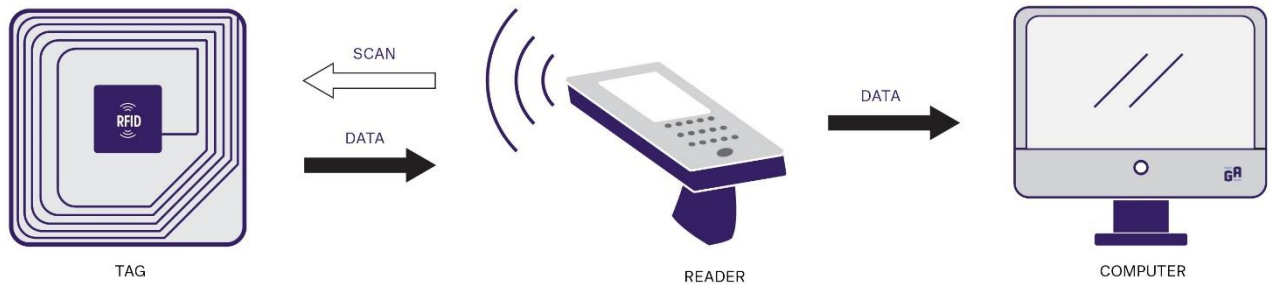


# Schlüsselmanagement-System

basierend auf RFID-Technologie



Diplomand: Kelmendi Egzon  
Diplomlehrer: Urs Gloggner  
Schule: TEKO Olten  
Klasse: O-TEL-20-S-a  
Ausbildung: Dipl. Elektrotechniker HF  
Abschlussjahr: 2023

# 1 Inhaltsverzeichnis

|        |   |    |
|--------|---|----|
| 1      | Inhaltsverzeichnis .....  | 2  |
| 2      | Management Summary .....  | 4  |
| 3      | Lebenslauf .....  | 5  |
| 4      | Qualifikationsprofil .....  | 6  |
| 5      | Projektinitialisierung .....                                      | 8  |
| 5.1    | Projektauftrag .....  | 8  |
| 5.2    | Aufgabenstellung von TEKO Olten .....                             | 10 |
| 5.3    | Ideenfindung für die Diplomarbeit .....                           | 11 |
| 5.4    | Pflichtenheft .....   | 12 |
| 5.4.1  | Einleitung .....  | 12 |
| 5.4.2  | Richtziel.....  | 12 |
| 5.5    | Zielscheibe.....  | 14 |
| 6      | Projektplanung.....   | 15 |
| 6.1    | Projektstrukturplanung .....                                      | 15 |
| 6.2    | Projekttafelplan/ Projektstrukturplan.....                        | 16 |
| 6.3    | Auswahl des Programms für die Erstellung der GUI.....             | 18 |
| 7      | Projektrealisierung.....  | 19 |
| 7.1    | Beschaffung von Hardware .....                                    | 19 |
| 7.1.1  | Materialliste und Infos.....                                      | 19 |
| 7.2    | Beschaffung von Software .....                                    | 24 |
| 7.2.1  | Programmierung Vorgang.....                                       | 24 |
| 7.2.2  | Flussdiagramm der Software für Arduino .....                      | 27 |
| 7.2.3  | Code für das Schreiben der Daten auf RFID-Tag.....                | 28 |
| 7.2.4  | Code für das Lesen der Daten auf RFID-Tag.....                    | 31 |
| 7.3    | Erstellung der Benutzeroberfläche .....                           | 32 |
| 7.3.1  | Flussdiagramm der Software für Benutzeroberfläche .....           | 32 |
| 7.3.2  | Code für das Erstellen der Benutzeroberfläche .....               | 33 |
| 7.4    | Der Prototyp .....  | 38 |
| 7.4.1  | Der Aufbau bei Garage Klaus.....                                  | 39 |
| 7.5    | Inbetriebnahme-Protokoll.....                                     | 40 |
| 7.6    | Ergebnisse der Inbetriebnahme .....                               | 40 |
| 8      | Projektabschluss .....  | 41 |
| 8.1    | Evaluation der Zielerreichung .....                               | 41 |
| 8.2    | Testberichte .....  | 42 |
| 8.2.1  | Testbericht Sascha Rytz: Kundendienst, Garage Klaus .....         | 42 |
| 8.2.2  | Testbericht: Marouen Garbaa: Automobilfachmann, Garage Klaus..... | 43 |
| 8.3    | Reflexion Weg zum Ziel.....                                       | 44 |
| 8.4    | Verdankung.....   | 45 |
| 8.5    | Eigenständigkeit-Erklärung.....                                   | 46 |
| 9      | Verzeichnisse .....   | 47 |
| 9.1    | Abbildungen .....   | 47 |
| 9.2    | Tabellenverzeichnis .....   | 48 |
| 9.3    | Linkverzeichnis .....   | 48 |
| 10     | Anhang .....  | 49 |
| 10.1   | Statusberichte .....  | 49 |
| 10.2   | Vollständiger Programm Code.....                                  | 55 |
| 10.2.1 | Arduino Teil.....   | 55 |

10.2.2 Windows Forms Teil..... 65

## 2 Management Summary

Die vorliegende Arbeit behandelt das Thema Schlüsselmanagement-System, basierend auf RFID-Technologie. In vielen Garagen stellt das Schlüsselmanagement bereits ein Problem dar, weshalb bereits zahlreiche Produkte entwickelt wurden. Allerdings erfüllen diese Produkte nicht die spezifischen Anforderungen einiger Garagen. Daher wurde dieses Projekt als Diplomarbeit gewählt. In Garagen kommt es häufig zu Schlüsselverwechslungen aufgrund der Vielzahl von Schlüsseln in der Umgebung, was zu Zeitverlust führt. Das Ziel war daher, ein digitalisiertes Schlüsselmanagement-System zu entwickeln, das mithilfe der RFID-Technologie die gewünschten Fahrzeughalterdaten anzeigt, Schlüsselverwechslungen verhindert und die Suche nach dem richtigen Schlüssel für das gewünschte Auto verkürzt.

Zunächst wurde das Problem im Detail analysiert und Überlegungen angestellt. Dabei wurde festgestellt, dass ein RFID-Tag eine äußerst praktische Lösung darstellt, da er leicht an einem Autoschlüssel angebracht werden kann und ein Lesegerät nicht viel Platz benötigt. Da Arduino bereits RFID-Technologie unterstützt, wurde dies als Grundlage für das Projekt verwendet. Es musste jedoch noch eine Benutzeroberfläche erstellt werden, um die Bedienung zu vereinfachen und verständlich zu gestalten.

Der Programmcode wurde erweitert und angepasst, um die gespeicherten Daten auszulesen, wenn der RFID-Tag auf das Lesegerät gelegt wird, und diese auf der Benutzeroberfläche erfolgreich anzuzeigen. Dies ermöglicht nicht nur eine schnellere Identifizierung des Fahrzeughalters, sondern auch das praktische Beschreiben der RFID-Tags, die aufgrund ihrer Stabilität kaum beschädigt werden oder verloren gehen. Die Benutzeroberfläche wurde mithilfe von Windows Forms erstellt, einer von Microsoft entwickelten Plattform, die Benutzern ermöglicht, eine Benutzeroberfläche per Drag & Drop zu erstellen und dann die Funktionen mithilfe der Programmiersprache C# zu programmieren.

Zukünftige Schritte umfassen die Implementierung eines zweiten Lesegeräts, um den Vorgang auch bei der Rückgabe des Fahrzeugs beim Empfang zu erleichtern.

Dieses RFID-basierte Schlüsselmanagement-System bietet eine effiziente Lösung für Garagen, insbesondere für Garage Klaus, und verbessert die Schlüsselverwaltung erheblich. Es spart Zeit und reduziert Schlüsselverwechslungen, was sowohl für die Garage als auch für die Kunden einen Mehrwert darstellt.

### 3 Lebenslauf



## Kelmendi Egzon

Adresse: 3186 Düringen, Freiburg/Fribourg

Geb.: 28.06.1996

Nationalität: Kosovo

E-Mail: [egzon.kelmendi@edu.teko.ch](mailto:egzon.kelmendi@edu.teko.ch)

Linkedin: [Linkedin](#)

### Beruflicher Werdegang

- April 2019 – heute **Automobilfachmann, TEC, Stellvertretender Werkstatt Chef**  
Garage Klaus, 3186 Düringen
- August 2018 – März 2019 **Mitarbeiter Montage**  
Stromer AG, 3173 Oberwangen bei Bern
- August 2014 – Juli 2018 **Lehre als Automobilfachmann**  
Ledermann Automobile, 3182 Ueberstorf

### Bildung

- 2002 – 2011 **Orientierungsschule** Kosovo  
2014 – 2018 **Ausbildung zur Automobilfachmann**  
GIBS Freiburg/Fribourg  
**Abschluss:** Automobilfachmann mit Eidg. Fachausweis  
2019 – 2023 **Dipl.- Techniker HF, Elektrotechnik**  
TEKO, Olten  
**Abschluss:** Dipl. Techniker HF Elektrotechnik  
2022 – 2025 Ausbildung zur **TEC (Tecnico Esperto di Concessionaria)**  
**Abschluss:** TEC (Tecnico Esperto di Concessionaria)  
FCA Switzerland (Fiat Chrysler Alfa Romeo)

### Sprachen

- Deutsch 2. Muttersprache  
Englisch Verhandlungssicher (B2 Abschluss TEKOL Olten)  
Albanisch Muttersprache  
Französisch A1

### Freizeit

- Bücher lesen  
Fitness  
Wald Spaziergang  
Kino

## 4 Qualifikationsprofil

### **Menschen führen**

#### (Prozess 1)

Als stellvertretender Werkstattchef habe ich das Team geleitet. Klare Erwartungen wurden festgelegt, Aufträge wurden vorbereitet und verteilt, Mitarbeiterprobleme wurden diskutiert und gelöst, das Team wurde motiviert und koordiniert.

### **Entscheidungen fällen**

#### (Prozess 2)

Jeden Tag müssen Entscheidungen getroffen werden, beispielsweise in Bezug darauf, welche Autos als erste in die Werkstatt auf die Hebebühne dürfen, um sicherzustellen, dass die Kunden nicht lange warten müssen. Dies ist oft mit den Verzögerungen aufgrund von Materialmangel bei den Lieferungen verbunden. Zusätzlich dazu muss ich wiederholt entscheiden, welche Teile priorisiert ausgetauscht werden müssen, um einen Fehler zu beheben, insbesondere wenn unklar ist, welcher Fehler vorliegt.

### **Sich sprachlich verständigen**

#### (Prozess 4)

Wenn Lehrlinge technische oder theoretische Fragen hatten, die fachliche Erklärungen im Zusammenhang mit Autos erforderten, konnte ich aufgrund meines umfangreichen Wissens diese Fragen auch mit Fachterminologie beantworten. Ebenso war ich in der Lage, Mitarbeiterinnen und Mitarbeitern dabei zu helfen, Herstelleranweisungen, wie beispielsweise Rückrufe, in verständliche Alltagssprache zu übersetzen, wenn sie Schwierigkeiten mit den Fachbegriffen hatten.

### **Wirkungsvoll präsentieren und Kommunizieren**

#### (Prozess 5)

Wenn ein Kunde ein Problem hatte und Interesse daran zeigte, das Problem zu verstehen, konnte ich mit meinem Wissen dies gut erklären und auch zeigen, wie das Vorgehen sein sollte, damit er auch einen Überblick hatte.

### **Geschäftsziele erreichen**

#### (Prozess 7)

Es kam oft vor, dass ich Aufgaben schneller erledigen musste, wodurch neue Arbeitszeiten für bestimmte Aufgaben festgelegt werden mussten, um sicherzustellen, dass die Firma nicht zu viel Zeit in eine Aufgabe investiert und die Kosten von niemandem übernommen werden mussten.

## **Umfeld berücksichtigen**

(Prozess 8)

Ich musste oft überprüfen, welche Materialien entleert und nicht benötigt wurden, sowie die bereits verwendeten chemischen Materialien wie Altöl, Kältemittel, Kompressoren Öl, Bremsflüssigkeit und Alteisen, um sicherzustellen, dass sie umweltgerecht entsorgt wurden. Außerdem war es wichtig, das gebrauchte Steuergerät ordnungsgemäß zu entsorgen oder es gegebenenfalls an den Hersteller zurückzusenden.

## **Probleme analysieren und lösen**

(Prozess 9)

Der Alltag eines Autofachmanns ist voll von der Lösung von Problemen an Autos. Meistens handelt es sich um wiederkehrende Probleme, für die bereits eine vorgegebene Lösung existiert. In einigen seltenen Fällen trifft dies jedoch nicht zu. In solchen Situationen muss ich eine vorgeschriebene Vorgehensweise anwenden, wie zum Beispiel bei elektrischen Problemen. Zuerst messe ich die Stromversorgung zu den Steuergeräten und überprüfe anschließend die Sicherungen, um Kabelbrüche festzustellen und diese zu reparieren.

## **Sich persönlich weiter entwickeln**

(Prozess 10)

Ich musste während meines Studiums auch ein weiteres Studium beginnen, nämlich T.E.C für die Autohersteller Alfa Romeo und JEEP, um stets auf dem neuesten Stand der neuen Technologien und Lösungswege für technische Probleme zu bleiben. Außerdem hatte ich die Gelegenheit, die Programme der FCA-Gruppe kennenzulernen.

## **In betrieb setzen**

(Prozess 14)

Ich durfte verschiedene Maschinen in Betrieb setzen, die für Autos mit neueren Technologien verwendet wurden und eine Kalibrierung oder neue Programmierung erforderten.

## **Elektrotechnische Anlagen Unterhalten**

(Prozess 15)

Da elektrische Komponenten regelmäßig neue Software erhalten, ist es auch möglich, hardwareseitig neue Komponenten einzusetzen. Ich konnte Komponenten ersetzen, sofern sie mit der neuen Software kompatibel waren, wie beispielsweise Sensoren oder Stellmotoren, usw.

## 5 Projektinitialisierung

### 5.1 Projektauftrag

|                      |   |
|----------------------|---|
| <b>Projekttitle:</b> | Schlüsselmanagement-System basierend auf RFID-Technologie |
|----------------------|---|

|                             |                |
|-----------------------------|----------------|
| <b>Projektauftraggeber:</b> | Kelmendi Egzon |
| <b>Projektleiter:</b>       | Kelmendi Egzon |

|                    |            |              |            |
|--------------------|------------|--------------|------------|
| <b>Projektdate</b> |            |              |            |
| <b>Start:</b>      | 11.09.2023 | <b>Ende:</b> | 23.10.2023 |

|   |   |
|---|---|
| <b>Projektbeschreibung</b>  |   |
| <b>Ausgangslage / Projektbegründung:</b>  | Die gegenwärtige manuelle Schlüsselverwaltung in unserer Garage ist zeitaufwendig, ungenau und führt gelegentlich zu Verwirrungen bei der Fahrzeugidentifikation.   |
| <b>Sinn und Zweck / Nutzen:</b>   | Die Arbeit ermöglicht es mir, mein erlerntes Wissen, meine Fähigkeiten in den Bereichen Projektmanagement, Forschung, Entwicklung, Implementierung und Dokumentation zu erweitern und meine Kreativität in der Praxis anzuwenden und unter Beweis zu stellen. |
| <b>Projektrichtziel:</b>  | Das Richtziel der Diplomarbeit ist es, ein funktionierendes digitales Schlüsselmanagement-System für die Garage zu entwickeln und zu implementieren.  |
| <b>Endergebnisse</b>  | <b>Erfolgskriterien</b>   |
| Ein Ablaufplan- und Projektstrukturplan ist erstellt  | Mind. 85% der Aufgabe ist so gelöst, dass es keine rückfragen auftreten.  |
| Eine Benutzeroberfläche als App für Desktop Anwendung ist erstellt  | Benutzer Oberfläche als App ist mit Windows Forms erstellt und erscheint ein App Icon auf Desktop   |
| Es ist eine grafische Lösung erstellt worden anhand von der Bibliothek der Arduino Plattform für RFID Write and Read.     | Die Arduino Bibliotheken verwenden, Code erweitern und anpassen an die Projektziele.  |
| Beim Lesen der Schlüssel Infos sind die Daten auf Benutzeroberfläche zu sehen.  | Programm code ist auf Fundoino geladen und durch eine Verbindung mit Computer erscheinen die Schlüssel Infos auf Windows Forms App.   |
| Das Schreiben der Kunden Daten auf RFID-Tags ist dem Benutzer möglich und wird vereinfacht durch eine Bedienungsanleitung | Im Fall für das erste Bedienung Schwierigkeiten vorkommen das Windows App zu benutzen, ist eine Bedienungsanleitung vorhanden   |
| Das erneute benutzen der RFID-Tags ist möglich  | Die gespeicherten Infos können gelöscht und neu geschrieben, somit wird das RFID-Tag erneut benutzt.  |
| Es liegt ein Testbericht vor  | Folgende Test arten werden durchgeführt: Smoke Tests, Integrationstests, Abnahmetests   |
| Selbstreflexion schreiben nach der Dokumentation gefertigt wurde  | Zeigen wo die Schwierigkeiten waren und was noch zu üben ist  |

|                         |   |
|-------------------------|---|
| <b>Zielgenehmigung:</b> | Die Ziele werden bewilligt.<br><br>Datum:<br>Unterschrift Auftraggeber: |
|-------------------------|---|

|  |                   |
|--|-------------------|
| <b>Projekttyp:</b>   |                   |
| <input type="checkbox"/> Routineprojekt<br><input type="checkbox"/> komplexes Standardprojekt<br><input checked="" type="checkbox"/> Potenzial- / Innovationsprojekt<br><input checked="" type="checkbox"/> Diplomarbeit | <b>Begründung</b> |

|   |  |   |
|---|--|---|
| <b>Projektorganisation:</b>   |  |   |
| <b>Organisationstyp:</b>  |  |   |
| <b>Projektmitarbeiter:</b>  | <b>Name / Vorname / OE</b><br>Kelmendi Egzon | <b>Stellenprozent für Projekt</b><br>100% |
| <b>Steering Committee:</b><br><input checked="" type="checkbox"/> Ja<br><input type="checkbox"/> Nein | Urs Gloggner (Diplomlehrer)                  |   |
| <b>Sonstige Beteiligte:</b>   | Mitarbeiter der Garage die das Gerät Testen. |   |

|                                      |  |
|--------------------------------------|--|
| <b>Projektplanung</b>                |  |
| <b>Projektphasen / Meilensteine:</b> | <ol style="list-style-type: none"> <li>1. Meilenstein Projektauftrag genehmigt</li> <li>2. Meilenstein Abschluss Planung Phase</li> <li>3. Meilenstein: Aufbau funktioniert</li> <li>4. Meilenstein: Abgabe der Dokumentation</li> </ol> |

|                          |  |
|--------------------------|--|
| <b>Projektentscheid:</b> | <input type="checkbox"/> Das Projekt wird bewilligt. |
|                          | <input type="checkbox"/> Das Projekt wird abgelehnt. |
|                          | Begründung:  |
|                          | Datum:   |
|                          | Unterschrift Auftraggeber:                           |

## 5.2 Aufgabenstellung von TEKO Olten

Die Diplomarbeit dient dazu, dein umfangreiches Grundlagenwissen, das du während deines Studiums erworben und in praxisorientierten Arbeiten vertieft hast, auf einen konkreten betrieblichen Auftrag anzuwenden. Dein Ziel besteht darin, diese Aufgabenstellung innerhalb einer vorgegebenen Zeitspanne eigenständig, umfassend und zweckmässig zu lösen. Gleichzeitig sollst du die Ergebnisse sauber dokumentieren, präsentieren und online veröffentlichen. Dies zeigt deine Fähigkeit, theoretisches Wissen in die Praxis umzusetzen und es einer breiteren Öffentlichkeit zugänglich zu machen.

