auch für diejenigen die lokal oder nur in Nachbarländer reisen. Für regelmäßige Zugfahrten beispielsweise bietet der X-Safe eine tolle Lösung.

Die Häufigkeit der Nachfrage wird sich pro Kunde auf eine einmalige Nachfrage beschränken. Als Kunde können hier gesamte Familienhaushalte betrachtet werden, da die Möglichkeit besteht mehrere Finger einzuscannen.

## 2.3 Revenue Streams

Der Bereich Revenue Streams soll Aufschluss über die geplanten Umsätze der Unternehmung liefern. In diesem Teil wird der Preis des Produktes aufgezeigt, wie hoch die Gewinnmarge sein soll und über welche Kanäle der X-Safe vertrieben wird. Zur genaueren Aufschlüsselung wird auch eine kurze Strukturierung der Kosten aufgezeigt, aus der sich die Gewinnmarge ableiten lässt.

Zunächst werden die Kosten der Herstellung eines X-Safe betrachtet. Der Plan ist die komplette Herstellung outzusourcen, wodurch den drei Gründern lediglich Aufgaben im Bereich Marketing, Verwaltung und Finance zufallen. Hierdurch können zusätzlich auch noch Personalkosten gespart werden. Bei der nachfolgenden Kalkulation sind Massenproduktions- und Synergieeffekte noch nicht berücksichtigt.

### Kostenrechnung X-Safe

<b>Herstellungsteile</b>	<b>Herstellungskosten pro Stück (Produktion 1000Stk/Mo.)</b>
Arduino Uno inkl. Sensorik/Elektronik	10€ / Stück
Gehäuse (Titan)	7€ / Stück
Schrauben u. ä.	0,1 € / Stück
Tragevorrichtung	0,5 € / Stück
<b>Gesamtkosten für Herstellung</b>	<b>17,6€ / Stück</b>
Gewinnmarge	+ 70,45%
<b>Verkaufspreis</b>	<b>30€ / Stück</b>

Abbildung 4: Kostenrechnung X-Safe

In der oben aufgeführten Abbildung 4: „Kostenrechnung X-Safe“ erfolgt eine grobe Aufschlüsselung der Kosten für einen X-Safe. Hier wurde zur Vereinfachung mit einem Durchschnittspreis sowohl für Kosten als auch für den Verkaufspreis gerechnet. Kosten und Preis sind je nach Modell des X-Safe zu differenzieren. Die größte Größe des Tresors kostet 35,-€, die mittlere Größe 30,-€ und die kleinste 25,-€. In der Rechnung wurde außerdem ein Absatz von 1000 Stück pro Monat angenommen. Dieser wurde gleichmäßig auf alle Modelle des X-Safe verteilt.

Allerdings muss erwähnt werden, dass diese Kostenrechnung einmalige Kosten, beispielsweise die Anschaffung von Laptops oder Software, nicht beinhaltet. Auch laufende Kosten wie die Miete für das Büro, Lagerkosten und der Unternehmerlohn sind nicht miteinberechnet. Allerdings soll mit dieser Kostenrechnung zunächst nur die Herstellungsseite betrachtet werden. Alle weiteren Kosten sollen mittels der Gewinnmarge gedeckt werden. Der Vertrieb soll über verschiedenste Absatzwege erfolgen. Zum einen soll Online über eine eigene Website und über Online-Plattformen wie Amazon vertrieben werden. Außerdem soll der X-Safe auch bei herkömmlichen Einzelhändlern, wie beispielsweise Hornbach oder Bauhaus, im Sortiment

auftauchen. Des Weiteren ist die Eröffnung eines Flagshipstores für innovative und mobile Sicherheitsbehältnisse in Frankfurt geplant. Hier können sich potenzielle Kunden informieren und auch vor Ort unsere Produkte erwerben

Bei einem Absatzvolumen von 1000 Stück pro Monat entsteht ein monatlicher Umsatz von 30.000€. Auf das gesamte Jahr hochgerechnet entsteht ein Umsatz von 360.000€. Diese Absatzzahlen sollen nach Möglichkeit über intensive Marketingmaßnahmen in den Folgejahren gesteigert werden. Die Gewinnmarge von 70,45% pro Stück ergibt einen monatlichen Gewinn von 12.400€. Auf das Jahr gesehen entsteht somit ein Gewinn von 148.800€. Ziel ist eine Gewinnmarge in den Folgejahren von mindestens 150% pro verkaufte Einheit.

Abschließend ist zu sagen, dass es sehr wichtig wird über intensive Marketingmaßnahmen den Safe an den Kunden zu bringen. Die Flexibilität ist einzigartig und bietet einen Mehrwert. Die Sicherheitsvorrichtung über einen Fingerscan vereinfacht die Bedienung.

## 3. Bauanleitung X-Safe

### 3.1 Materialliste:

- Arduino Uno
- Arduino Shield
- Relay
- Touch Button
- LED Socket
- Solenoid Door Lock
- 12 V DC Batterie
- Holzplatten 60x30cm (4x), 30x30cm (2x)
- Scharniere (2x)
- Plastikröhrchen (2x)
- Unterlegscheiben (18x)
- Schrauben/Nägel (60x)
- Winkel (8x)
- Farbspray
- Schablone
- Karabiner (4x)
- Schleifpapier
- Tragegurte (2x)
- Bohrmaschine
- Kegelbohrer
- Scharbeisen
- Akkuschauber
- Spanneisen
- Muttern (4x)
- Türknauf
- Füße (4x)
- Kabelbinder
- Kabelschuhe (2x)
- Presszange
- Skalpell
- Zeitung

## 3.2 Vorbereitung für den Bau des Prototyps

Vor dem Baubeginn des Prototyps galt es die Maße und Features des X-Safes festzulegen. Hierfür benutzten wir den Microsoft 3D Builder. Er half uns ein einfaches Modell zu kreieren und die Funktionen und Maße des Prototyps festzulegen. Außerdem bot es die Möglichkeit zu erörtern wie und wo welche Teile verbaut und in welcher Reihenfolge die Komponenten eingebaut werden sollen.