### Ablauf:

Am Beginn des letzten Semesters erfolgt eine umfassende Orientierung durch den Abteilungsvorstand oder die Schulleitung. Dabei erhältst du Informationen zum Ablauf der Diplomarbeit, den einzuhaltenden Terminen sowie den erforderlichen Vorbereitungsarbeiten.

Nach der Orientierung beginnst du damit, ein geeignetes Thema für deine Diplomarbeit zu suchen. Dies sollte eine klar abgrenzbare Aufgabe aus deinem erweiterten beruflichen Umfeld sein, die einen klaren Bezug zu deiner Studienrichtung aufweist. In der Regel handelt es sich bei der Diplomarbeit um eine Einzelarbeit, aber bei umfangreichen Projekten besteht auch die Möglichkeit, in kleinen Teams zu arbeiten.

Die erfolgreiche Durchführung der Diplomarbeit erfordert eine gründliche Planung, sorgfältige Umsetzung und die Einhaltung der Richtlinien der Schule sowie bewährter Phasen des Projektmanagements. Dieser Prozess ist eine wichtige Etappe deines Studiums und ermöglicht es dir, deine Fähigkeiten und Kenntnisse in der Praxis zu demonstrieren.

### 5.3 Ideenfindung für die Diplomarbeit

Die Suche nach dem richtigen Diplomarbeitsthema war eine knifflige Angelegenheit. Mein Ziel war es, ein Thema zu finden, das nicht nur meinen persönlichen Interessen entsprach, sondern auch in der realen Welt Anwendung finden konnte und mir die Möglichkeit gab, etwas Neues zu lernen.

Ich habe eine Weile darüber nachgedacht, aber es war schwierig, eine klare Richtung zu finden. In meiner Ratlosigkeit habe ich sogar meine Kollegen um Vorschläge gebeten, ob ihnen vielleicht etwas einfiele. Leider konnten meine Kollegen, die von Elektrotechnik nicht viel Ahnung hatten, mir nicht wirklich weiterhelfen.

Dann ist etwas Unerwartetes passiert. Kurz vor dem Termin zur Einreichung der Themen musste ich das Auto eines Kunden, das in die Werkstatt zur Reparatur gebracht worden war, an die Mitarbeiter des Kundendienstes zurückgeben, damit sie den Kunden benachrichtigen konnten, dass sein Auto bereit war. Als ich das Auto abgab, wurde ich nach dem Autoschlüssel gefragt und noch wichtiger nach dem Namen des Kunden, dem der Schlüssel gehörte. Doch in diesem Moment konnte ich mich nicht mehr erinnern, da täglich viele Autos kamen und gingen.

In diesem Moment kam mir eine Idee - hier war ein konkretes Problem, das uns schon lange gestört hatte. Ich begann mich zu fragen, ob es eine Lösung dafür geben könnte. Die Idee, dieses Problem anzugehen, begann in mir zu wachsen.

Ich erinnerte mich an eine Übung, die wir während eines Kurses zu Mikrocontrollern mit Arduino gemacht hatten - es ging um RFID-Tags. Und da kam mir die Idee, dass die RFID-Technologie vielleicht die Antwort auf unser Problem sein könnte.

Ich begann, Ideen zu sammeln und einen Plan zu entwickeln. Wie könnten die Schlüssel mit RFID-Tags beschrieben und gelöscht werden? Welche Informationen könnten auf diesen Tags gespeichert werden, und wie könnten sie gelesen und beschrieben werden? Wer sollte Zugriff auf diese Daten haben, und wie könnte die Sicherheit gewährleistet werden?

Um meine Idee weiterzuentwickeln, begann ich mit Recherchen und Informationen, die mir halfen, meine Vorstellungen zu konkretisieren. Schließlich füllte ich das Antragsformular für das Thema meiner Diplomarbeit auf der Teko-Website aus.

Diese Geschichte zeigt auch, wie aus einem alltäglichen Problem und einer unerwarteten Situation die Idee für meine Diplomarbeit entstanden ist. Sie verdeutlicht den kreativen Prozess der Ideenfindung, der oft unerwartete Wege einschlägt.

## 5.4 Pflichtenheft

### 5.4.1 Einleitung

Derzeit existieren verschiedene Ansätze und Produkte, die mit dem Management das Fahrzeug wie auch normale Gebäude Schlüssel befassen wie z.B.: 1. Elektrische Schlüsselkästen: diese Systeme bieten es den Benutzern einen Schlüssel elektronisch zu verfolgen, identifizieren und zuzuordnen. 2. Cloud-basierte Schlüsselmanagement-Plattformen: Da können Fahrzeugschlüsselerfasst, an den entsprechenden Fahrzeugen zugeordnet werden und verfolgt.

Als Grundlage für die Diplomarbeit dient, die bereits problematisch und mühsame manuelle Schlüssel Management in Garagen, die schon oft schlecht organisiert sind und oft zu Schlüsselverlust oder Verwechslung kommen kann. Darüber hinaus wie oben erwähnt gibt es bereits verschiedene Lösungen auf dem Markt, um eine Schlüssel Management zu digitalisieren, die aber nicht alle Anforderungen und Bedürfnisse einer Garage erfüllen.

Durch die Entwicklung eines solchen Systems soll diese Digitalisierung des Schlüsselmanagements einen Mehrwert gegenüber bereits vorhanden Lösungen bieten. Diese Digitalisierung umfasst die Integration von RFID-Technologie, die die anzeigen der gewünschte Fahrzeug- und Fahrzeughalter Daten.

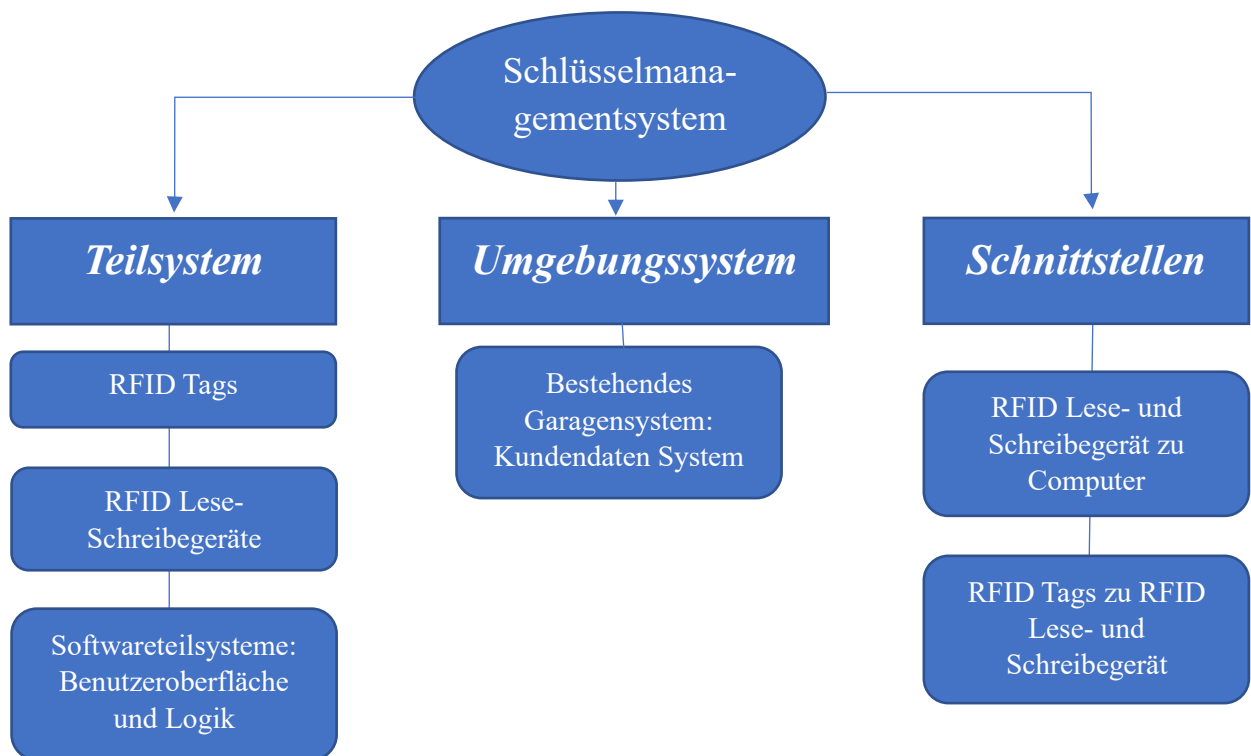
### 5.4.2 Richtziel

Das Richtziel besteht darin ein innovatives und funktionales digitales Schlüsselmanagement-System für Garagen zu entwickeln. Das System soll auf RFID-Technologie basieren. Das System soll eine benutzerfreundliche Oberfläche bieten, um den Mechanikern und andere Garagen Mitarbeiter die Arbeit mit dem System zu vereinfachen und eine intuitive Bedienung ermöglicht. Das System soll eine manuelle Schlüsselmanagement überwinden, eine sichere und zeitsparende Schlüsselmanagement zu ermöglichen. Das System soll eine erneute Benutzung des RFID tags ermöglichen. Durch eine erfolgreiche Umsetzung des Ziels soll die Diplomarbeit einen Beitrag zur Optimierung der Arbeitsabläufe in der Garage leisten. Nicht zu vergessen ist, dass die Arbeit hauptsächlich für die Garagen gezielt ist obwohl dass das System auch für andere Firmen, die mit viele Schlüssel zu tun haben, genutzt werden kann.

| <b>ID</b> | <b>Funktionale Anforderungen</b>  | <b>Zieloption<br/>( Kann / Muss )</b> |
|-----------|---|---------------------------------------|
| 1.        | Identifizierung des Schlüssels mittels RFID-Tags  | Muss                                  |
| 2.        | Installation von RFID-Lese- Schreibgeräten in einem Zentralen Schlüsselkasten oder neben Computer           | Muss                                  |
| 3.        | Erfassung und Speicherung der Schlüssel und Fahrzeugdaten in einer Datenbank                                | Kann                                  |
| 4.        | Anzeige von Fahrzeug- und Fahrzeughalterdaten beim Scannen des RFID-Tags                                    | Muss                                  |
| 5.        | Einfache Zuordnung der Schlüssel zu den entsprechenden Fahrzeugen   | Kann                                  |
| 6.        | Leuchten des RFID-Tags beim Betätigen auf der Benutzer Oberfläche   | Kann                                  |
| 7.        | Benutzen der Fundoino MEGA2560  | Kann                                  |
| 8.        | Benutzen der Raspberry PI 3   | Kann                                  |
| 9.        | GUI-Bibliotheken verwenden, um die Benutzeroberfläche zu erstellen  | Kann                                  |
| 10.       | Als Ausgangspunkt die bereits vorhandene Bibliotheken verwenden, die eine Interaktion mit RFID-Tags bieten. | Kann                                  |
| 11.       | Durch die Benutzeroberfläche Aktivitäten protokollieren   | Kann                                  |
|           |   |                                       |
|           | <b>Nicht funktionale Anforderungen</b>  |                                       |
| 12.       | Benutzerfreundliche und intuitive Benutzeroberfläche für den Bediener                                       | Kann                                  |
| 13.       | Sicherheit der gespeicherten Daten durch Zugriffskontrollen und Verschlüsselung                             | Kann                                  |
| 14.       | Möglichkeit bis 100 Fzg. Zu registrieren  | Kann                                  |

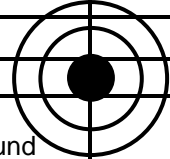
|     |  |      |
|-----|--|------|
| 15. | Reaktionszeit ist auf dem höchsten Niveau die die RFID-Technologie bietet  | Muss |
| 16. | <b>Technische Anforderungen</b>  |      |
| 17. | Kompatibilität mit gängigen Betriebssystemen und Hardwareplattformen   | Kann |
| 18. | Integration von RFID-Lese- und Schreibegeräten mit der Softwareanwendung über eine geeignete Schnittstelle (z.B. USB, WLAN...) | Muss |
| 19. | RFID-Technologieauswahl auf UHF-Niveau bis 30 m Reichweite   | Kann |
| 20. | Das System hat die integrierbarkeit nahtlos in IT-Infrastrukturen der Garage   | Kann |
| 21. | RFID Lese- und Schreibegerät ermöglicht das Schreiben und Lesen der Daten  | Muss |

Tabelle 1: Funktionale und nicht funktionale Anforderungen



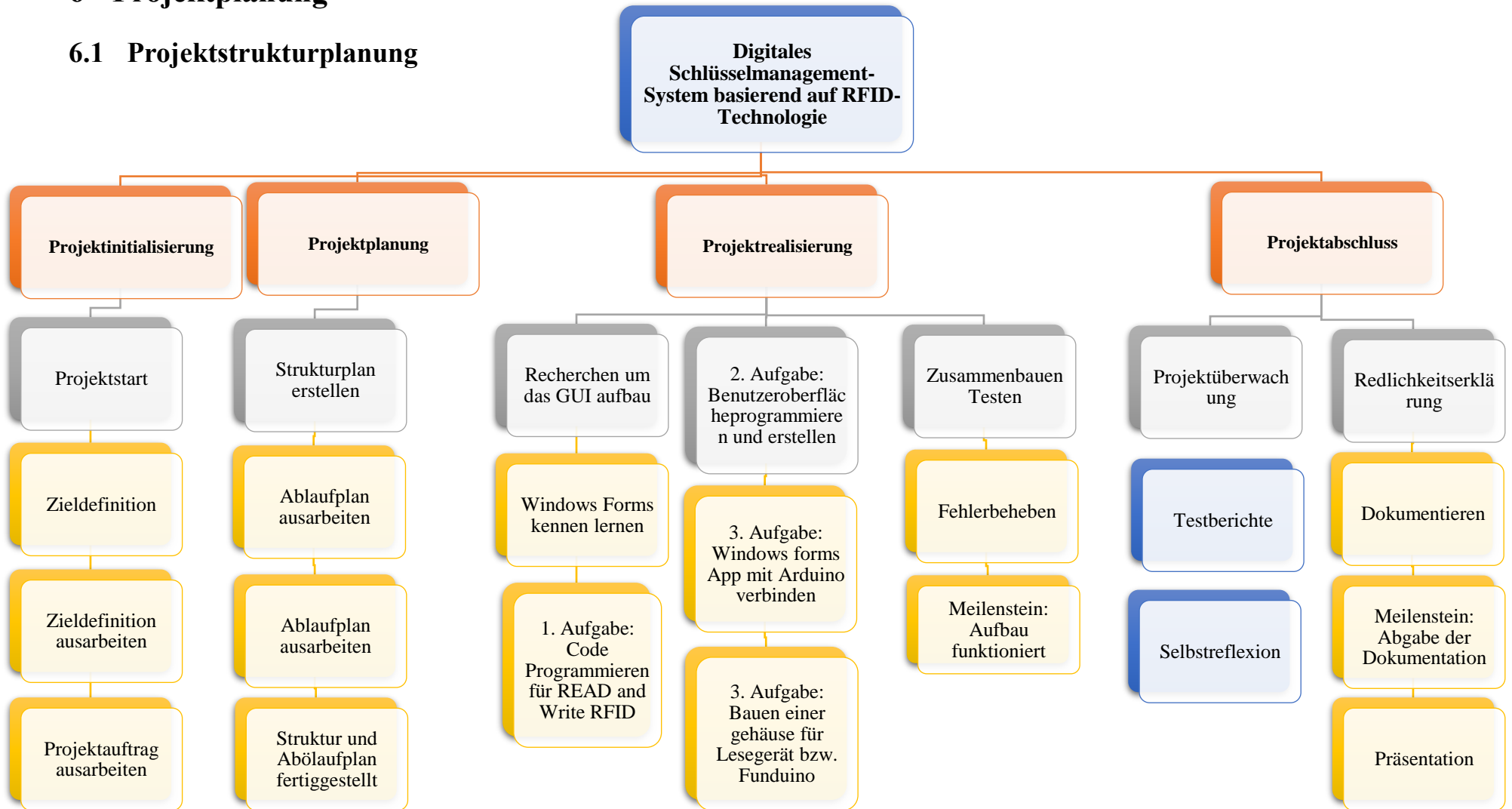
## 5.5 Zielscheibe

|  |   |
|--|---|
| Richtziel: Ein funktionales digitales Schlüsselmanagement-System basierend auf RFID-Tech. ist entwickelt, gebaut, getestet und dokumentiert.   |   |
| <ol style="list-style-type: none"> <li>1. Eine vollständige Projektdokumentation liegt in Digitaler Form mit folgenden Inhalten vor:             <ol style="list-style-type: none"> <li>1.1. Ein Ablaufplan und ein Projektstrukturplan sind erstellt</li> <li>1.2. Eine Benutzeroberfläche als App für eine Desktop-Anwendung ist erstellt</li> <li>1.3. Es ist eine grafische Lösung anhand der Bibliothek der Arduino-Plattform für RFID Write and Read erstellt</li> <li>1.4. Beim Lesen der Schlüsselinformationen sind die Daten auf der Benutzeroberfläche sichtbar.</li> <li>1.5. Das Schreiben der Kunden Daten auf RFID-Tags ist für den Benutzer möglich und wird durch eine Bedienungsanleitung vereinfacht.</li> <li>1.6. Das erneute verwenden der RFID-Tags ist möglich.</li> <li>1.7. Ein Testbericht liegt vor.</li> <li>1.8. Eine Selbstreflexion wurde verfasst, nachdem die Dokumentation erstellt wurde.</li> </ol> </li> </ol> | <ul style="list-style-type: none"> <li>- Kelmendi Egzon</li> </ul>  |
| <b>Endergebnisse</b>   | <b>Kunde</b>  |
| <b>Sinn und Zweck</b>  | <b>Erfolgskriterien</b>   |
| <ul style="list-style-type: none"> <li>- Lernen eigenständig und zielgerichtet an einem Projekt zu arbeiten und mit den erforderlichen Ressourcen umzugehen und Ergebnisse fristgerecht zu liefern.</li> <li>- Das erlernte Wissen und die Fähigkeiten in den Bereichen Projektmanagement und Dokumentation zu erweitern und die Kreativität in der Praxis anwenden.</li> </ul>  | <ol style="list-style-type: none"> <li>1. Mind. 85% der Aufgabe ist so gelöst, dass keine rückfragen auftreten.</li> <li>2. Die Benutzeroberfläche als App ist mit Windows Forms erstellt, und es erscheint ein App-Icon auf dem Desktop.</li> <li>3. Die Arduino-Bibliotheken wurden verwendet, der Code wurde erweitert und an die Projektziele angepasst.</li> <li>4. Programm code ist auf den Arduino geladen, und durch die Verbindung mit Computer werden die Schlüsselinformationen in der Windows Forms-App angezeigt.</li> <li>5. Im Falle von Schwierigkeiten bei der ersten Verwendung der Windows App steht eine Bedienungsanleitung zur Verfügung.</li> <li>6. Die gespeicherten Informationen können neu geschrieben werden, wodurch das RFID-Tag erneut verwendet werden kann.</li> <li>7. Folgende Arten von Tests werden durchgeführt: Smoke Tests, Integrationstests, Abnahmetests.</li> <li>8. Es wird aufgezeigt, wo die Schwierigkeiten aufgetreten sind und was noch geübt werden muss.</li> </ol> |