Anschließend gab es ein Brainstorming über die Materialien, Farbe etc. sodass man nun mit der Beschaffung der Materialien und dem anschließenden Bau des Prototyps beginnen konnte.

## 3.3 Bauanleitung X-Safe

### 3.3.1 Zusammensetzen der einzelnen Holzplatten des Tresorgehäuses.

Zu Beginn werden die vier 60x30cm Holzplatten (für die Seiten) mit den 30x30cm Holzplatten (für Dach und Boden) zusammengesetzt. Zur Montage werden Schrauben, Winkel und ein Akkuschauber genutzt. Des Weiteren wird der Boden einige Centimeter nach oben versetzt, um eine schönere Optik und eine verbesserte Standfestigkeit zu generieren.

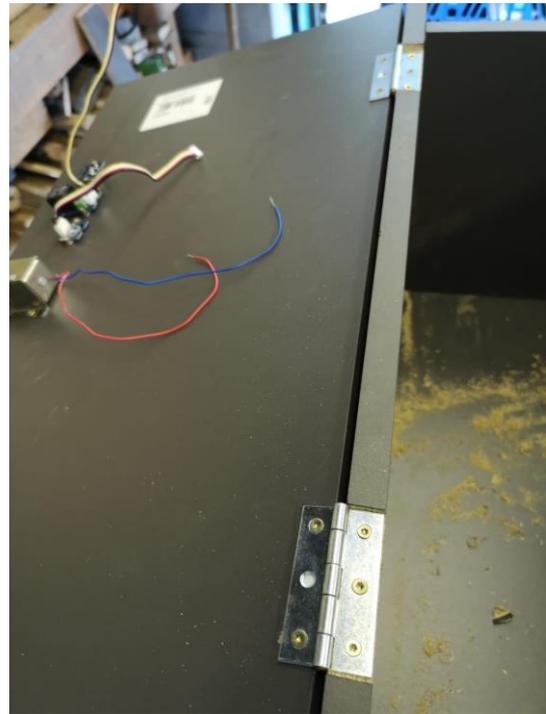


Abbildung 5: Boden des Tresors

### 3.3.2 Einsetzen der Tür mit Scharnieren.

Damit sich die Tür bestmöglich öffnen lässt wird hier mit Scharnieren gearbeitet. Um diese einzusetzen müssen die zwei Stellen, an denen die Scharniere eingesetzt werden sollen,

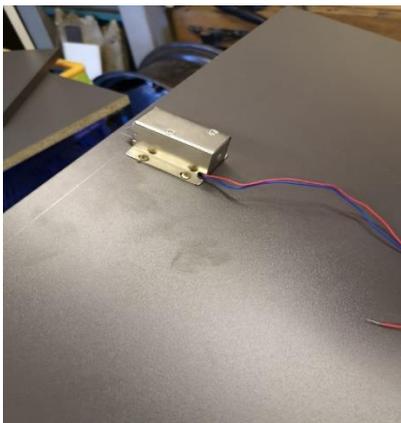
ausgefräst werden. Hierfür benötigt man Schrauben, Scharniere, Akkubohrer, Fräse und Schleifpapier.



*Abbildung 6: Tür mit Scharnieren*

### 3.3.3 Bearbeitung Türinnenseite und Anbringung des Arduino

Zuerst wird der Solenoid Door Lock mit kleinen Schrauben an der Türinnenseite befestigt. Um das Schloss einrasten zu lassen muss man auf der Gegenseite mit einem Scharbeisen ein ca. 3-4 Millimeter tiefes Rechteck in die Seitenplatte brechen.



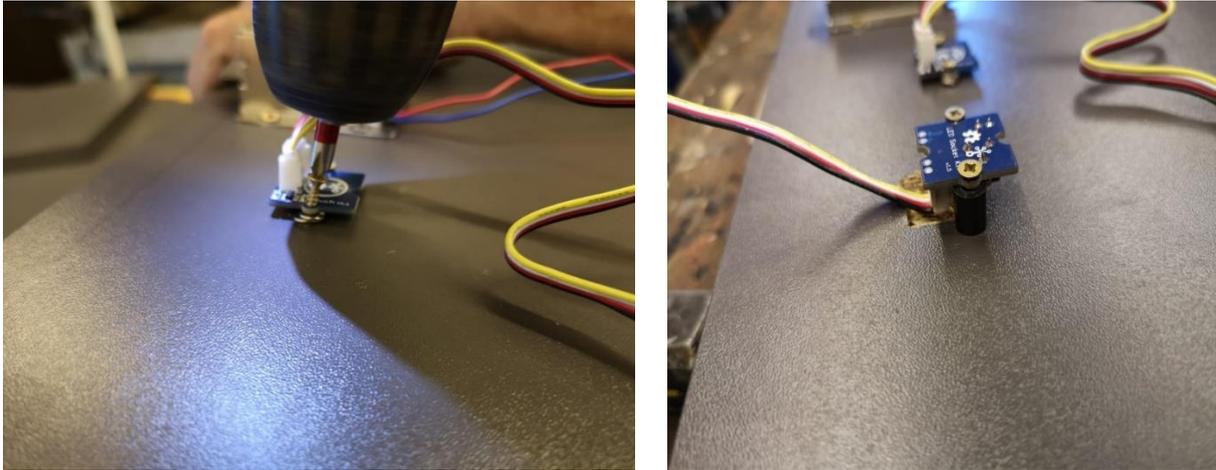
*Abbildung 7: Anbringung Solenoid Door Lock*

Anschließend wird ein daumendickes Loch in die Türfront gebohrt um den Touchsensor einbauen und betätigen zu können. Hier ist zu beachten, dass für den Bau des Prototyps aus Kostengründen der Touchbutton anstelle des Fingerabdrucksensors benutzt wird. Dies geschieht mit Hilfe eines Kegelbohrers, welcher dafür sorgt, dass das Loch von außen nach innen immer schmaler wird.