## 6 Projektplanung

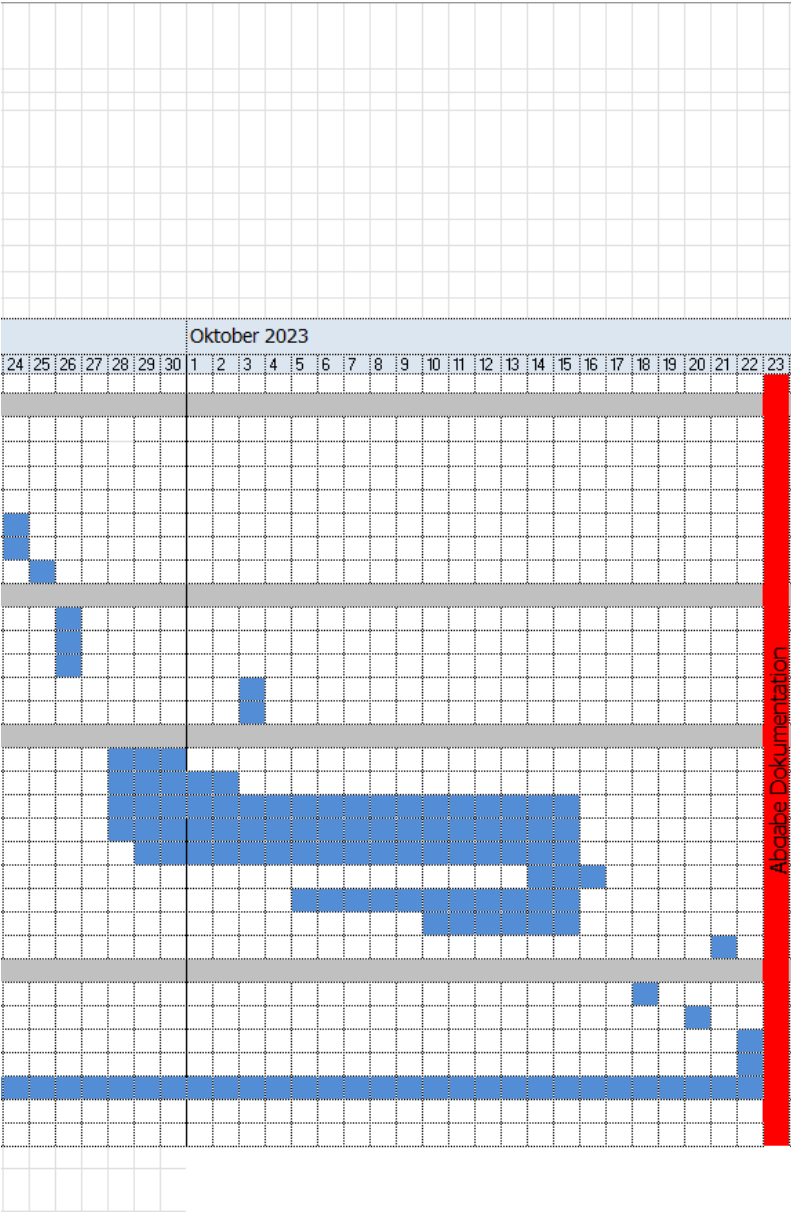
### 6.1 Projektstrukturplanung



## 6.2 Projektablaufplan/ Projektstrukturplan

| Projektplan - Digitales Schlüsselmanagement-System basierend auf RFID-Technologie |            |            |            |             |            |        |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|------------|------------|------------|-------------|------------|--------|-----------|---------------|----------------|--------|--------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Projektstart  | 11.09.2023 | Datum      | 11.09.2023 | Projektende | 23.10.2023 | Versio | 1.10      | Projektleiter | Kelmendi Egzon | Klasse | O_TEL_20_S_a |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Firma   | TEKO Olten |            |            |             |            |        |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Vorgang   | Start      | Ende       | Sachmitte  | Kosten      | Prozent    | Status | Juni 2023 |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | September 2023 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <b>Phase Initialisierung</b>  |            |            |            |             |            |        |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Themen Auswahl Abgabe   | 21.06.2023 | 21.06.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Projektstart  | 11.09.2023 | 11.09.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Zieldefinition  | 19.09.2023 | 19.09.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Zwischenbesprechung 1   | 19.09.2023 | 19.09.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Zieldefinition ausarbeiten  | 24.09.2023 | 24.09.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Projektauftrag ausarbeiten  | 24.09.2023 | 24.09.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Meilenstein Projektauftrag genehmigt  | 25.09.2023 | 25.09.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <b>Phase Planung</b>  |            |            |            |             |            |        |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Strukturplan erstellen  | 26.09.2023 | 26.09.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ablaufplan ausarbeiten  | 26.09.2023 | 26.09.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Struktur und Ablaufplan fertiggestellt  | 26.09.2023 | 26.09.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Zwischenbesprechung 2   | 03.10.2023 | 03.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Meilenstein Abschluss Planung Phase   | 03.10.2023 | 03.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <b>Phase Realisierung</b>   |            |            |            |             |            |        |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Recherchen um das GUI aufbau  | 28.09.2023 | 30.09.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Windows Forms kennen lernen   | 28.09.2023 | 02.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1. Aufgabe: Code Programmieren für READ and Write RFID                            | 28.09.2023 | 15.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2. Aufgabe: Benutzeroberfläche programmieren und erstellen                        | 28.09.2023 | 15.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3. Aufgabe: Windows forms App mit Arduino verbinden                               | 28.09.2023 | 15.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4. Aufgabe: Bauen einer gehäuse für Lesegerät bzw. Funduino                       | 14.10.2023 | 16.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Zusammenbauen, Testen   | 05.10.2023 | 15.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Fehler beheben  | 10.10.2023 | 15.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Meilenstein: Aufbau funktioniert  | 21.10.2023 | 21.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <b>Projektabschluss</b>   |            |            |            |             |            |        |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Projektüberwachung  | 18.10.2023 | 18.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testberichte  | 20.10.2023 | 20.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Selbstreflexion   | 22.10.2023 | 22.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Redlichkeitserklärung   | 22.10.2023 | 22.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Dokumentieren   | 11.09.2023 | 22.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Meilenstein: Abgabe der Dokumentation   | 22.10.2023 | 22.10.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Präsentation  | 10.11.2023 | 10.11.2023 |            |             | 100%       | done   |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <b>Gesamtaufwand in Personentage</b>  | <b>6 W</b> |            |            |             |            |        |           |               |                |        |              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Abbildung 2: Projektablaufplan



### 6.3 Auswahl des Programms für die Erstellung der GUI

Nachdem ich ausführlich recherchiert und verschiedene Optionen in Betracht gezogen hatte, entschied ich mich für die Verwendung von Windows Forms anstelle von Tkinter für die Entwicklung meiner GUI-Anwendung. Diese Wahl gründete sich auf mehreren Überlegungen und praktischen Erfahrungen, die ich während meiner Entwicklungsarbeit gemacht habe.

Zu Beginn meiner Recherche stieß ich auf eine Fülle von Ressourcen und Video-Tutorials, die sich auf Windows Forms konzentrierten. Dies ermöglichte es mir, schnell und effizient in die Welt der GUI-Programmierung einzusteigen. Die Möglichkeit, GUI-Elemente per Drag-and-Drop zu platzieren und zu gestalten, erwies sich als äußerst hilfreich, besonders für jemanden, der zuvor keine Erfahrung mit der Erstellung von Benutzeroberflächen hatte.

Die Verfügbarkeit von zahlreichen Online-Tutorials und Ressourcen, die speziell auf Windows Forms ausgerichtet sind, erleichterte meinen Lernprozess erheblich. Diese Tutorials halfen mir, die grundlegenden Konzepte der GUI-Programmierung in Windows Forms zu verstehen und mich schnell in die Entwicklung einzuarbeiten.

Ein weiterer Faktor, der meine Entscheidung beeinflusste, war die Tatsache, dass Windows Forms die Erstellung von Desktop-Anwendungen unterstützt. Da meine Anwendung für den Desktop bestimmt war, erwies sich Windows Forms als die natürlichere Wahl. Die Möglichkeit, eine Anwendung direkt auf dem Desktop auszuführen und darauf zuzugreifen, entsprach den Anforderungen meines Projekts. In den frühen Stadien meiner Arbeit verbrachte ich viel Zeit damit, sowohl zu programmieren als auch Online-Ressourcen zu konsultieren, um die erforderlichen Fähigkeiten zu erlernen. Die Lernkurve war zwar steil, aber durch die reichhaltigen Ressourcen und die intuitive Natur von Windows Forms konnte ich mich schnell verbessern und effizienter arbeiten.

Alles in allem war meine Wahl für Windows Forms eine Folge meiner persönlichen Präferenzen und der spezifischen Anforderungen meines Projekts. Die Möglichkeit, eine benutzerfreundliche Benutzeroberfläche zu erstellen und auf eine Vielzahl von Ressourcen zuzugreifen, machte Windows Forms zur idealen Wahl für meine GUI-Anwendung.

## 7 Projektrealisierung

### 7.1 Beschaffung von Hardware

#### 7.1.1 Materialliste und Infos

Das Material für den Projekt kann man auch online bestellen auf die Webseite der Funduino:

[Fachhändler für Mikroelektronik und 3D-Druck | Funduinoshop](#)

#### RFID TAG - Schlüsselanhänger, 13,56MHz

- RFID TAG im Format "Schlüsselanhänger"
- Maße: 40x32x4,3mm
- Frequenz: 13,56MHz
- Lesedistanz: 2,5-10cm



*Abbildung 3: RFID TAG - Schlüsselanhänger,  
13,56MHz*

#### RFID-KIT mit MIFARE RC522 Empfänger

- RFID-Set für Arduino. 90° Stiftleiste bereits angelötet
- Chip MFRC522
- ISO, IEC 14443A, MIFARE
- Protokoll: SPI
- Betriebsspannung: 3.3V
- Frequenz: 13,56 MHz
- Stromaufnahme: 13-26mA
- Stromaufnahme Standby: 10-13mA

- Betriebstemperatur: -20 bis 80°C



Abbildung 4: Mifare RC522 Empfänger

### Funduino MEGA 2560 R3 Mikrocontroller

- der Funduino MEGA2560 R3 ist mit der Arduino IDE vollständig kompatibel
- enthält den originalen ATMEGA16U2 Chip
- mit beschrifteten Pins
- mit 54 digitalen I/O Pins (davon 15 mit PWM-Ausgang)
- mit 16 analogen Eingängen
- Eingangsspannung: 7-12V

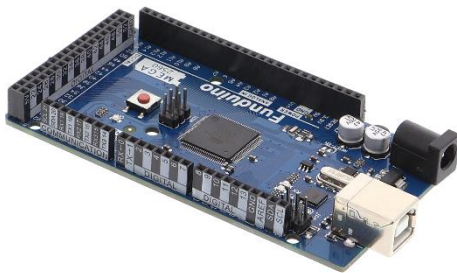


Abbildung 5: Funduino MEGA 2560 R3  
Mikrocontroller

**USB-A Kabel auf USB-B**

- USB2.0 Kabel für Arduino Mikrocontroller
- Länge: Auswählbar
- Farbe: Schwarz



*Abbildung 6: USB-A Kabel auf USB-B*

**Batteriefach - 6x AA (9V), mit DC-Stecker**

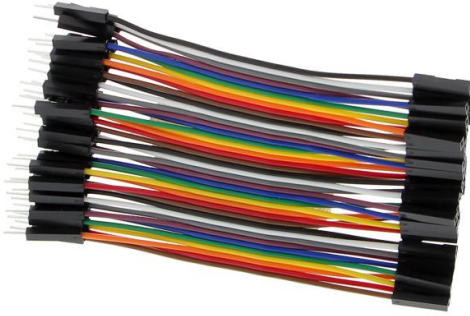
- Batteriefach für 6x AA-Batterien (9V)
- Kabellänge: Auswählbar



*Abbildung 7: Batteriefach - 6x AA (9V), mit DC-Stecker*

**Breadboardkabel männlich/weiblich**

- Kabellänge: Freiwillig wählbar
- Breadboardkabel: männlich/weiblich und männlich/männlich
- Farben: braun, rot orange, gelb, grün, blau, lila, grau, weiß. schwarz



*Abbildung 8: Breadboardkabel*

### 7.1.1.1 Hardware-Aufbau

Wie oben in der Ideenfindung bereits erwähnt, habe ich mich entschieden die Basis meines Projektes mit Arduino zu bauen. Als erstes selbstverständlich musste ich mich wieder bekannt mit Microcontroller machen. Ich baute als erstes die Aufgabe auf dem Arduino Buch auf. Nach der Installation der Software und Aufbau der Hardware wird es programmiert. Selbstverständlich das Benutzen der RFID-Tags ist keine Rocket-Science, aber man sollte schon ein paar Details beachten, und die werden unten in Kapitel Programmierung Vorgang erklärt.

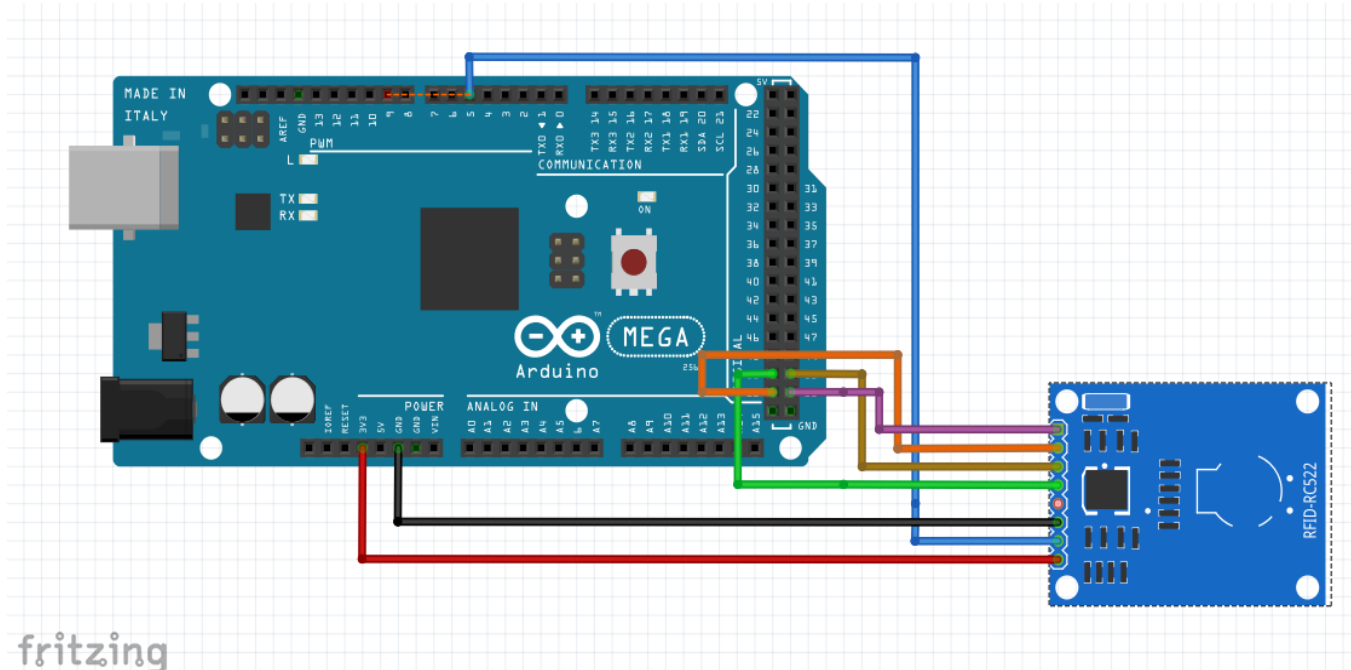


Abbildung 9: Simulation mit «Fritzing»

### 7.1.1.2 Pin-Layouts

Die Pin-Belegung sollte man sehr Große Achtung geben, da wenn ein falsches Einstecken würde das Projekt unbrauchbar machen. Da jedes Arduino ein spezielles PIN-Belegung hat musste diese an das Arduino angepasst werden. Die benannte "Signal" spalte bedeutet, dass die Befehle durch diese PINS von Computer mittels den Seriellen Monitor an das Arduino MEGA gesendet werden. Für die Kommunikation und Datenaustausch wird in andere Kapitel in Details gesprochen.

| Signal    | MFRC522<br>Reader/PCD<br>Pin-Belegung | Arduino<br>MEGA<br>Pin-Belegung |
|-----------|---------------------------------------|---------------------------------|
| RST/Reset | RST                                   | 5                               |
| SPI SS    | SDA (SS)                              | 53                              |
| SPI MOSI  | MOSI                                  | 51                              |
| SPI MISO  | MISO                                  | 50                              |
| SPI SCK   | SCK                                   | 52                              |
| GND       | GND                                   | GND                             |
| 3.3V      | 3.3V                                  | 3.3V                            |
| Unbelegt  | IRQ                                   | Unbelegt                        |

Tabelle 2: PIN-Belegung

## 7.2 Beschaffung von Software

Da dieses Projekt mit Arduino aufgebaut wird, muss es auch mit einem Programm gesteuert werden, und dazu müssen Programmiercodes verfasst werden. Arduino verfügt über eine eigene Software, die Open-Source ist. Sie kann kostenlos von <https://www.arduino.cc/en/software> heruntergeladen werden. Mithilfe dieser Arduino-Software können Programme geschrieben und an den Mikrocontroller gesendet werden, der sie dann ausführt.

Nach dem Herunterladen und Installieren der Software muss man am Anfang dies nun zuerst Einstellen. Es müssen 2 wichtige Einstellungen beachtet werden und nicht nur am Anfang, sondern jedes Mal, wenn man das Programm durchführt.

1: Beim Öffnen des Arduino Software sollte man zuerst das "Board" auswählen. Oben auf die leiste "Werkzeuge" auswählen dann " Board:" danach Untermenü "Boardverwalter" und schlussendlich das Board mit den man Arbeitet, in diesen Fall Funduino Mega2560 Board entsprechend als "Arduino MEGA2560".

2: Die richtige "Serielle Schnittstelle" auswählen. Genau unter dem Untermenü "Board" befindet sich die Auswahl des Ports. Unter "Port" können Sie die Serielle Schnittstelle auswählen. Achtung: Dieser Port muss immer derselbe sein und sollte stets gemerkt bzw. regelmäßig kontrolliert werden, da er mit Windows Forms übereinstimmen muss. In diesem Fall wird der Port "COM 3" ausgewählt.

### 7.2.1 Programmierung Vorgang

Nach der Installation der Software und dem Aufbau der Hardware wird sie programmiert. Bevor mit dem Programmieren begonnen wird, sollte man verstehen, wie das RFID-System funktioniert, wie die Daten gespeichert werden und später gelesen werden können. Da dies ein umfangreiches Thema ist und man sich in Details verlieren könnte, folgt die Erklärung nur den Punkten, die wichtig sind, sich auf dieses Projekt beziehen und das Programmieren erleichtern.

Um dies zu verstehen, wird zuerst ein "Bibliothek" installiert. Da wenn man die Software installiert, bereits ein paar Bibliotheken installiert sind, muss man für bestimmte Sachen weiter Bibliotheken installieren. Oben auf die leiste im Menü "Werkzeuge" auswählen und im Untermenü "Bibliotheken Verwalten" wird nachfolgende Bibliothek gesucht: "MFRC52 by GithubCommunity". Es wird immer das letzte Version installiert. Nach der Installation, wieder oben auf die leiste wird das Menü "Datei" aufgerufen und auf das Untermenü "Beispiele" die Bibliothek von MFRC522 als "DumpInfo" beschrieben ausgewählt. Nach Auswahl erscheint das Code, um den Aufbau der Speicherung der Daten auf RFID-Tag zu zeigen. Was bedeutet dies? Ein "Radio-Frequency Identification" oder Kurz RFID werden die Tags oder Karten genannt die Identifikationsinformationen speichern. Diese sind passiv, was bedeutet das sie keine eigenen Stromquellen haben. Sie werden durch Funkwellen, die von einem RFID-Lesegerät ausgestrahlt werden, aktiviert und dann die gespeicherten Infos zurück an das Lesegerät schicken.

**Achtung: Das Leseabstand zwischen RFID-Tag und Lesegerät (MFRC255) sollte Min. 2.5cm und Max. 10 cm sein, ansonsten wird die Kommunikation gestört oder nicht ermöglicht.**

Diese Bibliothek auf das Funduino Board laden und den seriellen Monitor aufmachen auch möglich mit Tasten Kombinationen (*Strg+Umschalt+M*). Es öffnet ein separates Fenster.

Das RFID-Tag hat keine bestimmte Kommunikation Seiten, das heißt man könnte auf Beide Seiten verwenden. Wenn das RFID-Tag (siehe Abbildung 3) auf das Lesegerät gehalten wird, werden auf das Fenster der Serielle Monitor wie unten auf dem Bild folgende Infos gezeigt. In diesem Fall sollte man RFID für ungefähr 10 Sek. auf das Lesegerät stehen lassen da dies ein bisschen länger braucht, um daten zu lesen als sonst, da hiermit wird jeder Block gelesen.

```

11:18:13.754 -> Card UID: B7 8D 40 B3 1
11:18:13.754 -> Card SAK: 08
11:18:13.801 -> PICC type: MIFARE 1KB 2
11:18:13.801 -> 3 Sector 5 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 8 AccessBits
11:18:13.895 -> 15 63 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:13.988 -> 4 62 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:14.036 -> 61 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:14.129 -> 6 60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:14.223 -> 14 59 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF 7 [ 0 0 1 ]
11:18:14.270 -> 58 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:14.364 -> 57 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:14.457 -> 56 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:14.551 -> 13 55 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:14.598 -> 54 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:14.691 -> 53 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:14.785 -> 52 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:14.879 -> 12 51 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:14.926 -> 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:15.020 -> 49 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:15.113 -> 48 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:15.207 -> 11 47 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:15.254 -> 46 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:15.348 -> 45 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:15.441 -> 44 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:15.535 -> 10 43 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:15.582 -> 42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:15.676 -> 41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:15.785 -> 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:15.817 -> 9 39 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:15.910 -> 38 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:16.004 -> 37 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:16.098 -> 36 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:16.145 -> 8 35 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:16.238 -> 34 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:16.332 -> 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:16.426 -> 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:16.472 -> 7 31 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:16.566 -> 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:16.660 -> 29 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:16.707 -> 28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:16.800 -> 6 27 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:16.894 -> 26 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:16.988 -> 25 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:17.035 -> 24 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:17.129 -> 5 23 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:17.224 -> 22 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:17.317 -> 21 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:17.364 -> 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:17.457 -> 4 19 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:17.552 -> 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:17.645 -> 17 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:17.692 -> 16 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:17.801 -> 3 15 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:17.880 -> 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:17.926 -> 13 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:18.020 -> 12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:18.114 -> 2 11 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:18.207 -> 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:18.254 -> 9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:18.348 -> 8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:18.443 -> 1 7 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:18.536 -> 6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:18.583 -> 5 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
11:18:18.677 -> 4 0A 66 72 32 39 35 36 37 34 20 20 20 20 20 20 20 20 [ 0 0 0 ]
11:18:18.770 -> 0 3 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
11:18:18.818 -> 2 0A 65 67 7A 6F 6E 20 20 20 20 20 20 20 20 20 20 20 [ 0 0 0 ]
11:18:18.912 -> 1 6B 65 6C 6D 65 6E 64 69 20 20 20 20 20 20 20 20 20 [ 0 0 0 ]
11:18:19.006 -> 0 B7 8D 40 B3 C9 08 04 00 62 63 64 65 66 67 68 69 [ 0 0 0 ]
11:18:19.100 ->
    
```

Abbildung 10: RFID Daten Speicherung Blocks „Dump Info“

### 7.2.1.1 Erklärung des "DumpInfo"

In Abbildung 9 sind alle Informationen und der Strukturaufbau des RFID-Tags zu sehen. Dies kann wie folgt erklärt werden:

**Nr. 1:** Wird das "Unique Identifier" kurz UID bekannt gegeben. Dies bezieht sich auf die eindeutige Identifikationsnummer, die einem RFID-Transponder zugeordnet ist. Diese wird bei der Herstellung des Tags festgelegt und kann nicht geändert werden. Dies ist auf HEX zahlen gespeichert und kann auch als DEC angezeigt werden, aber dafür sollte man den Programmiercode anpassen. Da es für dieses Projekt nicht relevant ist wird es so gelassen.

**Nr.2:** "*Proximity Integrated Circuit Card*" oder "*RFID-Karte*" oder "*PICC-Type*" ist die Karte/Tag oder der Chip die die Identifikationsinformationen speichert. In diesem Fall ist eine Karte mit MIFARE Technologie und ein Speicher Kapazität von 1 Kilobyte.

**Nr. 3, 4, 5, 6, 7:** Die Speicherung der Daten ist möglich in 16 Sektoren zu speichern, und jeder Sektor hat jeweils 3 Blocks, die beschrieben werden können. Das 4 Block ist nicht beschreibbar, auf den einzigen Grund der Sicherheitsdesign der MIFARE Technologie, also auf Sektor 1 ist Block 0 für Schlüsselinformationen und Block 3 für Sicherheitsbits und Zugriffsberechtigung. Die anderen Blocks können beschrieben werden und das muss auf HEX zahlen sein.

**Nr. 8:** Die *AccessBits*, auf Deutsch "Zugriffsbits" oder "Zugriffsberechtigungen," sind eine der Sicherheitsfunktionen, die in MIFARE Tags und -Karten verwendet werden. Sie dienen dazu, den Zugriff auf die auf dem Tag oder der Karte gespeicherten Daten zu regeln. Die *AccessBits* werden in den Datenblöcken eines MIFARE Classic-Tags gespeichert und bestimmen, wer Lese- und Schreibzugriff auf diese Daten hat.

## 7.2.2 Flussdiagramm der Software für Arduino

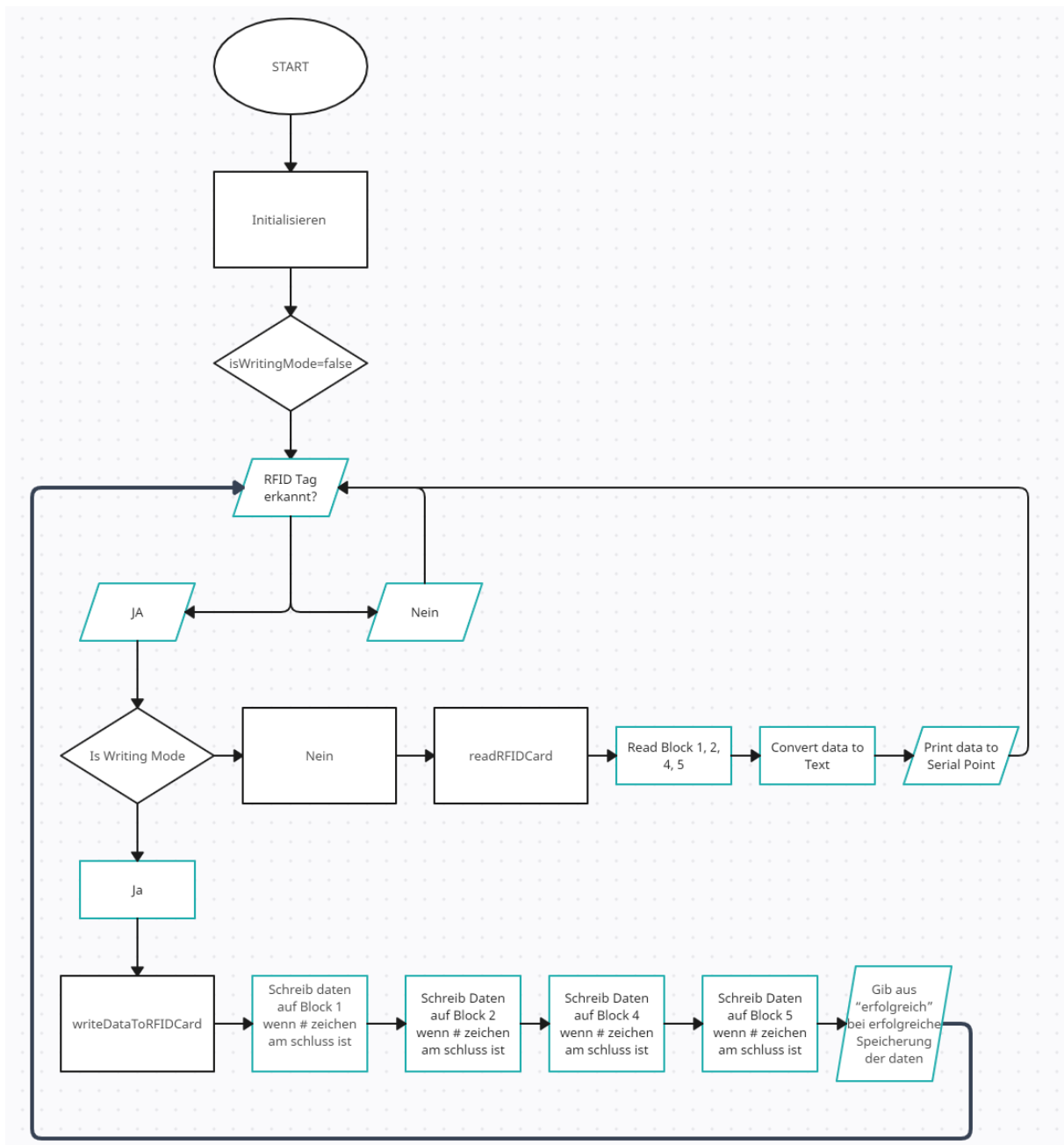


Abbildung 11: Flussdiagramm Software, Arduino

### 7.2.3 Code für das Schreiben der Daten auf RFID-Tag

Da man jetzt eine Ahnung davon hat, wie die Daten auf einem RFID-Tag gespeichert werden können, kann man anhand dessen den Code schreiben, um die Daten zu lesen und zu beschreiben. Ein neu geliefertes RFID-Tag ist nicht beschrieben, es scheint, dass jeder Block mit 00-Bits gefüllt ist. Beim Lesen des RFID-Tags erscheint nichts. Daher muss zuerst das RFID-Tag beschrieben werden. Da bereits eine freie Arduino-Bibliothek vorhanden ist, darf diese selbstverständlich als Basis verwendet werden. Allerdings erfüllt dieser Teil des Codes nicht das gewünschte Ziel, daher musste er angepasst und erweitert werden. Weitere Informationen zum Code und Details zum RFID-Tag/Karte oder RFID-Lesegerät können unter folgendem Link gefunden werden: <https://github.com/miguelbalboa/rfid.git>

```

1 #include <SPI.h> // Inkludieren der SPI-Bibliothek für die Kommunikation mit dem RFID-Modul
2 #include <MFRC522.h> // Inkludieren der MFRC522-Bibliothek für RFID-Kommunikation
3 #include <deprecated.h> // Inkludieren einer veralteten Bibliothek (deprecated) - möglicherweise nicht erforderlich
4 #include <MFRC522Extended.h> // Inkludieren einer erweiterten MFRC522-Bibliothek
5 #include <require_cpp11.h> // Inkludieren einer Bibliothek, die C++11 erfordert - möglicherweise nicht erforderlich
6
7 #define RST_PIN 8 // Definition des Reset-Pins für das RFID-Modul
8 #define SS_PIN 53 // Definition des Slave-Select-Pins für das RFID-Modul
9
10 MFRC522 mfc522(SS_PIN, RST_PIN); // Erstellung einer MFRC522-Instanz mit den Pins RST und SS
11 MFRC522::MIFARE_Key key; // Erstellung eines MIFARE_Key-Objekts für die RFID-Authentifizierung
12 MFRC522::StatusCode status; // Erstellung eines StatusCode-Objekts für die RFID-Kommunikation

```

Abbildung 12: Code für RFID Modul Kommunikation

Zunächst werden die notwendigen Bibliotheken und Header Dateien inkludiert, um die Kommunikation mit RFID-Modul zu ermöglichen. Dann werden 2 Pins definiert. Diese Pins sind an den Arduino angeschlossen und man sollte beachten, dass diese miteinander übereinstimmen. Oben auf Abbildung 9 ist die PIN-Belegung vorgeschrieben.

Nach dem sind 3 Objekte erstellt, die zum Kommunizieren und das Überwachung der Kommunikation dienen und die Authentifizierung durchführen.

```

13
14 bool isWritingMode = false; // Eine Flagge, um zwischen Schreib- und Lese-Modus zu wechseln
15

```

Abbildung 13: Erstellung eines Boolesche Variable

Diese Variable wird als "Flagge" oder "Schalter" verwendet, um zwischen zwei Betriebsmodi, nämlich dem "Schreibmodus" und dem "Lesemodus", umzuschalten. Beim Starten des Programms wird diese als falsch gesetzt damit mit lesen sofort begonnen kann, da dieses Projekt meistens am Anfang das Lesen fördert.

```

16 void setup() {
17   Serial.begin(9600); // Initialisierung der seriellen Kommunikation mit einer Baudrate von 9600
18   SPI.begin(); // Initialisierung der SPI-Schnittstelle
19   mfc522.PCD_Init(); // Initialisierung des RFID-Moduls
20 }

```

Abbildung 14: Initialisierung der Kommunikation und Schnittstelle

Hier wird die Schnittstelle zwischen Computer und RFID-Modul initialisiert und auch die Baudrate von 9600 zwischen beiden Geräten.

```
22 void loop() {
23   if (isWritingMode) {
24     // Code für das Schreiben von Daten auf RFID-Karten ausführen
25     writeToRFIDCard();
26   } else {
27     // Code für das Lesen von RFID-Karten ausführen
28     readRFIDCard();
29   }
30
31   if (Serial.available() > 0) { // Überprüfen, ob Daten von der seriellen Schnittstelle verfügbar sind
32     char command = Serial.read(); // Ein Zeichen von der seriellen Schnittstelle lesen
33     if (command == 'R') {
34       isWritingMode = false; // In den Lese-Modus wechseln
35     } else if (command == 'W') {
36       isWritingMode = true; // In den Schreib-Modus wechseln
37     }
38   }
39 }
```

Abbildung 15: Loop Code. Hauptprogramm

Dies teil ermöglicht die Steuerung des RFID-Moduls über die serielle Schnittstelle, wodurch Sie zwischen Lesen und Schreiben umschalten können, ohne den Arduino-Code neu hochladen zu müssen. Je nachdem Welche Modus ausgewählt wird. Die Modi werden durch die Benutzeroberfläche gesteuert für das Lesen oder Schreiben der Daten. Dieses wird später erklärt, wie es funktioniert, wenn das Code für die Windows Forms erklärt wird, da dieses teil abhängig von Windows Forms ist.

```

188 void writeToRFIDCard()
189 {
190 // Prepare key - all keys are set to FFFFFFFFh at chip delivery from the factory.
191 MFRC522::MIFARE_Key key;
192 for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
193
194 // Reset the loop if no new card present on the sensor/reader. This saves the entire process when idle.
195 if ( ! mfrc522.PICC_IsNewCardPresent() )
196 {
197     return;
198 }
199 // Select one of the cards
200 if ( ! mfrc522.PICC_ReadCardSerial() )
201 {
202     return;
203 }
204
205 Serial.print("Card UID:"); //Dump UID
206 for (byte i = 0; i < mfrc522.uid.size; i++) {
207     Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
208     Serial.print(mfrc522.uid.uidByte[i], DEC);
209 }
210
211 byte buffer[34];
212 byte block;
213 MFRC522::StatusCode status;
214 byte len;
215
216 //*****Block 1: Name*****//
217
218 Serial.setTimeout(10000L); // wait 10 seconds for input from serial
219 len = Serial.readBytesUntil('*', (char *) buffer, 30); // read family name from serial
220 for (byte i = len; i < 30; i++) buffer[i] = ' '; // pad with spaces
221
222 block = 1;
223
224 status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block, &key, &(mfrc522.uid));
225 if (status != MFRC522::STATUS_OK) {
226     Serial.print("PCD_Authenticate() failed: ");
227     Serial.println(mfrc522.GetStatusCodeName(status));
228     return;
229 }
230 else Serial.println(F("PCD_Authenticate() success: "));
231
232 // Write block
233 status = mfrc522.MIFARE_Write(block, buffer, 16);
234 if (status != MFRC522::STATUS_OK) {
235     Serial.print("MIFARE_Write() failed: ");
236     Serial.println(mfrc522.GetStatusCodeName(status));
237     return;
238 }
239 else Serial.println("MIFARE Write() success: ");

```

Abbildung 16: Code für das Schreiben der Daten auf Blocks

Dieser Code ist für das Schreiben der Daten auf das RFID-Tag. Dieses ermöglicht das Schreiben auf einen speziellen vorgewählten Block, in diesem Fall auf Das Block 1. Da das Ziel das Projekt ist das Speichern der Name, Vorname, Auto Kennzeichen und Typ des Autos (z.B. Renegade, Giulietta, Cherokee), wird auf das Erste Block nach Namen bzw. Name des Kunde gefragt und diese wird auf das Erste Block gespeichert, mit der Regel, dass am Schluss das Hashtag (\*) Zeichen ist, und dann auf Benutzeroberfläche "WRITE" Button gedrückt wird. Im Hintergrund der Benutzeroberfläche wird es so programmiert, dass die ASCII Zeichen auf HEX umgesetzt werden damit diese auf RFID-Tag gespeichert werden können. Auf die Zeile 222 wird das Block ausgewählt, in dem die Daten gespeichert werden. Im Falle einer Speicherung zeigt die Benutzeroberfläche, dass die daten gespeichert oder nicht gespeichert wurden. Dieser Code wird nun dort angepasst auf welche Blocks gespeichert will. Also das gleiche Code wird nochmal überkopiert und nur die Zahlen der Blocks neu geschrieben also Block 2, 4 und 5.