*Abbildung 8: Kegelbohrer für Loch des Touchbuttons*

Hinter dieses Loch wird von innen der Touchsensor mit Schrauben befestigt. Wichtig ist hierbei, dass man nur das silberne Touchfeld sieht und man von außen keinerlei Zugriff auf die Kontakte hat, da man ansonsten das ganze System aushebeln kann und den Tresor auch ohne Touch bzw. den richtigen Fingerabdruck öffnen könnte.



*Abbildung 9: Anbringung Touchsensor und LED*

Nachdem der Touchsensor angebracht wurde, wird nun ein weiteres kleineres Loch für die LED Leuchte gebohrt. Hier muss man den Sensor unterbauen, damit er stabil am Türgehäuse bleibt und die LED Leuchte optimal mit der Vorderseite der Tür abschließt. Außerdem werden hier mit dem Scharbeisen ein paar Millimeter entfernt, damit der Anschluss des Sensors nicht beschädigt wird und der Abstand durch die Unterlegung mit Plastikröhrchen gewährleistet werden kann.



*Abbildung 10: Fertige Vorderseite des Tresors*

Anschließend wird das Relay und der Arduino Uno eingebaut. Der Arduino wird mit Schrauben befestigt und das Relay wird genauso wie der Touchsensor mit Schrauben und Unterlegscheiben an der Innentür befestigt.

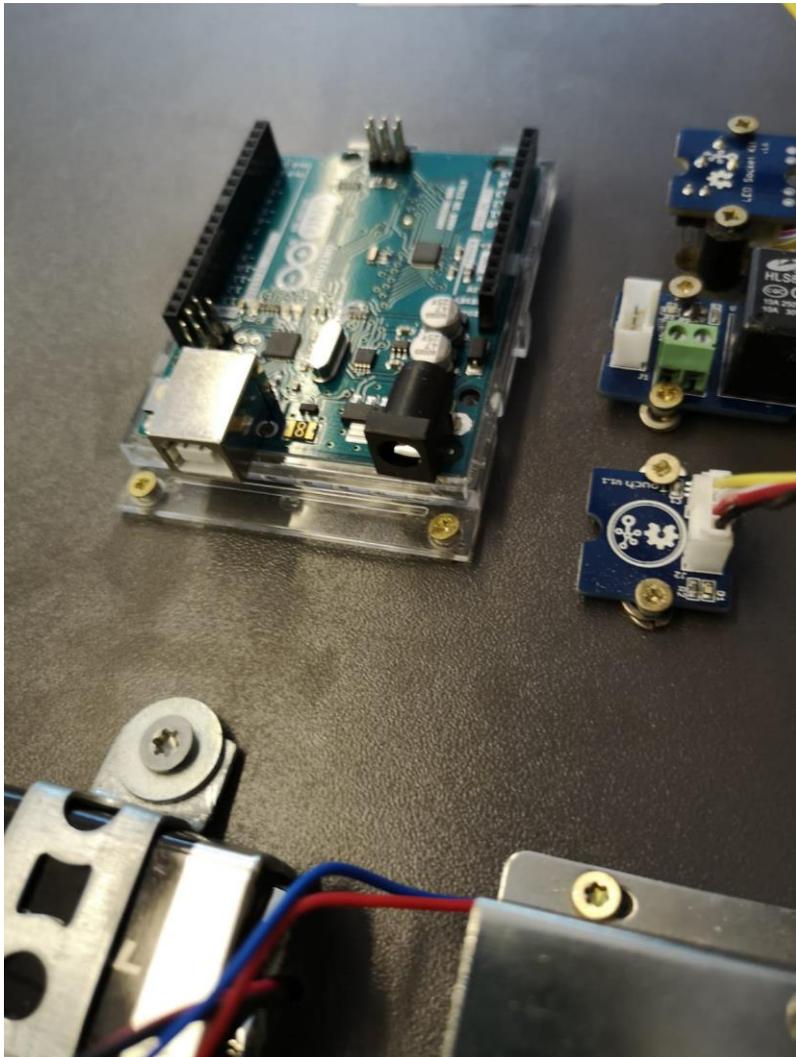


Abbildung 11: Befestigung Arduino

Nachdem alle Komponenten verbaut wurden gilt es die Batterien einzubauen um die nötige Stromversorgung zu gewährleisten. Hierzu ist eine 12 Volt Batterie notwendig, da für den Solenoid Door Lock die Leistung einer 9 Volt Batterie nicht ausreicht. Dies geschieht mit zwei Lochbändern, die mit Unterlegscheibe und Schraube in das Gehäuse gebohrt werden. Durch die Befestigung mit Lochbändern haben sowohl die 9 Volt als auch die 12 Volt Batterie einen festen Halt und können bei Bedarf auch einfach und ohne großen Aufwand ausgetauscht werden. Die 12 Volt Batterie wurde durch sogenannte Schuhe ein- und aussteckbar gemacht, sodass die Batterie keine Leistung verliert, wenn der Tresor nicht in Betrieb sein sollte. Dafür benötigen wir eine Presszange und Metallschuhe, die an die Kabel angeschlossen werden.

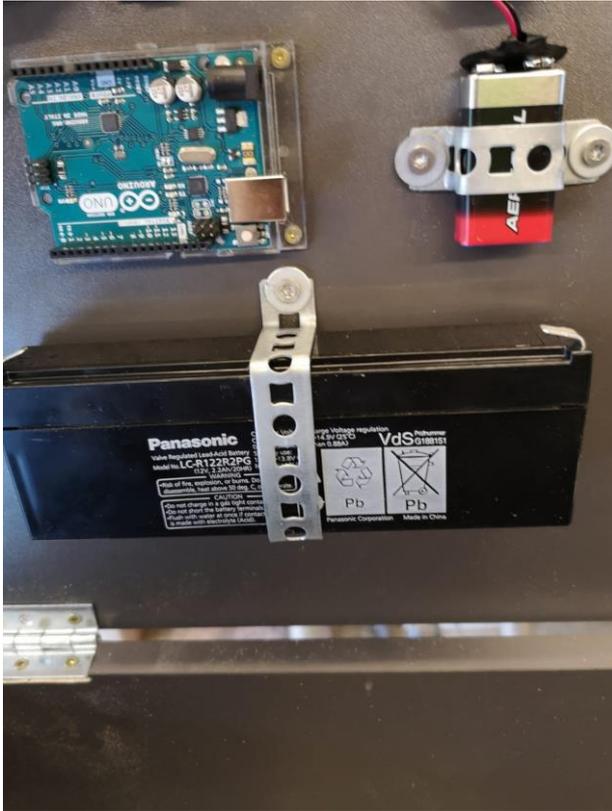


Abbildung 12: Anbringung der Batterien mit Lochbändern



Abbildung 13: Presszange und Verkleidung des Kabels mit Metallschuh

### 3.3.4 Verkabelung der Sensorik und Elektronik

Nachdem die einzelnen Teile in die Tür des Tresors eingebaut wurden, muss nun eine 12V DC Batterie mit dem Door Lock verbunden werden. Die einzelnen Teile und Sensoren werden nun an den Minicomputer/Controller mit den jeweiligen Pins angeschlossen. Das rote Kabel geht an den Minuspol und das Schwarze Kabel muss an den Pluspol der 12V Batterie angeschlossen werden, um den Stromkreislauf herstellen zu können.

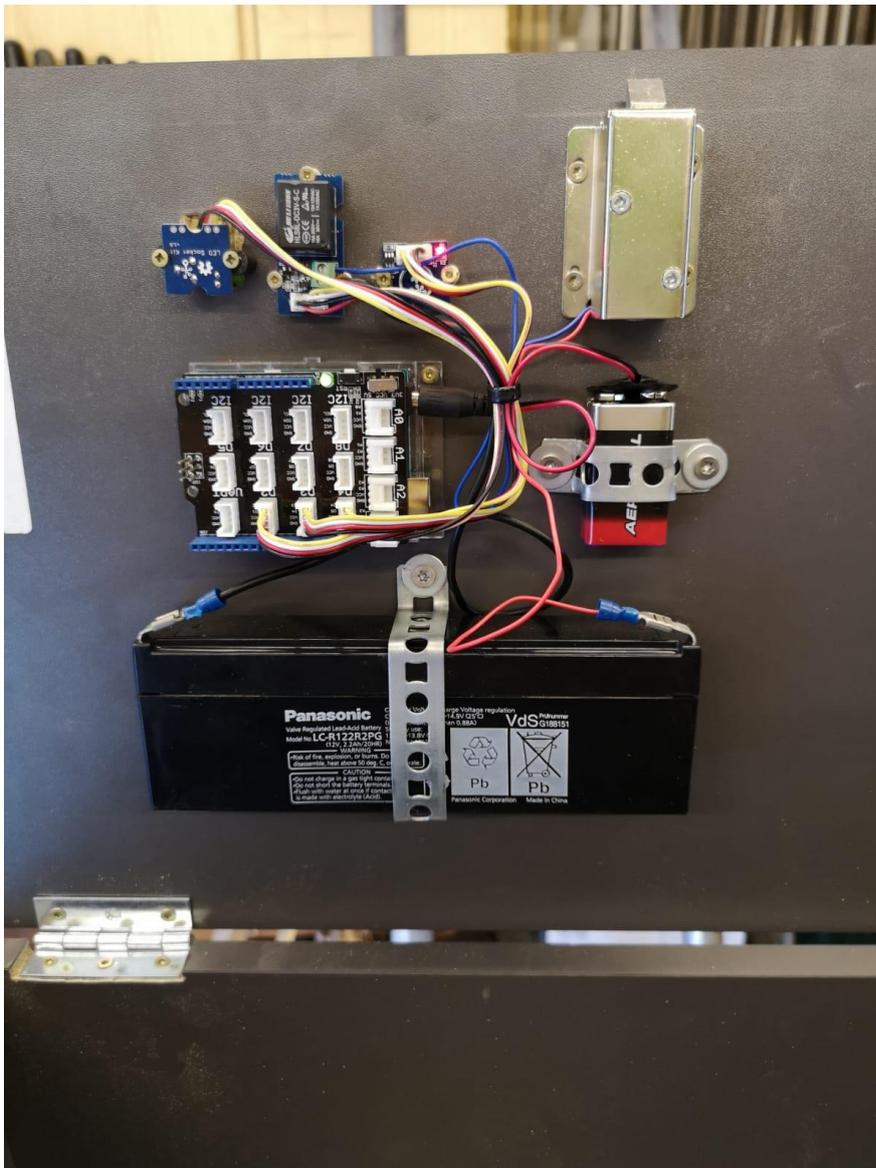


Abbildung 14: Innenseite der Tür

### 3.3.5 Einbau Tragefunktion

Um den X-Safe mobil zu machen wird nun das Trageelement eingebaut - eine abnehmbare Rucksackfunktion. Hier werden die Tragegurte an Karabinerhaken befestigt, sodass man jederzeit die Möglichkeit hat den Tresor auch als Rucksack von A nach B zu befördern. Die Karabinerhaken kann man ganz einfach ein- und aushängen. Dies ermöglicht ein eleganteres Design. Hierfür benötigen wir Winkel, Akkuschauber, Schrauben, Karabiner und Tragegurte.



Abbildung 15: Anbringung Tragegurte

### 3.3.6 Anbringen des Teamnamens auf der Seite des X-Safe.

Um das Produkt zu branden wird eine Schablone erstellt, sodass man später mit einer Spraydose den Namen auf die Außenseite des Tresors sprühen kann. Außerdem muss der Tresor gut abgeklebt werden. Dies verhindert, dass die Farbe an unerwünschten Stellen aufgetragen wird.