## 7.2.4 Code für das Lesen der Daten auf RFID-Tag

```

50 //*****"Read Part"*****
51 void readRFIDCard() {
52 // Die folgenden Zeilen dienen dazu, den Authentifizierungsschlüssel für die RFID-Karte vorzubereiten
53 for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
54
55 byte block;
56 byte len;
57 //-----
58
59 // Wenn keine neue Karte auf dem Sensor/Reader vorhanden ist,
60 // wird die Schleife zurückgesetzt, um den gesamten Prozess im Leerlauf zu sparen.
61 if (! mfrc522.PICC_IsNewCardPresent()) {
62     return;
63 }
64 // Eine Karte auswählen
65 if (! mfrc522.PICC_ReadCardSerial()) {
66     return;
67 }
68
69 // Hier werden Puffer für verschiedene RFID-Blöcke initialisiert
70 byte buffer1[16]; // Puffer für Block 1
71 byte buffer2[16]; // Puffer für Block 2
72 byte buffer4[16]; // Puffer für Block 4
73 byte buffer5[16]; // Puffer für Block 5
74
75 // Block 1 lesen
76 block = 1;
77 len = 16;
78
79 // Authentifizierung mit dem Schlüssel und UID der Karte
80 status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1, &key, &(mfrc522.uid));
81 if (status != MFRC522::STATUS_OK) {
82     Serial.print(F("Authentifizierung fehlgeschlagen: "));
83     Serial.println(mfrc522.GetStatusCodeName(status));
84     return;
85 }
86
87 // Block 1 lesen
88 status = mfrc522.MIFARE_Read(block, buffer1, &len);
89 if (status != MFRC522::STATUS_OK) {
90     Serial.print(F("Lesen fehlgeschlagen: "));
91     Serial.println(mfrc522.GetStatusCodeName(status));
92     return;
93 }
94
95 String value = "";
96 for (uint8_t i = 0; i < 16; i++)
97 {
98     value += (char)buffer1[i];
99 }
100

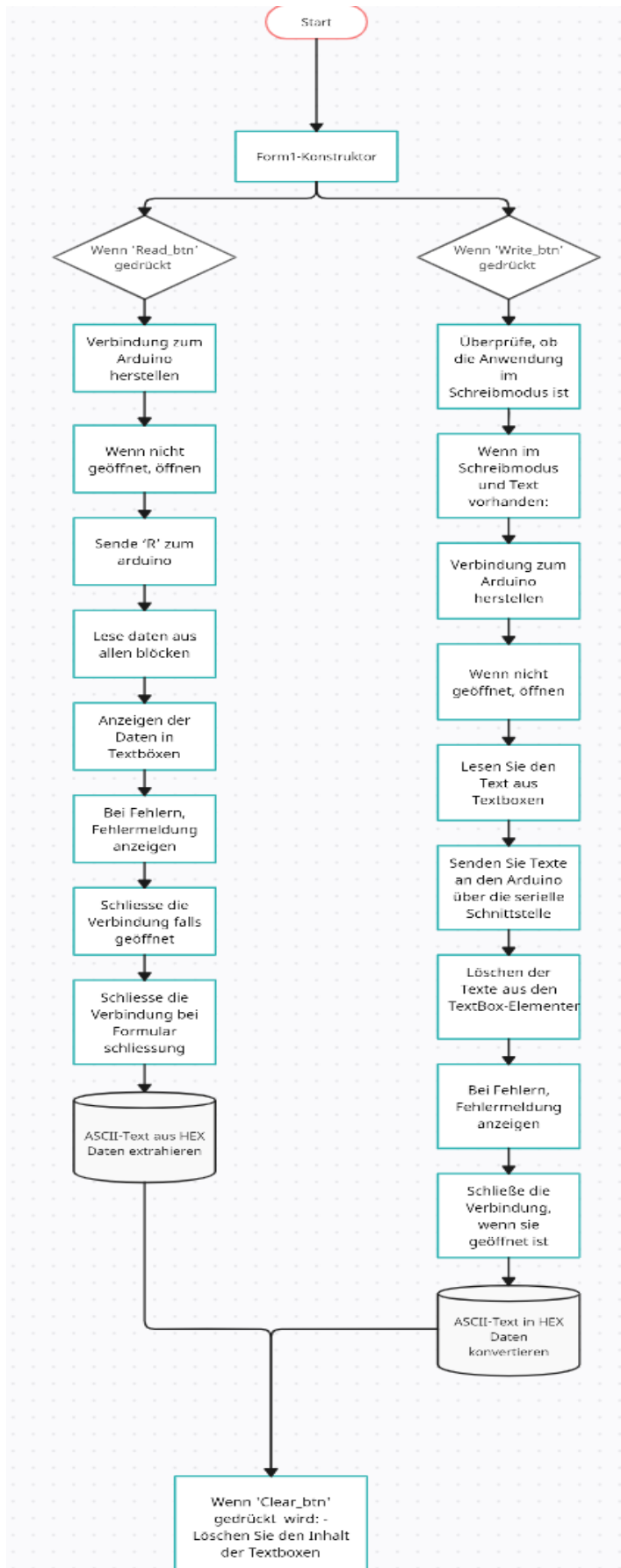
```

Abbildung 17: Code für das Lesen der Daten auf die Blocks

Dieser Code dient dazu, Daten von bestimmten Blocks, in diesem Fall Block 1 einer erkannten RFID-Karte zu lesen, nachdem die Karte erfolgreich authentifiziert wurde. Dieser Code wurde so angepasst und erweitert, um weitere Blöcke zu lesen. Da wird von Zeile 65 bis 89 das gleiche code kopiert und nur auf die Zeile 66 und 70 so angepasst an jeweiligen Block und neu unten hinzugefügt, um die Blocks zu lesen. Da es vorher auf die Blocks 1, 2, 4, 5 daten gespeichert wurden wird diesen Code auch nur so weit angepasst, dass nur von diesen Blocks gelesen wird.

## 7.3 Erstellung der Benutzeroberfläche

### 7.3.1 Flussdiagramm der Software für Benutzeroberfläche



### 7.3.2 Code für das Erstellen der Benutzeroberfläche

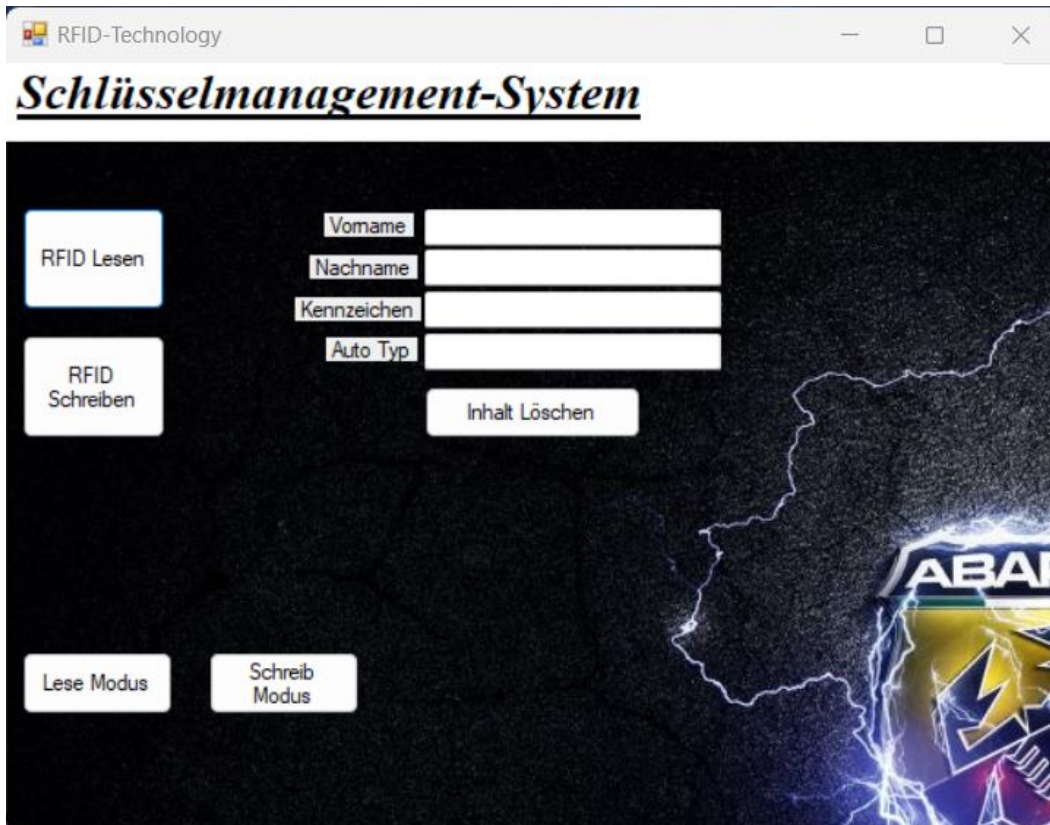


Abbildung 18: Benutzeroberfläche

Mit Windows-Forms wurde die Benutzeroberfläche erstellt. Da der gebrauch verlangt wie Lesen der Daten als auch Schreiben der Daten musste dies in 2 Programme geteilt. Das Lesen folgt sofort beim Legen der RFID-Tag auf das RFID-Reader. Wenn das Schreiben einer RFID-Tag gewünscht ist, dann muss die Option "Schreib Modus" ausgewählt. Nach dem Werden die daten auf die Text boxen geschrieben mit <\*> Zeichen am Schluss. Erst wenn ein <\*> Zeichen vorhanden ist kann auf der RFID-Tag geschrieben. Das RFID-Tag muss diese Zeit drauf bleiben. Und am Schluss nach dem Schreiben kann man auf dem Button "Schreiben" drücken und somit werden die Daten auf RFID-Tag geschrieben.

```
namespace Schlüsselmanagement_System
{
    3 Verweise
    public partial class Form1 : Form
    {
        private SerialPort Port;
        private bool isWritingMode = false;
        1 Verweis
        public Form1()
        {
            InitializeComponent();
            Port = new SerialPort("COM3", 9600);
        }
    }
}
```

Abbildung 19: Windows Forms Code für die Kommunikation zwischen App und Arduino

Dieses Teil der Code dient zur Kommunikation über einen seriellen Anschluss (COM 3) mit Arduino mit einer Geschwindigkeit von 9600 Baudrate.

```
26 private void Read_btn_Click(object sender, EventArgs e)
27 {
28     try
29     {
30         if (!Port.IsOpen)
31         {
32             Port.Open();
33         }
34
35         Port.WriteLine("R"); // Fordern Sie den Arduino auf, in den Lesemodus zu wechseln
36
37         // Lese die Daten aus allen Blöcken
38         string receivedDataBlock1 = Port.ReadLine(); // Block 1 lesen
39         string receivedDataBlock2 = Port.ReadLine(); // Block 2 lesen
40         string receivedDataBlock4 = Port.ReadLine(); // Block 4 lesen
41         string receivedDataBlock5 = Port.ReadLine(); // Block 5 lesen
42
43         // Anzeigen der Daten in den Textboxen
44         textBox1.Text = receivedDataBlock1; // Block 1 in textBox1 anzeigen
45         textBox2.Text = receivedDataBlock2; // Block 2 in textBox2 anzeigen
46         textBox3.Text = receivedDataBlock4; // Block 4 in textBox3 anzeigen
47         textBox4.Text = receivedDataBlock5; // Block 5 in textBox4 anzeigen
48     }
49     catch (Exception ex)
50     {
51         MessageBox.Show("Fehler: " + ex.Message);
52     }
53     finally
54     {
55         if (Port.IsOpen)
56         {
57             Port.Close();
58         }
59     }
60 }
61
62
63 }
```

Abbildung 20: Lese Knopf Erstellung

Dieser Teil der Code ist ein Event-Handler für das Lese Button in Windows Forms-Anwendung. Wenn der Button "Lesen" geklickt wird, wird der Code ausgeführt. Als erstes wird das Port geöffnet und dann in Port die Commando R geschickt damit der Arduino code den Bereich "Read data from the Blocks" ausführt. Nach dem die Daten gelesen und zurückgeschickt wurden werden diese zu jeweiligen Text Box zugestellt und dem entsprechend angezeigt. Falls ein Fehler in der Kommunikation Zeit vorkommt, wird ein Message Box angezeigt und der Fehler gemeldet.

```
1 Verweis
153 private void Write_btn_Click(object sender, EventArgs e)
154 {
155     // Überprüfen, ob die Anwendung sich im Schreibmodus befindet
156     if (isWritingMode)
157     {
158         // Überprüfen, ob Text vorhanden ist
159         if (!string.IsNullOrEmpty(textBox1.Text))
160         {
161             try
162             {
163                 if (!Port.IsOpen)
164                 {
165                     Port.Open();
166                 }
167
168                 // Lesen Sie den Text aus textBox1
169                 string textToWriteBlock1 = textBox1.Text;
170                 string textToWriteBlock2 = textBox2.Text;
171                 string textToWriteBlock4 = textBox3.Text;
172                 string textToWriteBlock5 = textBox4.Text;
173
174                 // Senden der Texte an den Arduino über die serielle Schnittstelle
175                 Port.WriteLine(textToWriteBlock1); // Schreiben Sie den Text auf Block 1
176                 Port.WriteLine(textToWriteBlock2); // Schreiben Sie den Text auf Block 2
177                 Port.WriteLine(textToWriteBlock4); // Schreiben Sie den Text auf Block 4
178                 Port.WriteLine(textToWriteBlock5); // Schreiben Sie den Text auf Block 5
179
180
181                 //Löschen der Texte aus den TextBox-Elementen, nachdem sie gesendet wurden
182                 textBox1.Clear();
183                 textBox2.Clear();
184                 textBox3.Clear();
185                 textBox4.Clear();
186             }
187             catch (Exception ex)
188             {
189                 MessageBox.Show("Fehler: " + ex.Message);

```

Abbildung 21: Write Button Code

Dieser Teil der Code ist ein Event-Handler für das Schreibe Button in Windows Forms-Anwendung. Wenn der Button "Schreiben" geklickt wird, wird der Code ausgeführt. Als erstes wird überprüft ob es in Schreib Modus ist, wenn die Anwendung sich im Schreibmodus befindet, wird überprüft, ob Text in den TextBox-Elementen "textBox1", "textBox2", "textBox3" und "textBox4" vorhanden ist. Wenn Text vorhanden ist, wird der Schreibvorgang fortgesetzt. Falls der Kommunikation Port nicht geöffnet ist, wird dies geöffnet und dann in die entsprechende variablen kopiert. Auch ein Button für das Löschen der Texte aus den Text Boxen ist erstellt. Falls ein Fehler in der Kommunikation Zeit vorkommt, wird ein Message Box angezeigt und der Fehler gemeldet.

```
222 | 1 Verweis  
223 | private void write_modus_btn_Click(object sender, EventArgs e)  
224 | {  
225 |     // Fordern Sie den Arduino auf, in den Schreibmodus zu wechseln  
226 |     if (!Port.IsOpen)  
227 |     {  
228 |         Port.Open();  
229 |     }  
230 |     Port.WriteLine("W");  
231 |  
232 |     // Aktivieren Sie den Schreibmodus  
233 |     isWritingMode = true;  
234 | }  
235 |  
236 | 1 Verweis  
237 | private void read_modus_btn_Click(object sender, EventArgs e)  
238 | {  
239 |     // Fordern Sie den Arduino auf, in den Lese-Modus zu wechseln  
240 |     if (!Port.IsOpen)  
241 |     {  
242 |         Port.Open();  
243 |     }  
244 |     Port.WriteLine("R");  
245 |     Port.Close();  
246 |  
247 |     // Deaktivieren Sie den Schreibmodus  
248 |     isWritingMode = false;  
249 | }
```

Abbildung 22: Auswahl Buttons für Lese oder schreib Modus

Dieses Teil der Code ermöglicht das Erstellen 2 Buttons. Eine ist dafür, dass Arduino in den Schreibmodus zu versetzen und das andere in Lesemodus zu versetzen. Es wird zuerst wie immer den Port geöffnet und dann in den Port der Befehl geschrieben. Je nachdem welcher Button geklickt wurde wird Arduino in Lese bzw. Schreibmodus gesetzt. Wenn der Befehl "R" geschickt wird, dann setzt der Arduino Code in Lesemodus und wenn der Befehl "W" geschickt wird, dann setzt der Arduino Code in Schreibmodus und man kann den RFID-Tag beschreiben.

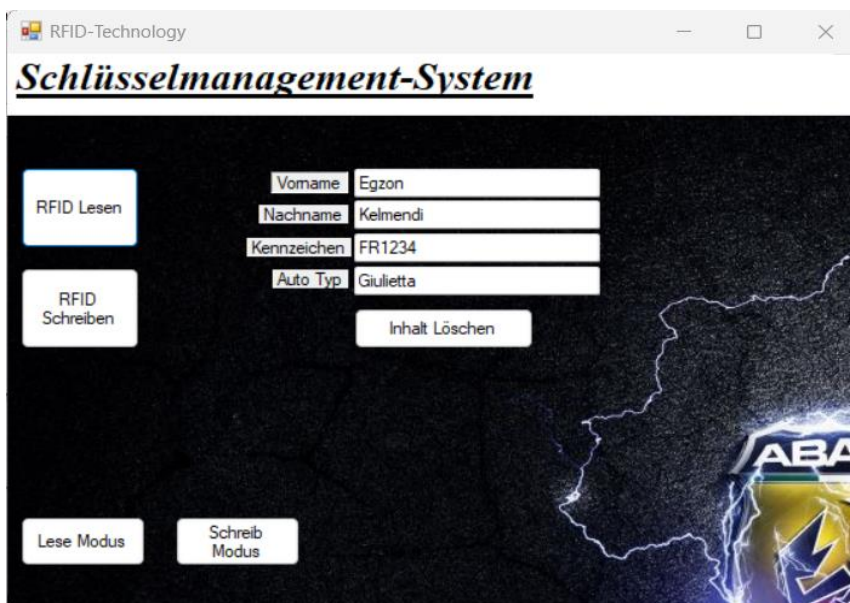


Abbildung 23: RFID-Tag Beispiel: das Lesen der Daten

Auf Bild 23 sieht man ein Beispiel wie nach lesen der Daten eine Darstellung aussieht. Und unten auf dem Bild 24 ist zu sehen, wie das Schreiben der Daten folgen sollte damit diese erfolgreich gespeichert werden.