Hierzu benötigen wir: eine Schablone, Skalpell, Sprühdose (weiß, matt), Zeitung, Tesafilm.



Abbildung 16: Anbringung Teamname



Abbildung 17: Fertiger Safe

## 4. Programmierung

Vor dem Start der Programmierung müssen sämtliche Teile, die für die Funktionsweise entscheidend sind angebracht werden. Dies bedeutet wie auch in Abbildung 18 bildlich dargestellt muss die LED an D3 und der Touch bzw. Fingerprintsensor an D2 angeschlossen werden. Zuletzt wird noch das Relay an D4 angeschlossen und hierüber der Solenoid Door Lock verbunden.

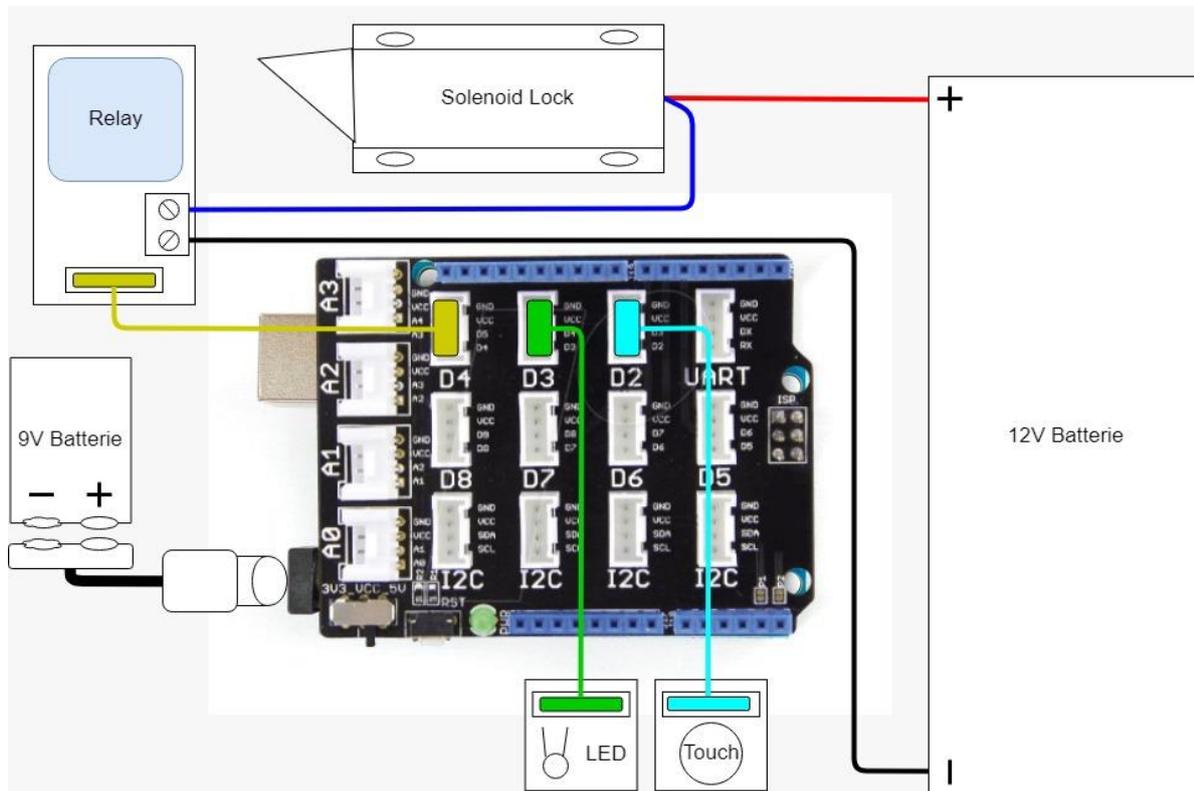


Abbildung 18: Schematischer Aufbau Arduino

Der nachfolgend in Abbildung 19 gezeigte Sketch lässt sich in drei Teile gliedern. Der erste Teil erstreckt sich von Programmzeile 1 „int buttonPin = 2“ bis Programmzeile 4 „int buttonState = 0“. Der zweite Teil umfasst die Bereiche von Programmzeile 5 „void setup ()“ bis Programmzeile 10 „void setup()“. Der letzte Teil geht von Programmzeile 11 „buttonState = digitalRead(buttonPin)“ bis zum Ende.

```
int buttonPin = 2;
int ledPin = 4;
int relayPin = 3;

int buttonState = 0;

void setup() {

  pinMode(relayPin , OUTPUT);
  pinMode(ledPin, OUTPUT);

  pinMode(buttonPin, INPUT);
}

void loop() {

  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {

    digitalWrite(ledPin, HIGH);
    digitalWrite(relayPin, HIGH);
    delay(3000);

  } else {

    digitalWrite(ledPin, LOW);
    digitalWrite(relayPin , LOW);

  }
}
```

Abbildung 19: Gesamter Programmcode

Der erste Teil des Codes geht von Programmzeile 1 „int buttonPin = 2“ bis Programmzeile 4 „int buttonState = 0“ und zeigt nochmals auf in welche Pins die Komponenten eingesteckt wurden. Mithilfe des Befehls „int“ werden die Pins definiert.

Der Arduino stellt maximal 5 Volt zur Verfügung. Das Solenoid Door Lock benötigt jedoch zum Öffnen und Schließen 12 Volt. Um den Strom von der Batterie an das Solenoid Door Lock weiterzuleiten wird ein Relay zwischengeschaltet. Das Relay sorgt dafür, dass ein Output (Solenoid Door Lock) ein- und ausgeschaltet wird. Des Weiteren muss der Touchsensor mit dem Befehl „intbuttonState = 0“ auf Null geschaltet werden. Null gibt hierbei den Ruhezustand des Touchbuttons an. Bei Berührung des Buttons geht dieser in den aktiven Zustand über.

In der folgenden Abbildung 20 wird der eben beschriebene Teil des Programmcodes nochmals aufgezeigt.

```
int buttonPin = 2;
int ledPin = 4;
int relayPin = 3;

int buttonState = 0;
```

Abbildung 20: Programmzeilen 1 bis 4 des Programmcodes

In den Programmzeilen 5 „void setup()“ bis Programmzeile 10 „void loop()“ werden die Inputs und Outputs definiert. In diesem Fall werden Relay und LED mit Hilfe des Befehls „pinMode“ als Output deklariert. Der Touchbutton wird, ebenfalls mit Hilfe des Befehls „pinMode“, als Input definiert. Die Funktion Input besagt, dass ein Pin eine Information an den Arduino überträgt und dadurch einen Output auslöst. In unserem Fall bedeutet das: Man betätigt den Touchsensor, dieser gibt die Information an den Arduino weiter und löst die Outputs LED-Pin und Relay-Pin aus. Die LED leuchtet also grün auf und der Relay überträgt die Information an das Solenoid Door Lock, welches sich öffnet. Die folgende Abbildung zeigt den eben beschriebenen Programmcode.

```
void setup() {
    pinMode(relayPin , OUTPUT);
    pinMode(ledPin, OUTPUT);

    pinMode(buttonPin, INPUT);
}

void loop() {
```

Abbildung 21: Programmzeilen 5 bis 10 des Programmcodes

Im letzten Teil, von der Programmzeile 10 „void loop ()“ bis Programmzeile 20 „}“ werden dem Arduino die Befehle gegeben, welche dauerhaft, also in einer Dauerschleife, ausgeführt werden. Hier wird dem Arduino der Befehl gegeben den digitalen Zustand des Touchbuttons zu lesen. Dieser ist entweder HIGH (es werden 5 Volt übertragen) oder LOW (es wird nichts übertragen). Die Zeile „buttonState = digitalRead(buttonPin);“ sagt genau das eben beschriebene aus.

In der nachfolgenden Funktion „if (buttonState == HIGH) {“ wird gezeigt, dass wenn der Touchsensor gedrückt wird (HIGH), leuchtet die LED-Leuchte und das Relay wird aktiviert. Des Weiteren ist über die Funktion „delay3000“ festgelegt, dass das Schloss und die LED drei Sekunden lang geöffnet sind bzw. leuchten, bevor sie wieder in den Ursprungszustand

zurückgehen. Über den Befehl „digitalWrite“ wird definiert, dass ein bestimmter Output ausgelöst wird. In der nachfolgenden Abbildung wird der eben beschriebene Code nochmals verbildlicht.

```
if (buttonState == HIGH) {  
  digitalWrite(ledPin, HIGH);  
  digitalWrite(relayPin, HIGH);  
  delay(3000);  
}
```

Abbildung 22: Programmzeile 12 bis 15 des Programmcodes

Ab Programmzeile 16 „} else {“ bis zum Ende des Programmcodes wird beschrieben welchen Zustand die Outputs haben, wenn der „buttonState“ nicht „HIGH“ ist. Dies bewirkt, dass sowohl die LED als auch das Relay den Zustand „LOW“ haben und somit nicht aktiviert werden.

```
} else {  
  digitalWrite(ledPin, LOW);  
  digitalWrite(relayPin , LOW);  
}
```

Abbildung 23: Programmzeile 16 bis 20 des Programmcodes

## 4.1 Zusatzprogrammierung Fingerprint

In der Programmierung, die im Punkt 4 beschrieben wurde, wurde der Input durch einen Touchbutton dargestellt. Der X-Safe soll in Realität jedoch mit einem Fingerprintsensor ausgestattet werden. Im Folgenden wird kurz aufgezeigt, wie das Einscannen der Finger funktioniert. Der Programmcode für den Fingerprint wird im Anhang dieser Ausarbeitung angehängt.

Um den Fingerprint Sensor programmieren zu können, muss eine Bibliothek eingebunden werden. Dies erfolgt in wenigen kleinen Schritten. Man klickt auf „Sketch-Bibliothek einbinden“-„Bibliothek verwalten“.

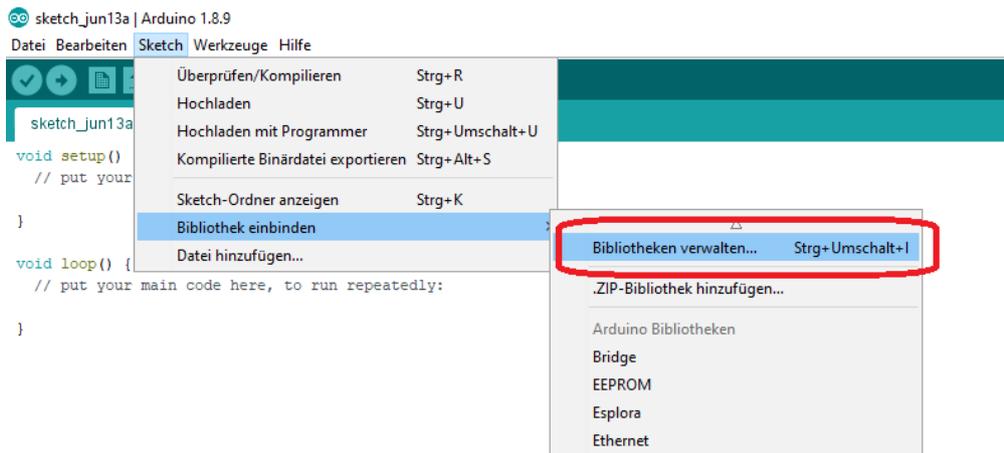


Abbildung 24: Herunterladen der Bibliothek für den Fingerprintsensor

Nachdem man auf „Bibliotheken verwalten“ geklickt hat, erscheint ein neues Fenster mit dem Namen „Bibliotheksverwalter“, siehe Abb. 25. Die für die Programmierung entscheidende Bibliothek nennt sich „Adafruit Fingerprint Sensor Library“. Diese muss nun heruntergeladen und installiert werden.