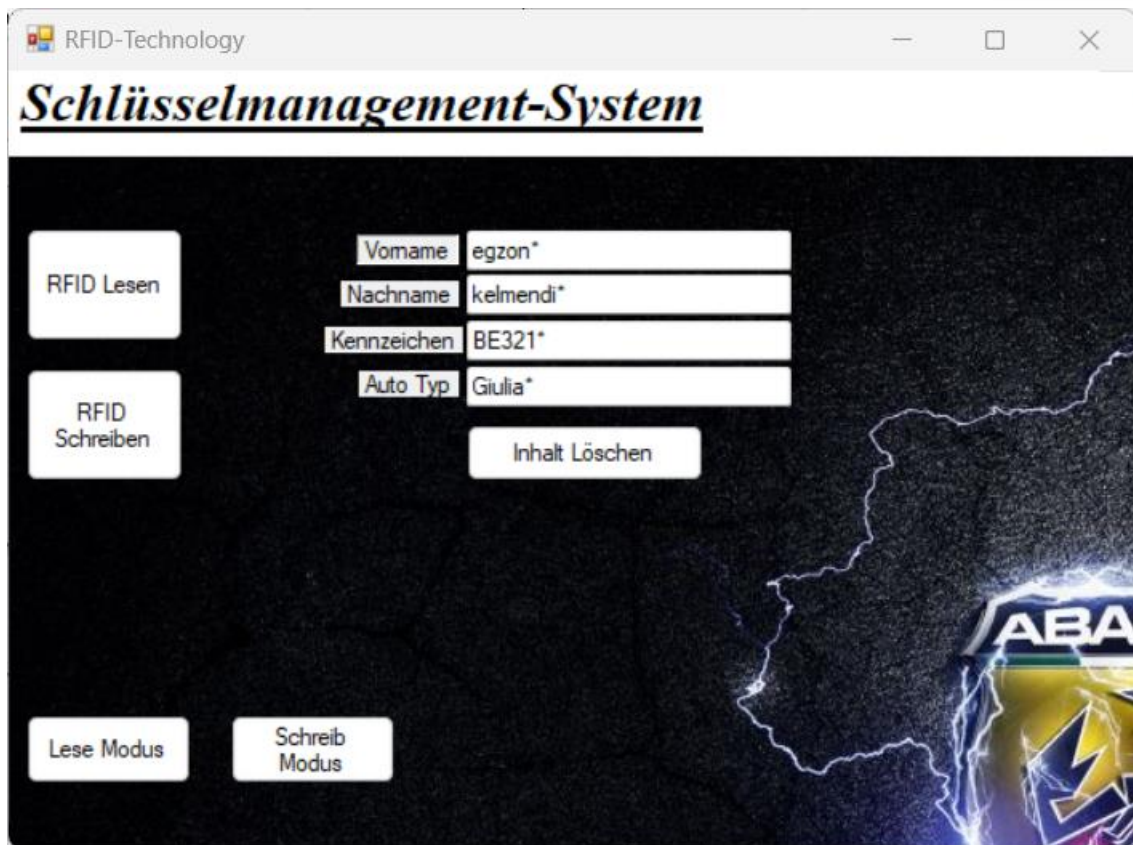


Abbildung 24: Beispiel: Schreiben der Daten auf RFID-Tag

Wie auch oben erwähnt muss zuerst auf das Knopf "Schreib Modus" gedrückt werden, die Daten schreiben und am Schluss auf den Knopf „RFID Schreiben“

## 7.4 Der Prototyp



Abbildung 25: Prototyp Aufbau

Der Prototyp wurde fertiggestellt, und seine Funktionsweise wurde erfolgreich überprüft. Es wurde für Holz anstelle von Stahl entschieden, da Stahl leitend ist und es die Gefahr eines Kurzschlusses in Verbindung mit den gelöteten Pins im hinteren Bereich der Elektronik erhöhen könnte. Der Aufbau dürfte auch nicht grösser sein da der Platzbedarf in der Garage klein ist.

### 7.4.1 Der Aufbau bei Garage Klaus

Der Aufbau folgte neben dem Computer der Werkstatt. Da dort am meisten die Schlüssel Verlust oder das Management sehr schlecht ist, wurde mit dem Mitarbeiter der Garage entschieden, dass dort ein idealer Platz wäre für dieses Prototyp. Nach dem Aufbau wurde es getestet und die Bedienung gezeigt.

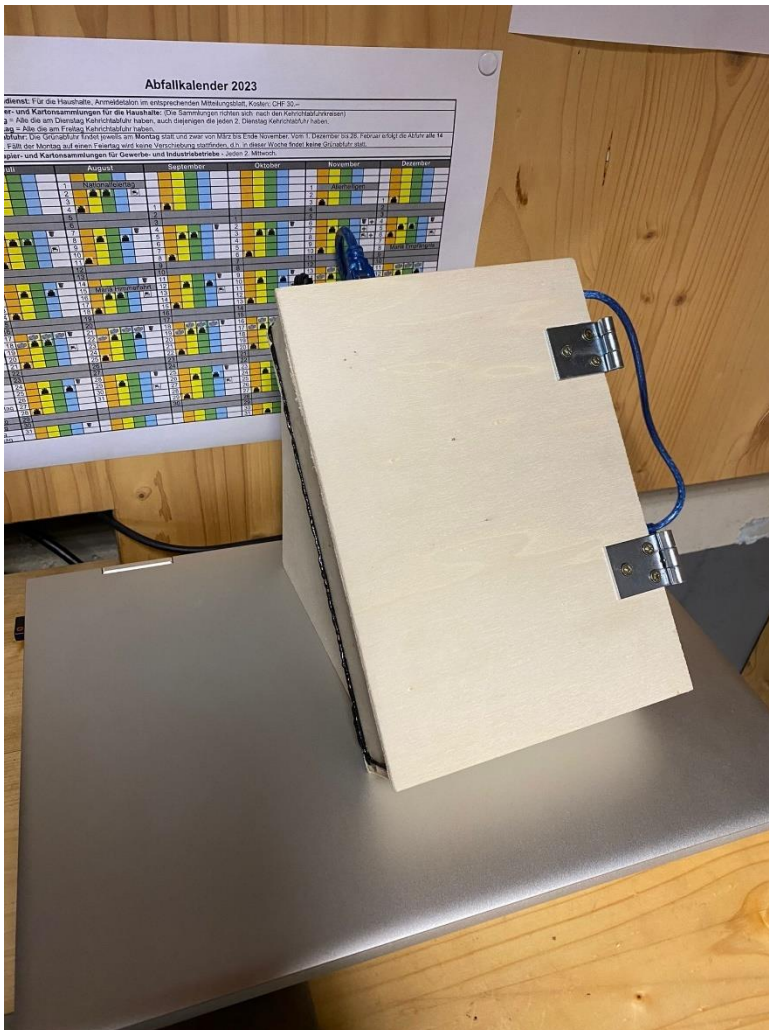


Abbildung 26: Aufbau bei der Garage Klaus

## 7.5 Inbetriebnahme-Protokoll

| Schlüsselmanagement-System basierend auf RFID-Technologie |  | Verantwortlicher:<br>Kelmendi Egzon |      |
|---|--|-------------------------------------|------|
|   |  | Ja                                  | Nein |
| <b>Hardware-Überprüfung</b>                               |  |                                     |      |
|   | RFID-Modul korrekt mit Arduino verbunden   | X                                   |      |
|   | RFID-Modul hat ausreichende Spannungsversorgung                                  | X                                   |      |
|   | Die Verbindungen sind fest und korrekt   | X                                   |      |
|   | Arduino an einen Computer angeschlossen  | X                                   |      |
| <b>Software-Überprüfung</b>                               |  |                                     |      |
|   | Windows App gestartet.   | X                                   |      |
|   | Windows App erkennt richtigen COM-Port.  | X                                   |      |
|   | Windows App gestartet, ohne Fehler oder Warnungen                                | X                                   |      |
|   | Die richtige Board-Einstellung ist ausgewählt                                    | X                                   |      |
|   | Das App zeigt keine Fehler beim Starten  |                                     |      |
| <b>Funktionsüberprüfung</b>                               |  |                                     |      |
|   | Arduino gestartet und RFID-Modul initialisiert                                   | X                                   |      |
|   | RFID-Modul ist in der Lage, RFID-Karten zu erkennen                              | X                                   |      |
|   | RFID-Modul kann Daten von erkannten Karten lesen                                 | X                                   |      |
|   | Der Wechsel zwischen Lese- und Schreibmodus funktioniert                         | X                                   |      |
|   | Die gelesenen Daten werden erfolgreich auf der seriellen Schnittstelle angezeigt | X                                   |      |
|   | Das Schreiben von Daten auf eine RFID-Karte funktioniert korrekt                 | X                                   |      |
| <b>Fazit</b>  |  |                                     |      |
|   | Das RFID-Modul und der Arduino-Code funktionieren wie erwartet                   | X                                   |      |
|   | Das Inbetriebnahme-Protokoll wurde erfolgreich abgeschlossen                     | X                                   |      |
|   | Es wurden keine schwerwiegenden Probleme oder Fehler festgestellt                |                                     | X    |

## 7.6 Ergebnisse der Inbetriebnahme

Nach der Fertigstellung des Prototyps und der Inbetriebnahme war es Zeit für den Test. Zuerst wurde die Bedienung erklärt, und dann konnten die Mitarbeiter sie ausprobieren. Nach dem Test wurden einige Details bemängelt. Ein Mitarbeiter bemerkte, dass beim Schreiben der Daten kein "#" Zeichen am Ende stehen sollte, da er nicht besonders versiert im Umgang mit Computern ist und es für ihn schwierig war, dies zu finden. Nach der Besprechung mit ihm wurde entschieden, am Ende ein "\*" Zeichen hinzuzufügen, um die Bedienung zu erleichtern. Schließlich ist das Hauptziel dieses Projekts eine einfache Bedienung des Prototyps zu ermöglichen.

## 8 Projektabschluss

### 8.1 Evaluation der Zielerreichung

Das Projekt wurde erfolgreich umgesetzt, und die Ziele wurden mit positiven Ergebnissen erreicht. Die Zeitplanung konnte größtenteils eingehalten werden, jedoch gab es Momente, in denen sie nicht eingehalten werden konnte. Beim Programmieren stieß ich auf viele Probleme, die zusätzliche Zeit in Anspruch nahmen. Das Programmieren konnte erst gegen Ende der 5. Semesterarbeitswoche abgeschlossen werden. Da ich jedoch bereits früher mit der Hardware begonnen hatte, konnte ich den Abschluss des Hardwareaufbaus erreichen. Eine grafische Darstellung wurde erfolgreich erstellt, nachdem ich viele Stunden damit verbracht hatte, Informationen dazu zu sammeln. Diese funktioniert einwandfrei mit dem Arduino-Code und dem Funduino-Aufbau. Eine Bedienungsanleitung wurde nicht erstellt, da die Benutzeroberfläche sehr intuitiv war und nur einmal gezeigt werden musste.

Es gäbe sicherlich noch einige Verbesserungsmöglichkeiten, wie das Hinzufügen weiterer Hardwarekomponenten. Allerdings wäre der Zeitaufwand dafür erheblich größer, und da wir nur 6 Wochen Zeit hatten für die Diplomarbeit, war dies nicht machbar. Eine weitere mögliche Verbesserung könnte darin bestehen, einen energiesparenden Sleep-Modus zu implementieren, um den Energieverbrauch zu reduzieren und die Lebensdauer der Batterien zu verlängern.

Was mich am meisten wundert, ist die Langzeit-Betriebstauglichkeit des Prototyps. Wie funktioniert sie, und ob könnten Änderungen notwendig sein, oder zusätzliche Bedienungsanforderungen auftreten, da diese erst nach einiger Zeit oder in bestimmten Situationen und zu bestimmten Zeiten auftreten können.

## 8.2 Testberichte

### 8.2.1 Testbericht Sascha Rytz: Kundendienst, Garage Klaus

1. Wie zuverlässig ist die Identifikation von RFID-Tags in einer Skala von 1-10?

10.

2. Wie einfach ist die Handhabung des RFID-Systems für dir als Mitarbeiter in der Garage?

Ziemlich Simple!

3. Bietet das System eine klare benutzerfreundliche Oberfläche für die Verwaltung von Fahrzeugen?

Super aufgebaut.

4. Ist dieses System verwendbar für die Garage? Wenn nein, Warum?

Praktisch für den Überblick zu halten, man weiß immer, wem der Schlüssel gehört.

5. Vereinfacht dieses System die Schlüssel Identifikation gegenüber dem alten System?

Ja definitiv, dieses Gerät macht vieles einfacher.

6. Deiner Meinung nach, sollten noch weitere Optionen auf der Benutzeroberfläche sein?

Vielleicht nicht mehr, aber etwas angepasster z.B. Modell oder Typ

7. Wie würdest du schätzen, wenn dieses Projekt weitergeführt wird als Privatprojekt, um in Praxis zu verwenden?

Finde ich eine super Sache, wenn jede Garage so etwas hat.

**8.2.2 Testbericht: Marouen Garbaa: Automobilfachmann, Garage Klaus**

1. Wie zuverlässig ist die Identifikation von RFID-Tags in einer Skala von 1-10?

10.

2. Wie einfach ist die Handhabung des RFID-Systems für dir als Mitarbeiter in der Garage?

Es braucht ein wenig Übung, um das ganze richtig zu bedienen.

3. Bietet das System eine klare benutzerfreundliche Oberfläche für die Verwaltung von Fahrzeugen?

Ja ist einfach aufgebaut.

4. Ist dieses System verwendbar für die Garage? Wenn nein, Warum?

Ja ist sehr praktisch z.B., wenn man viele Autos hat.

5. Vereinfacht dieses System die Schlüssel Identifikation gegenüber dem alten System?

Man hat das Vorteil, dass alles elektronisch abgespeichert ist und man muss nicht alles schriftlich machen.

6. Deiner Meinung nach, sollten noch weitere Optionen auf der Benutzeroberfläche sein?

Evtl. Ja, z.B., Chassis Nummer, Farbe, Tel.

7. Wie würdest du schätzen, wenn dieses Projekt weitergeführt wird als Privatprojekt, um in Praxis zu verwenden?

Ja ist ein Vorteil für jede Garage.

Es hätte eine Zukunft für Massenproduktion.

### 8.3 Reflexion Weg zum Ziel

Diese sehr anspruchsvolle Abschlussarbeit belegt, dass ich am Ende meines Studiums als Elektrotechniker HF stehe und das erlernte Wissen erfolgreich in die Praxis umsetzen kann. Nach drei erfolgreichen Studienjahren in Olten habe ich nun meinen Abschluss erreicht.

Ich war sehr erfreut darüber, dass ich am Ende an einem Thema arbeiten durfte, das ich selbst auswählen konnte. Mit großer Begeisterung und Motivation habe ich meine Arbeit begonnen. Obwohl meine Motivation in den sechs Wochen gelegentlich geschwankt hat, fand ich stets Gründe, um weiterzumachen und das Projekt abzuschließen. Schon vor der Auswahl des Themas hatte ich den Wunsch, meine Programmierfähigkeiten zu erweitern.

Als ich dann ein passendes Thema fand, um dieses Ziel zu erreichen, war meine Motivation nicht immer auf ihrem Höhepunkt. Ich musste viele Informationen im Internet recherchieren, um meine Arbeit effizient durchzuführen. Noch anspruchsvoller war es, nicht nur das Arduino-Programmieren zu beherrschen, sondern auch die Benutzeroberfläche zu erstellen, die in einer anderen Programmiersprache entwickelt wurde. Daher musste ich doppelt so viel Zeit investieren, wenn nicht sogar noch mehr, um den Aufbau des Programms zu erlernen. Zuerst erstellte ich die Objekte auf der Benutzeroberfläche per Drag & Drop und programmierte sie dann so, dass sie mit Arduino zusammenarbeiteten. Daher schwankte meine Motivation gelegentlich. Dennoch bereue ich es nicht, da mir das Programmieren der Windows Forms tatsächlich mehr Spaß gemacht hat als das Programmieren mit Arduino. Würde ich das gleiche Thema noch einmal wählen, wenn ich in der Zeit zurückreisen könnte? Nein, definitiv nicht, da die Programmierung viel Zeit in Anspruch nimmt und nicht mein Hauptziel war, als ich mein Studium begann.

Der Aufbau des Prototyps war für mich keine Herausforderung, da ich in meiner Arbeit oft basteln und Werkzeuge erstellen muss, um verschiedene Geräte zu reparieren.

Abschließend bin ich sehr zufrieden mit meiner Arbeit und stolz darauf, dass ich es geschafft habe. Ich habe gelernt, dass alles, was ich in diesen drei Jahren gelernt habe, mir sehr geholfen hat und es mir ermöglicht hat, dieses Projekt erfolgreich abzuschließen. Das zeigt, dass das Gelernte in der Schule sehr praxisnah ist, und das freut mich, denn ich kann in Zukunft mit diesem Wissen eigenständig weitermachen.

## 8.4 Verdankung

Ich möchte mich zunächst bei der Schule TEKO Olten bedanken, die mir und anderen Studierenden die Möglichkeit bietet, ein berufsbegleitendes Studium zu absolvieren. In meinem Fall wäre dies ansonsten unmöglich gewesen.

Zweitens möchte ich mich bei all meinen Lehrern bedanken, die ihre wertvolle Zeit mit mir verbracht haben, um mir alles beizubringen, was ich gelernt habe. Ich schätze ihre Geduld, wenn sie sich Zeit genommen haben, um jedes Problem im Detail zu erklären. Außerdem danke ich meinen Lehrern für ihr Verständnis, dass es für mich möglicherweise doppelt so schwierig war, da ich von weit hergekommen bin, um meinen Traum von einem HF-Studium zu verwirklichen.

Ich möchte auch meinen Arbeitskollegen Marouen Garbaa und Sascha Rytz danken, die mich unterstützt haben und sich Zeit genommen haben, um mit meinem erstellten Prototyp zu arbeiten und mir Feedback dazu zu geben, wie sie ihn empfunden haben und ihre Meinung dazu geäußert haben. Besonderen Dank möchte ich Sascha Rytz aussprechen, da er sich die Zeit genommen hat, meine Arbeit zu lesen und mir Rückmeldung zur Rechtschreibung gegeben hat, damit ich diese verbessern konnte.

Zuletzt, aber keinesfalls weniger wichtig, möchte ich mich auch bei Joe Räber und dem Sekretariat bedanken, die jedes Mal sofortige Hilfe angeboten haben, wenn es während meines Studiums schwierig für mich war.

## 8.5 Eigenständigkeit-Erklärung

Die Verfasserinnen und Verfasser bestätigen mit ihrer Unterschrift, dass die vorliegende Arbeit selbstständig, ohne fremde Hilfe und ohne Benutzung anderer als die angegebenen Hilfsmittel angefertigt wurde.

Die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit ist in gleicher oder ähnlicher Form noch nicht vorgelegt worden.

Unterschriften:

Datum/Ort: 22.10.2023, Düringen

Vorname / Name

Unterschrift 1

Kelmendi Egzon

Kelmendi E

## 9 Verzeichnisse

### 9.1 Abbildungen

|   |    |
|---|----|
| Abbildung 1: Titelblatt .....   | 1  |
| Abbildung 2: Projektablaufplan .....  | 16 |
| Abbildung 3:RFID TAG - Schlüsselanhänger, 13,56MHz.....                               | 19 |
| Abbildung 4:Mifare RC522 Empfänger .....  | 20 |
| Abbildung 5:Funduino MEGA 2560 R3 Mikrocontroller.....                                | 20 |
| Abbildung 6:USB-A Kabel auf USB-B.....  | 21 |
| Abbildung 7:Batteriefach - 6x AA (9V), mit DC-Stecker.....                            | 21 |
| Abbildung 8: Breadboardkabel .....  | 22 |
| Abbildung 9: Simulation mit «Fritzing» .....  | 23 |
| Abbildung 10: RFID Daten Speicherung Blocks „Dump Info“ .....                         | 25 |
| Abbildung 11:Flussdiagramm Software, Arduino.....                                     | 27 |
| Abbildung 12: Code für RFID Modul Kommunikation.....                                  | 28 |
| Abbildung 13:Erstellung eines Boolesche Variable .....                                | 28 |
| Abbildung 14: Initialisierung der Kommunikation und Schnittstelle .....               | 28 |
| Abbildung 15: Loop Code. Hauptprogramm.....   | 29 |
| Abbildung 16:Code für das Schreiben der Daten auf Blocks.....                         | 30 |
| Abbildung 17: Code für das Lesen der Daten auf die Blocks .....                       | 31 |
| Abbildung 18: Benutzeroberfläche.....   | 33 |
| Abbildung 19: Windows Forms Code für die Kommunikation zwischen App und Arduino ..... | 33 |
| Abbildung 20: Lese Knopf Erstellung.....  | 34 |
| Abbildung 21: Write Button Code.....  | 35 |
| Abbildung 22:Auswahl Buttons für Lese oder schreib Modus .....                        | 36 |
| Abbildung 23: RFID-Tag Beispiel: das Lesen der Daten.....                             | 36 |
| Abbildung 24: Beispiel: Schreiben der Daten auf RFID-Tag.....                         | 37 |
| Abbildung 25: Prototyp Aufbau .....   | 38 |

## 9.2 Tabellenverzeichnis

|   |    |
|---|----|
| Tabelle 1: Funktionale und nicht funktionale Anforderungen..... | 13 |
| Tabelle 2: PIN-Belegung.....                                    | 23 |

## 9.3 Linkverzeichnis

Abbildung 1: Microsoft Online-Bilder

Abbildung 2: [RFID TAG - Schlüsselanhänger, 13,56MHz günstig online kaufen | Funduinoshop](#)

Abbildung 3: [RFID-KIT mit Mifare RC522 Empfänger günstig online kaufen | Funduinoshop](#)

Abbildung 4: [CH340 Mikrocontroller Board - kompatibel mit Arduino MEGA2560 R3 günstig online kaufen | Funduinoshop](#)

Abbildung 5: [USB-A Kabel auf USB-B - 150cm, schwarz günstig online kaufen | Funduinoshop](#)

Abbildung 6: [Batteriefach - 6x AA \(9V\), mit DC-Stecker für Mikrocontroller günstig online kaufen | Funduinoshop](#)

Abbildung 7: [40 Stück Breadboardkabel männlich/männlich \(20cm\) günstig online kaufen | Funduinoshop](#)

# 10 Anhang

## 10.1 Statusberichte

**Projekt:** *Schlüsselmanagement-System basierend auf RFID-Tech*

**Statusbericht:** 17.09.2023

|   |  |                       |                           |                       |                     |                |                |                   |  |              |              |              |              |              |                |   |   |   |   |   |
|---|--|-----------------------|---------------------------|-----------------------|---------------------|----------------|----------------|-------------------|--|--------------|--------------|--------------|--------------|--------------|----------------|---|---|---|---|---|
| <b>Projektleiter</b><br>Kelmendi Egzon  | <b>Projektziele:</b> Digitalisierung des Schlüsselmanagements in eine Garage basierend auf RFID-Technologie  | <b>Verteiler</b><br>• |                           |                       |                     |                |                |                   |  |              |              |              |              |              |                |   |   |   |   |   |
| <table style="width: 100%; text-align: center;"> <tr> <td style="width: 15%;"><b>Gesamt-Beurteilung</b></td> <td style="width: 15%;"><b>Projektverlauf</b></td> <td style="width: 15%;"><b>Projektklima</b></td> <td style="width: 15%;"><b>Termine</b></td> <td style="width: 15%;"><b>Risiken</b></td> <td style="width: 15%;"><b>Ressourcen</b></td> </tr> <tr> <td></td> <td>□ <b>X</b> □</td> <td><b>X</b> □ □</td> <td>□ <b>X</b> □</td> <td>□ <b>X</b> □</td> <td><b>X</b> □ □</td> </tr> <tr> <td><b>Tendenz</b></td> <td>➡</td> <td>↗</td> <td>➡</td> <td>↘</td> <td>↗</td> </tr> </table> |  |                       | <b>Gesamt-Beurteilung</b> | <b>Projektverlauf</b> | <b>Projektklima</b> | <b>Termine</b> | <b>Risiken</b> | <b>Ressourcen</b> |  | □ <b>X</b> □ | <b>X</b> □ □ | □ <b>X</b> □ | □ <b>X</b> □ | <b>X</b> □ □ | <b>Tendenz</b> | ➡ | ↗ | ➡ | ↘ | ↗ |
| <b>Gesamt-Beurteilung</b>   | <b>Projektverlauf</b>  | <b>Projektklima</b>   | <b>Termine</b>            | <b>Risiken</b>        | <b>Ressourcen</b>   |                |                |                   |  |              |              |              |              |              |                |   |   |   |   |   |
|   | □ <b>X</b> □   | <b>X</b> □ □          | □ <b>X</b> □              | □ <b>X</b> □          | <b>X</b> □ □        |                |                |                   |  |              |              |              |              |              |                |   |   |   |   |   |
| <b>Tendenz</b>  | ➡  | ↗                     | ➡                         | ↘                     | ↗                   |                |                |                   |  |              |              |              |              |              |                |   |   |   |   |   |
| <b>Aktueller Projektstand</b> <ul style="list-style-type: none"> <li>• Zielscheibe wurde erstellt</li> <li>• Projektablaufplan wurde erstellt und ist in Bearbeitung</li> <li>• Projektstrukturplan wurde erstellt und ist in Bearbeitung</li> </ul>  | <b>Was läuft gut?</b> <ul style="list-style-type: none"> <li>• Dokumentation</li> </ul> <b>Was läuft nicht gut?</b> <ul style="list-style-type: none"> <li>• Zu spät und zu wenig Zeit an die Diplomarbeit zu arbeiten da noch die Semesterarbeit im Fach Robotik auch nächste Woche abgegeben werden muss.</li> </ul> |                       |                           |                       |                     |                |                |                   |  |              |              |              |              |              |                |   |   |   |   |   |
| <b>Geplante nächste Schritte / getroffene Massnahmen</b> <ul style="list-style-type: none"> <li>Software programmieren mit Fundoio und testen.</li> <li>Üben mit Windows Forms, um eine Idee zu bekommen wie die Benutzerfläche aufgebaut konnte damit dies Benutzerfreundlich ist. Dann starten und aufbauen.</li> </ul>   |  |                       |                           |                       |                     |                |                |                   |  |              |              |              |              |              |                |   |   |   |   |   |

Projekt-Statusbericht; Stefan Thöni, Josef Räber

**Projekt:** *Schlüsselmanagement-System basierend auf RFID-Tech.*

**Statusbericht:** 26.09.2023

|  |   |                                    |
|--|---|------------------------------------|
| <b>Projektleiter</b><br>Kelmendi Egzon | <b>Projektziele:</b> Digitalisierung des Schlüsselmanagements in eine Garage basierend auf RFID-Technologie | <b>Verteiler</b><br>• Gloggnor Urs |
|--|---|------------------------------------|

|                                |                       |                     |                |                |                   |
|--------------------------------|-----------------------|---------------------|----------------|----------------|-------------------|
| <b>Gesamt-<br/>Beurteilung</b> | <b>Projektverlauf</b> | <b>Projektklima</b> | <b>Termine</b> | <b>Risiken</b> | <b>Ressourcen</b> |
|                                | □ X □                 | X □ □               | □ □ X          | □ X □          | X □ □             |
| <b>Tendenz</b>                 | ⇒                     | ↗                   | ⇒              | ↘              | ↗                 |

|  |   |
|--|---|
| <b>Aktueller Projektstand</b> <ul style="list-style-type: none"> <li>• Zielscheibe wurde angepasst</li> <li>• Projektstrukturplan wurde erstellt und ist in Bearbeitung</li> <li>• Ich nehme mir online Informationen wie ich eine Benutzeroberfläche mit Windows-Forms erstelle. Also schaue Videos und probiere es selber auch.</li> </ul> | <b>Was läuft gut?</b> <ul style="list-style-type: none"> <li>• Infos für die Erstellung der Benutzeroberfläche. Es sieht aus als wären ziemlich viel Infos online.</li> </ul> <b>Was läuft nicht gut?</b> <ul style="list-style-type: none"> <li>• Zeit Planung. Es läuft nicht nach meinen Plänen. Hatte schon vor bereits die Benutzeroberfläche zu erstellen, aber da ich Probleme bei meine Semesterarbeit hatte, musste ich noch Zeit dort investieren.</li> </ul> |
|--|---|

|  |
|--|
| <b>Gepante nächste Schritte / getroffene Massnahmen</b><br>Software programmieren mit Fundoio und testen.<br>Üben mit Windows Forms, um eine Idee zu bekommen wie die Benutzerfläche aufgebaut konnte damit dies Benutzerfreundlich ist.<br>Dann starten und aufbauen. |
|--|

Projekt-Statusbericht; Stefan Thöni, Josef Räber

**Projekt:** *Schlüsselmanagement-System basierend auf RFID-Tech.*

**Statusbericht:** 01.10.2023

|  |   |  |  |   |  |
|--|---|--|--|---|--|
| <b>Projektleiter</b><br>Kelmendi Egzon   | <b>Projektziele:</b> Digitalisierung des Schlüsselmanagements in eine Garage basierend auf RFID-Technologie   | <b>Verteiler</b><br><ul style="list-style-type: none"> <li>• Gloggner Urs</li> </ul>                     |  |   |  |
| <b>Gesamt-Beurteilung</b>  | <b>Projektverlauf</b><br>  | <b>Projektklima</b><br> | <b>Termine</b><br> | <b>Risiken</b><br> | <b>Ressourcen</b><br> |
| <b>Tendenz</b>   |    |                         |                   |                    |                       |
| <b>Aktueller Projektstand</b> <ul style="list-style-type: none"> <li>• Programm Code für Arduino zum Lesen und beschreiben der Daten auf RFID-Tag wurde geschrieben</li> <li>• Eine Grafik für die Benutzeroberfläche provisorisch erstellt aber noch nicht programmiert.</li> <li>• Vorbereitung für die 2. Vorzeigetermin der Diplomarbeit.</li> </ul> | <b>Was läuft gut?</b> <ul style="list-style-type: none"> <li>• Erstellen der Grafik für die Benutzeroberfläche.</li> <li>• Programmieren von Arduino für READ and Write von Tags.</li> </ul> <b>Was läuft nicht gut?</b> <ul style="list-style-type: none"> <li>• Das Programmieren der Benutzeroberfläche. Es ist zu kompliziert und habe zu wenig Infos wie man eine Verbindung von Windows Forms mit Arduino aufbaut.</li> </ul> |  |  |   |  |
| <b>Geplante nächste Schritte / getroffene Massnahmen</b> <ul style="list-style-type: none"> <li>• Informationen einholen, wie eine Windows Forms App mit Arduino verbunden wird.</li> <li>• Windows Forms App also Benutzeroberfläche programmieren.</li> <li>• Ein Gehäuse für die Hardware zum Auflegen der Tags bauen.</li> </ul>                     |   |  |  |   |  |

Projekt-Statusbericht; Stefan Thöni, Josef Räber

**Projekt:** *Schlüsselmanagement-System basierend auf RFID-Technologie.*

**Statusbericht:** 09.10.2023

| <b>Projektleiter</b><br>Kelmendi Egzon  | <b>Projektziele:</b> Die Digitalisierung des Schlüsselmanagements in einer Garage basiert auf RFID-Technologie.   | <b>Verteiler</b><br><ul style="list-style-type: none"> <li>• Gloggner Urs</li> </ul>  |   |   |   |         |         |            |  |   |   |   |   |   |                |   |   |   |   |   |
|---|---|---|---|---|---|---------|---------|------------|--|---|---|---|---|---|----------------|---|---|---|---|---|
| <table style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 15%;">Gesamt-Beurteilung</th> <th style="width: 15%;">Projektverlauf</th> <th style="width: 15%;">Projektklima</th> <th style="width: 15%;">Termine</th> <th style="width: 15%;">Risiken</th> <th style="width: 15%;">Ressourcen</th> </tr> <tr> <td></td> <td style="text-align: center;"> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> </td> <td style="text-align: center;"> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> </td> <td style="text-align: center;"> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> </td> <td style="text-align: center;"> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> </td> <td style="text-align: center;"> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> </td> </tr> <tr> <td><b>Tendenz</b></td> <td style="text-align: center;">➔</td> <td style="text-align: center;">➔</td> <td style="text-align: center;">➔</td> <td style="text-align: center;">➔</td> <td style="text-align: center;">➔</td> </tr> </table> |   |   | Gesamt-Beurteilung  | Projektverlauf  | Projektklima  | Termine | Risiken | Ressourcen |  | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <b>Tendenz</b> | ➔ | ➔ | ➔ | ➔ | ➔ |
| Gesamt-Beurteilung  | Projektverlauf  | Projektklima  | Termine   | Risiken   | Ressourcen  |         |         |            |  |   |   |   |   |   |                |   |   |   |   |   |
|   | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>   | <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> |         |         |            |  |   |   |   |   |   |                |   |   |   |   |   |
| <b>Tendenz</b>  | ➔   | ➔   | ➔   | ➔   | ➔   |         |         |            |  |   |   |   |   |   |                |   |   |   |   |   |
| <b>Aktueller Projektstand</b> <ul style="list-style-type: none"> <li>• Das Programm "Code" wurde auf der Arduino-Seite geschrieben, während die Windows Forms-Anwendung immer noch in Bearbeitung ist.</li> <li>• Die Dokumentation wird kontinuierlich ergänzt und an die Bedürfnisse angepasst. Zudem wird ein Gehäuse für das RFID-Lese- und Schreibgerät gebaut.</li> </ul>   | <b>Was läuft gut?</b> <ul style="list-style-type: none"> <li>• Das Bauen der Gehäuse und die Dokumentation laufen ziemlich gut.</li> </ul> <b>Was läuft nicht gut?</b> <ul style="list-style-type: none"> <li>• Das Programmieren der Benutzeroberfläche auf Windows Forms, um Daten über die GUI auf das RFID-Tag zu schreiben.</li> <li>• Ein Ersatz-RFID-Tag muss noch bestellt werden, da die RFID-Karte von Arduino nicht brauchbar ist, aber ich finde nicht die gleiche</li> </ul> |   |   |   |   |         |         |            |  |   |   |   |   |   |                |   |   |   |   |   |
| <b>Geplante nächste Schritte / getroffene Massnahmen</b> <p>Software programmieren für die Windows Forms Seite..</p> <p>Gehäuse für mein RFID-Lese- und Schreibgerät bauen. Ich werde mich noch entscheiden, womit ich es am besten bauen kann.</p> <p>Dokumentation weiter bearbeiten und bis zum 23.10.2023 fertigstellen.</p>  |   |   |   |   |   |         |         |            |  |   |   |   |   |   |                |   |   |   |   |   |

Projekt-Statusbericht; Stefan Thöni, Josef Räber

**Projekt:** *Schlüsselmanagement-System basierend auf RFID-Technologie.*

**Statusbericht:** 15.10.2023






|  |   |   |   |   |   |                |                |                   |  |   |   |   |   |   |                |   |   |   |   |   |
|--|---|---|---|---|---|----------------|----------------|-------------------|--|---|---|---|---|---|----------------|---|---|---|---|---|
| <b>Projektleiter</b><br>Kelmendi Egzon   | <b>Projektziele:</b> Die Digitalisierung des Schlüsselmanagements in einer Garage basiert auf RFID-Technologie.   | <b>Verteiler</b><br><ul style="list-style-type: none"> <li>• Gloggner Urs</li> </ul>  |   |   |   |                |                |                   |  |   |   |   |   |   |                |   |   |   |   |   |
| <table style="width: 100%; text-align: center;"> <tr> <td style="width: 15%;"><b>Gesamt-Beurteilung</b></td> <td style="width: 15%;"><b>Projektverlauf</b></td> <td style="width: 15%;"><b>Projektklima</b></td> <td style="width: 15%;"><b>Termine</b></td> <td style="width: 15%;"><b>Risiken</b></td> <td style="width: 15%;"><b>Ressourcen</b></td> </tr> <tr> <td></td> <td> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> </td> <td> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> </td> <td> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> </td> <td> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> </td> <td> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> </td> </tr> <tr> <td><b>Tendenz</b></td> <td>⇒</td> <td>⇒</td> <td>⇒</td> <td>⇒</td> <td>⇒</td> </tr> </table> |   |   | <b>Gesamt-Beurteilung</b>   | <b>Projektverlauf</b>   | <b>Projektklima</b>   | <b>Termine</b> | <b>Risiken</b> | <b>Ressourcen</b> |  | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <b>Tendenz</b> | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ |
| <b>Gesamt-Beurteilung</b>  | <b>Projektverlauf</b>   | <b>Projektklima</b>   | <b>Termine</b>  | <b>Risiken</b>  | <b>Ressourcen</b>   |                |                |                   |  |   |   |   |   |   |                |   |   |   |   |   |
|  | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>   | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> |                |                |                   |  |   |   |   |   |   |                |   |   |   |   |   |
| <b>Tendenz</b>   | ⇒   | ⇒   | ⇒   | ⇒   | ⇒   |                |                |                   |  |   |   |   |   |   |                |   |   |   |   |   |
| <b>Aktueller Projektstand</b> <ul style="list-style-type: none"> <li>• Beide Programm-Codes, sowohl auf dem Arduino als auch in Windows Forms, funktionieren gut.</li> <li>• Die Dokumentation wird kontinuierlich aktualisiert und an die Bedürfnisse angepasst. Außerdem wird ein Gehäuse für das RFID-Lese- und Schreibgerät gebaut.</li> </ul>   | <b>Was läuft gut?</b> <ul style="list-style-type: none"> <li>• Das Bauen der Gehäuse und die Dokumentation laufen ziemlich gut.</li> </ul> <b>Was läuft nicht gut?</b> <ul style="list-style-type: none"> <li>• Ein Ersatz-RFID-Tag musste noch bestellt werden, ist jedoch noch nicht eingetroffen, und es gibt noch keinen Liefertermin.</li> </ul> |   |   |   |   |                |                |                   |  |   |   |   |   |   |                |   |   |   |   |   |
| <b>Geplante nächste Schritte / getroffene Massnahmen</b><br>Ich werde ein Gehäuse für mein RFID-Lese- und Schreibgerät aus Holz herstellen. Anschließend werde ich es vor Ort testen, dabei wird es in der Garage gebaut und von Mitarbeitern getestet. In der Abschlussphase werde ich die Projektabwicklung überwachen, Testberichte verfassen, eine Reflexion durchführen und ein persönliches Schlusswort verfassen."  |   |   |   |   |   |                |                |                   |  |   |   |   |   |   |                |   |   |   |   |   |