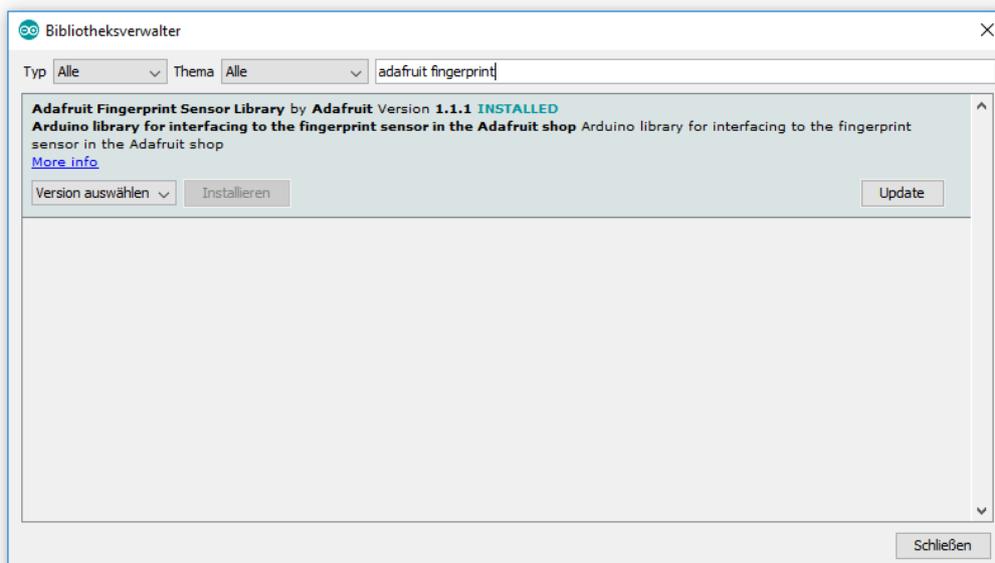


Abbildung 25: Bibliotheksverwalter

Nach der Installation der Bibliothek können nun die Finger eingescannt werden. Hierfür muss man auf „Datei“-„Beispiele“-„Adafruit Fingerprint Sensor Library“-„Enroll“ klicken, siehe Abb. 26.

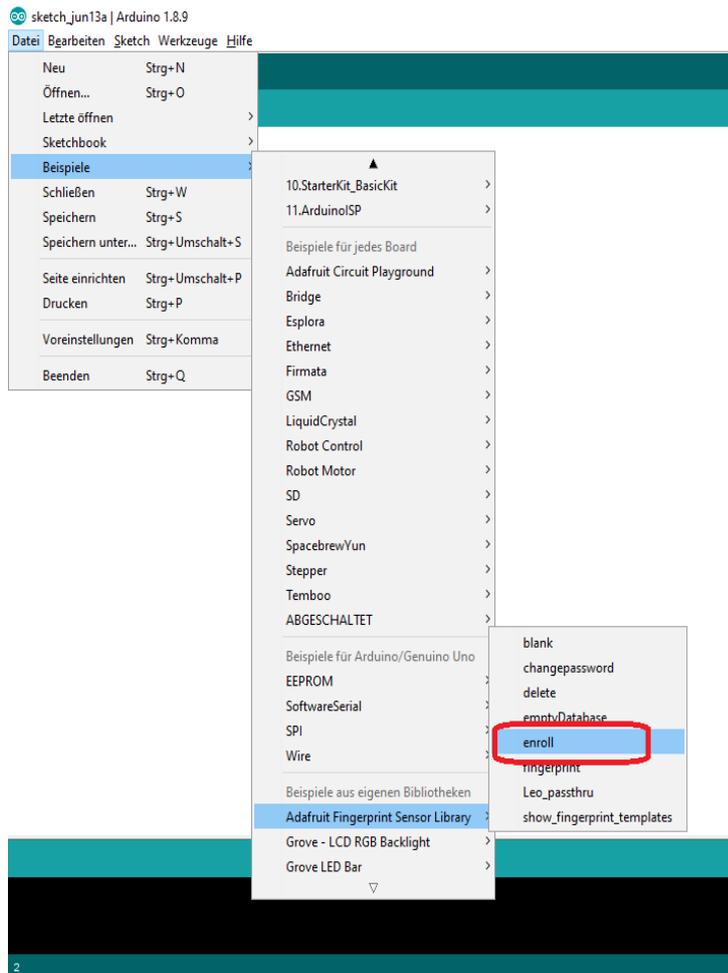


Abbildung 26: Enroll-Funktion

Ein neuer Sketch mit einem neuen Code ist erschienen. Die Zusammensetzung dieses Codes ist irrelevant, denn er hat lediglich die Funktion, einen oder mehrere Fingerabdrücke einzuscannen und zu speichern. Dieser Code kann wie in Abbildung 27 gezeigt wird, auf den Arduino übertragen werden.

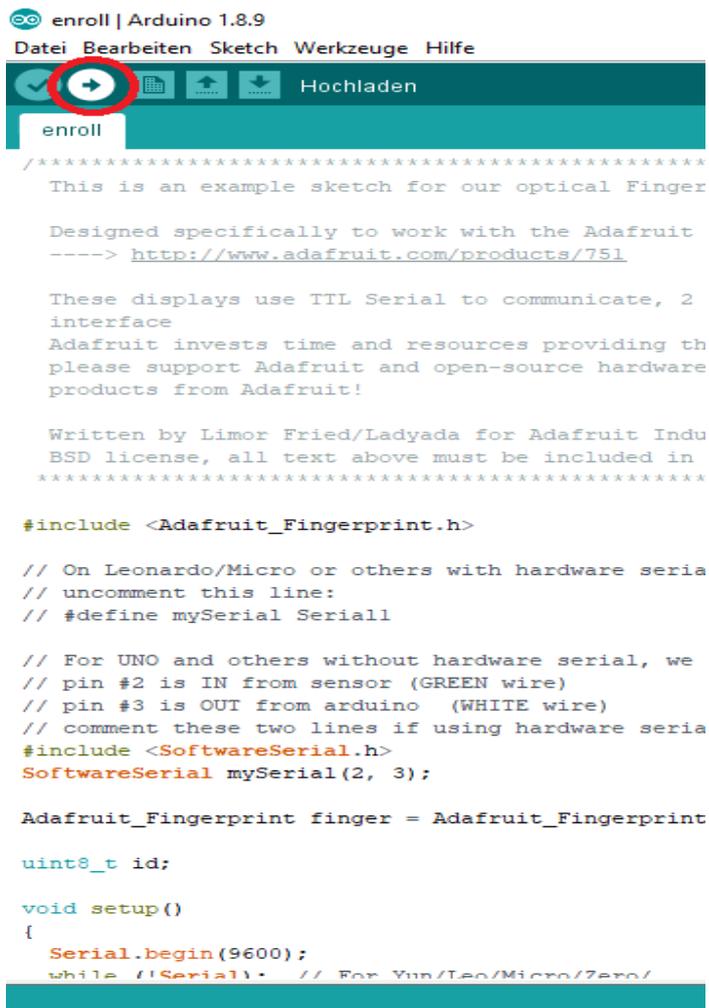


Abbildung 27: Einspeichern Finger

Um den Fingerabdruck zu speichern, muss jeder einzelne Finger mit einer ID versehen werden. Insgesamt ist es möglich 1 bis 127 Finger einzuscannen. Dementsprechend kann beim Einscannen nun die Wahl zwischen einer ID zwischen 1 und 127 getroffen werden. Man wählt also eine Zahl aus und drückt auf „Send“ (s. Abb. 28). Sobald dieser Befehl ausgeführt ist, muss der einzuscannende Finger für circa drei Sekunden auf den Fingerscan aufgelegt werden. Danach wird der Vorgang nochmals wiederholt und der Finger ist eingescannt.

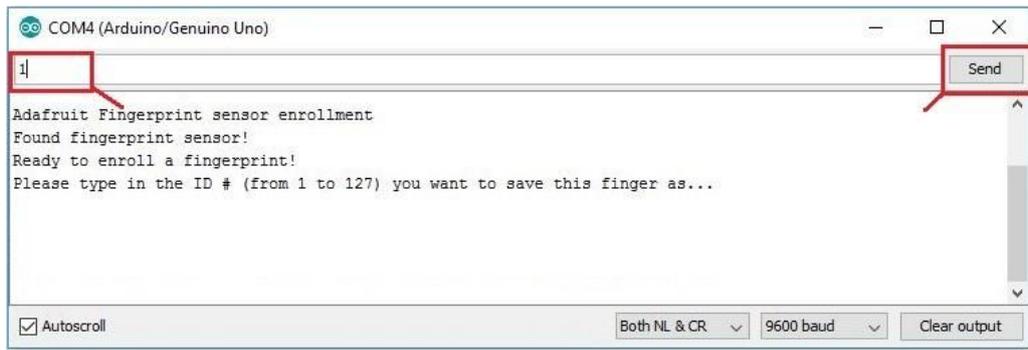


Abbildung 28: Übertragung an Arduino

## II. Anhang

### - Finger Print Sensor - Programmiercode

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

int getFingerprintIDez();

//pins deklarieren
int pinRelay =12;    // Relay schaltet den Solenoid an und aus

// pin #2 ist IN vom sensor
// pin #3 ist OUT vom arduino
SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  Serial.begin(9600);
  Serial.println("fingertest");

  // SET PIN AS OUTPUT
  pinMode(12, OUTPUT);           //angeben, dass durch Relay gesteuerte Solenoid ein OUTPUT ist

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1); { delay(1); }
  }
}

finger.getTemplateCount();
Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");
Serial.println("Waiting for valid finger...");
```

---

```

}

void loop()          // immer wieder laufen
{

  getFingerprintIDez();
  delay(50);          //wie häufig soll die LED blinken

}

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECEIVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }

  // OK success!

  p = finger.image2Tz();
  switch (p) {
    case FINGERPRINT_OK:

```

---

```

        Serial.println("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");
        return p;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Communication error");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
        return p;
    default:
        Serial.println("Unknown error");
        return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

// found a match!

Serial.print("Found ID #");
Serial.print(finger.fingerID);

Serial.print(" with confidence of ");
Serial.println(finger.confidence);

return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) return -1;

    // found a match!
    Serial.print("Found ID #");
    digitalWrite(12, HIGH); //Relay einschalten
    delay(3000); //3 Sekunden warten
    digitalWrite(12, LOW); //Relay ausschalten
    Serial.print(finger.fingerID);
    Serial.print(" with confidence of ");
    Serial.println(finger.confidence);
    return finger.fingerID;
}

```

### III. Literaturverzeichnis

Freihoff 24 Sicherheitsservice (2018), Kriminalstatistik 2017: *Weniger Einbrüche dank umfangreicher Präventionsmaßnahmen – Anteil der Kriminalitätsbereiche an Straftaten gesamt*, <https://www.freihoff.de/news/kriminalstatistik-2017/> (aufgerufen am 15.06.2019)

Statista (2014), *Wo schlafen Sie meistens auf Urlaubsreisen?*, <https://de.statista.com/statistik/daten/studie/288015/umfrage/beliebteste-uebernachtungsmoeglichkeiten-auf-urlaubsreisen-nach-geschlecht/> (aufgerufen am 16.06.2019)

Statista (2017), *Geschäftsreisen nehmen zu – Anzahl der Geschäftsreisen von deutschen Unternehmen (in Mio.)*, <https://de.statista.com/infografik/11008/immer-mehr-geschaeftsreisen/> (aufgerufen am 16.06.2019)