Projekt-Statusbericht; Stefan Thöni, Josef Räber

**Projekt:** *Schlüsselmanagement-System basierend auf RFID-Technologie.*

**Statusbericht:** 22.10.2023

|  |   |  |
|--|---|--|
| <b>Projektleiter</b><br>Kelmendi Egzon | <b>Projektziele:</b> Die Digitalisierung des Schlüsselmanagements in einer Garage basiert auf RFID-Technologie. | <b>Verteiler</b><br><ul style="list-style-type: none"> <li>• Gloggner Urs</li> </ul> |
|--|---|--|

|                           |   |   |   |   |   |
|---------------------------|---|---|---|---|---|
| <b>Gesamt-Beurteilung</b> | <b>Projektverlauf</b>   | <b>Projektklima</b>   | <b>Termine</b>  | <b>Risiken</b>  | <b>Ressourcen</b>   |
|                           |  |  |  |  |  |
| <b>Tendenz</b>            | ⇒   | ⇒   | ⇒   | ⇒   | ⇒   |

|   |   |
|---|---|
| <b>Aktueller Projektstand</b> <ul style="list-style-type: none"> <li>• Das Projekt ist fast abgeschlossen</li> <li>• Gehäuse ist fertig.</li> </ul> | <b>Was läuft gut?</b> <ul style="list-style-type: none"> <li>• Das ganze Projekt ist fast fertig.</li> </ul> <b>Was läuft nicht gut?</b> <ul style="list-style-type: none"> <li>• Zeit ist zu wenig und konnte nicht alles machen, was ich wollte.</li> </ul> |
|---|---|

|  |
|--|
| <b>Geplante nächste Schritte / getroffene Massnahmen</b><br>Wird am 22.10.2023 abgegeben da ich am 23.10.2023 arbeite und nicht möglich ist bis um 18:00 Uhr dies abzugeben. |
|--|

Projekt-Statusbericht; Stefan Thöni, Josef Räber

## 10.2 Vollständiger Programm Code

### 10.2.1 Arduino Teil

```
#include <SPI.h>           // Inkludieren der SPI-Bibliothek für die Kommunikation mit dem RFID-Modul
#include <MFRC522.h>       // Inkludieren der MFRC522-Bibliothek für RFID-Kommunikation
#include <deprecated.h>    // Inkludieren einer veralteten Bibliothek (deprecated)
#include <MFRC522Extended.h> // Inkludieren einer erweiterten MFRC522-Bibliothek
#include <require_cpp11.h> // Inkludieren einer Bibliothek, die C++11 erfordert

#define RST_PIN 8         // Definition des Reset-Pins für das RFID-Modul
#define SS_PIN 53        // Definition des Slave-Select-Pins für das RFID-Modul

MFRC522 mfrc522(SS_PIN, RST_PIN); // Erstellung einer MFRC522-Instanz mit den Pins RST und SS
MFRC522::MIFARE_Key key;         // Erstellung eines MIFARE_Key-Objekts für die RFID-Authentifizierung
MFRC522::StatusCode status;     // Erstellung eines Statuscode-Objekts für die RFID-Kommunikation

bool isWritingMode = false;      // Eine Flagge, um zwischen Schreib- und Lese-Modus zu wechseln

void setup() {
  Serial.begin(9600);           // Initialisierung der seriellen Kommunikation mit einer Baudrate von 9600
  SPI.begin();                 // Initialisierung der SPI-Schnittstelle
  mfrc522.PCD_Init();          // Initialisierung des RFID-Moduls
}

void loop() {
  if (isWritingMode) {
    // Code für das Schreiben von Daten auf RFID-Karten ausführen
    writeDataToRFIDCard();
  } else {
    // Code für das Lesen von RFID-Karten ausführen
    readRFIDCard();
  }
}
```

```

}

if (Serial.available() > 0) // Überprüfen, ob Daten von der seriellen Schnittstelle verfügbar sind
{
  char command = Serial.read(); // Ein Zeichen von der seriellen Schnittstelle lesen
  if (command == 'R')
  {
    isWritingMode = false; // In den Lese-Modus wechseln
  }
  else if (command == 'W')
  {
    isWritingMode = true; // In den Schreib-Modus wechseln
  }
}

delay(100); // Zum Beispiel 100 Millisekunden
}
}

/*****Read Part*****/
void readRFIDCard() {
  // Die folgenden Zeilen dienen dazu, den Authentifizierungsschlüssel für die RFID-Karte vorzubereiten
  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

  byte block;
  byte len;
  //-----

  // Wenn keine neue Karte auf dem Sensor/Reader vorhanden ist,
  // wird die Schleife zurückgesetzt, um den gesamten Prozess im Leerlauf zu sparen.
  if ( ! mfr522.PICC_IsNewCardPresent() ) {
    return;
  }
}

```

```
// Eine Karte auswählen
if ( ! mfr522.PICC_ReadCardSerial() ) {
    return;
}

// Hier werden Puffer für verschiedene RFID-Blöcke initialisiert
byte buffer1[18]; // Puffer für Block 1
byte buffer2[18]; // Puffer für Block 2
byte buffer4[18]; // Puffer für Block 4
byte buffer5[18]; // Puffer für Block 5

// Block 1 lesen
block = 1;
len = 18;

// Authentifizierung mit dem Schlüssel und UID der Karte
status = mfr522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1, &key, &(mfr522.uid));
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Authentifizierung fehlgeschlagen: "));
    Serial.println(mfr522.GetStatusCodeName(status));
    return;
}

// Block 1 lesen
status = mfr522.MIFARE_Read(block, buffer1, &len);
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Lesen fehlgeschlagen: "));
    Serial.println(mfr522.GetStatusCodeName(status));
    return;
}

String value = "";
for (uint8_t i = 0; i < 16; i++)
```

```
{
    value += (char)buffer1[i];
}

// Block 2 lesen
block = 2;
len = 18;

// Authentifizierung für Block 2
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 2, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Authentifizierung fehlgeschlagen für Block 2: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}

// Block 2 lesen
status = mfrc522.MIFARE_Read(block, buffer2, &len);
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Lesen fehlgeschlagen für Block 2: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}

for (uint8_t i = 0; i < 16; i++) {
    value += (char)buffer2[i];
}

// Block 4 lesen
block = 4;
len = 18;

// Authentifizierung für Block 4
```

```
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 4, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Authentifizierung fehlgeschlagen für Block 4: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}

// Block 4 lesen
status = mfrc522.MIFARE_Read(block, buffer4, &len);
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Lesen fehlgeschlagen für Block 4: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}

for (uint8_t i = 0; i < 16; i++) {
    value += (char)buffer4[i];
}

// Block 5 lesen
block = 5;
len = 18;

// Authentifizierung mit dem Schlüssel und UID der Karte
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 5, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Authentifizierung fehlgeschlagen: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}

// Block 5 lesen
status = mfrc522.MIFARE_Read(block, buffer5, &len);
```

```
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Lesen fehlgeschlagen: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}

for (uint8_t i = 0; i < 16; i++)
{
    value += (char)buffer5[i];
}

value.trim();
Serial.println(value);

delay(1000);

mfrc522.PICC_HaltA();
mfrc522.PCD_StopCrypto1();
}
```

```
/******Write Part******/
```

```
void writeDataToRFIDCard()
{
    // Prepare key - all keys are set to FFFFFFFFh at chip delivery from the factory.
    MFRC522::MIFARE_Key key;
    for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

    // Reset the loop if no new card present on the sensor/reader. This saves the entire process when idle.
    if ( ! mfrc522.PICC_IsNewCardPresent() )
    {
        return;
    }
}
```

```
}
// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial()
{
    return;
}

Serial.print("Card UID:"); //Dump UID
for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], DEC);
}

byte buffer[34];
byte block;
MFRC522::StatusCode status;
byte len;

//*****Block 1: Name*****//

Serial.setTimeout(10000L); // wait 10 seconds for input from serial
len = Serial.readBytesUntil('*', (char *) buffer, 30); // read family name from serial
for (byte i = len; i < 30; i++) buffer[i] = ' '; // pad with spaces

block = 1;

status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
    Serial.print("PCD_Authenticate() failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}
else Serial.println(F("PCD_Authenticate() success: "));
```

```
// Write block
status = mfr522.MIFARE_Write(block, buffer, 16);
if (status != MFRC522::STATUS_OK) {
    Serial.print("MIFARE_Write() failed: ");
    Serial.println(mfr522.GetStatusCodeName(status));
    return;
}
else Serial.println("MIFARE_Write() success: ");

//*****Block 2: Vorname*****//

Serial.setTimeout(10000L); // wait 10 seconds for input from serial
len = Serial.readBytesUntil('*', (char *) buffer, 30); // read family first name from serial
for (byte i = len; i < 30; i++) buffer[i] = ' '; // pad with spaces

block = 2;

status = mfr522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block, &key, &(mfr522.uid));
if (status != MFRC522::STATUS_OK) {
    Serial.print("PCD_Authenticate() failed: ");
    Serial.println(mfr522.GetStatusCodeName(status));
    return;
}
else Serial.println(F("PCD_Authenticate() success: "));

// Write block
status = mfr522.MIFARE_Write(block, buffer, 16);
if (status != MFRC522::STATUS_OK) {
    Serial.print("MIFARE_Write() failed: ");
    Serial.println(mfr522.GetStatusCodeName(status));
    return;
}
```

```
}
else Serial.println("MIFARE_Write() success: ");

//*****Block 4: Kennzeichen*****//

Serial.setTimeout(10000L); // wait 10 seconds for input from serial
len = Serial.readBytesUntil('*', (char *) buffer, 30); // read plate number name from serial
for (byte i = len; i < 30; i++) buffer[i] = ' '; // pad with spaces

block = 4;

status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
  Serial.print("PCD_Authenticate() failed: ");
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
else Serial.println(F("PCD_Authenticate() success: "));

// Write block
status = mfrc522.MIFARE_Write(block, buffer, 16);
if (status != MFRC522::STATUS_OK) {
  Serial.print("MIFARE_Write() failed: ");
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
else Serial.println("MIFARE_Write() success: ");

//*****Block 5: Typ*****//

Serial.setTimeout(10000L); // wait 10 seconds for input from serial
len = Serial.readBytesUntil('*', (char *) buffer, 30); // read Typ from serial
```

```
for (byte i = len; i < 30; i++) buffer[i] = ' '; // pad with spaces

block = 5;

status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block, &key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
    Serial.print("PCD_Authenticate() failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}
else Serial.println(F("PCD_Authenticate() success: "));

// Write block
status = mfrc522.MIFARE_Write(block, buffer, 16);
if (status != MFRC522::STATUS_OK) {
    Serial.print("MIFARE_Write() failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}
else Serial.println("MIFARE_Write() success: ");

mfrc522.PICC_HaltA(); // Halt PICC
mfrc522.PCD_StopCrypto1(); // Stop encryption on PCD
}
```

### 10.2.2 Windows Forms Teil

```
using System;  
using System.IO.Ports;  
using System.Text;  
using System.Windows.Forms;
```

```
namespace Schlüsselmanagement_System  
{  
    public partial class Form1 : Form  
    {  
        private SerialPort Port;          // Initialisierung des SerialPort-Objekts  
        private bool isWritingMode = false; // Initialisierung eines Schreibmodus-Flags  
        public Form1()  
        {  
            InitializeComponent();          // Initialisierung der Windows Forms-Oberfläche  
            Port = new SerialPort("COM3", 9600); // Konfiguration des SerialPort-Objekts für COM3 mit einer Baudrate von 9600  
        }  
  
        private void Read_btn_Click(object sender, EventArgs e)  
        {  
            try  
            {  
                if (!Port.IsOpen)          // Überprüfen, ob die serielle Schnittstelle nicht geöffnet ist  
                {  
                    Port.Open();          // Öffnen der seriellen Schnittstelle  
                }  
  
                Port.WriteLine("R");      // Fordere den Arduino auf, in den Lesemodus zu wechseln  
  
                // Lesen der Daten aus allen Blöcken  
                string receivedDataBlock1 = Port.ReadLine(); // Block 1 lesen  
                string receivedDataBlock2 = Port.ReadLine(); // Block 2 lesen  
            }  
        }  
    }  
}
```

```
string receivedDataBlock4 = Port.ReadLine(); // Block 4 lesen
string receivedDataBlock5 = Port.ReadLine(); // Block 5 lesen

// Anzeigen der Daten in den Textboxen
textBox1.Text = receivedDataBlock1; // Block 1 in textBox1 anzeigen
textBox2.Text = receivedDataBlock2; // Block 2 in textBox2 anzeigen
textBox3.Text = receivedDataBlock4; // Block 4 in textBox3 anzeigen
textBox4.Text = receivedDataBlock5; // Block 5 in textBox4 anzeigen

}
catch (Exception ex)
{
    MessageBox.Show("Fehler: " + ex.Message); // Anzeigen einer Fehlermeldung bei einem Fehler
}

finally
{
    if (Port.IsOpen)
    {
        Port.Close(); // Schließen der seriellen Schnittstelle (wenn geöffnet)
    }
}

}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    if (Port.IsOpen)
    {
        Port.Close();
    }
}

// Hilfsfunktion, um den ASCII-Text aus HEX-Daten zu extrahieren
```

```
private string ExtractAsciiTextFromBlock1(string hexData)
{
    try
    {
        // Extrahiere die HEX-Daten (die ersten 16 Zeichen)
        hexData = hexData.Substring(0, 16);

        // Verwende die Methode ExtractAsciiText, um den ASCII-Text zu extrahieren
        string asciiText = ExtractAsciiText(hexData);

        return asciiText;
    }
    catch
    {
        return "Ungültige Daten";
    }
}
private string ExtractAsciiTextFromBlock2(string hexData)
{
    try
    {
        // Extrahiere die HEX-Daten (die ersten 16 Zeichen)
        hexData = hexData.Substring(0, 16);

        // Verwende die Methode ExtractAsciiText, um den ASCII-Text zu extrahieren
        string asciiText = ExtractAsciiText(hexData);

        return asciiText;
    }
    catch
    {
        return "Ungültige Daten";
    }
}
```

```
}  
// Hilfsfunktion, um den ASCII-Text aus HEX-Daten zu extrahieren  
private string ExtractAsciiText(string hexData)  
{  
    try  
    {  
        // Entferne Leerzeichen und Zeilenumbrüche aus den HEX-Daten  
        hexData = hexData.Replace(" ", "").Replace("\r", "").Replace("\n", "");  
  
        // Konvertiere HEX in einen Byte-Array  
        byte[] byteData = new byte[hexData.Length / 2];  
        for (int i = 0; i < byteData.Length; i++)  
        {  
            byteData[i] = Convert.ToByte(hexData.Substring(i * 2, 2), 16);  
        }  
  
        // Dekodiere das Byte-Array in ASCII-Text  
        string asciiText = Encoding.ASCII.GetString(byteData);  
  
        return asciiText;  
    }  
    catch  
    {  
        return "Ungültige Daten";  
    }  
}  
  
private void textBox1_TextChanged(object sender, EventArgs e)  
{  
  
}  
  
private void textBox2_TextChanged(object sender, EventArgs e)
```

```
{  
  
}  
  
private void textBox3_TextChanged(object sender, EventArgs e)  
{  
  
}  
  
private void textBox4_TextChanged(object sender, EventArgs e)  
{  
  
}  
  
private void Write_btn_Click(object sender, EventArgs e)  
{  
    // Überprüfen, ob die Anwendung sich im Schreibmodus befindet  
    if (isWritingMode)  
    {  
        // Überprüfen, ob Text vorhanden ist  
        if (!string.IsNullOrEmpty(textBox1.Text))  
  
            try  
            {  
                if (!Port.IsOpen)  
                {  
                    Port.Open();  
                }  
  
                // Lese den Text aus textBoxen  
                string textToWriteBlock1 = textBox1.Text;  
                string textToWriteBlock2 = textBox2.Text;  
                string textToWriteBlock4 = textBox3.Text;
```

```
string textToWriteBlock5 = textBox4.Text;

// Senden der Texte an den Arduino über die serielle Schnittstelle
Port.WriteLine(textToWriteBlock1); // Schreiben Sie den Text auf Block 1
Port.WriteLine(textToWriteBlock2); // Schreiben Sie den Text auf Block 2
Port.WriteLine(textToWriteBlock4); // Schreiben Sie den Text auf Block 4
Port.WriteLine(textToWriteBlock5); // Schreiben Sie den Text auf Block 5

//Löschen der Texte aus den TextBox-Elementen, nachdem sie gesendet wurden
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
}
catch (Exception ex)
{
    MessageBox.Show("Fehler: " + ex.Message);
}
finally
{
    if (Port.IsOpen)
    {
        Port.Close();
    }
}
}
else
{
    MessageBox.Show("Bitte geben Sie einen Text ein, um ihn auf das RFID-Tag zu schreiben.");
}
}
// Helper function to convert text to HEX
```

```
private string ConvertTextToHex(string text)
{
    byte[] bytes = Encoding.ASCII.GetBytes(text);
    string hex = BitConverter.ToString(bytes).Replace("-", "");
    return hex;
}

private void button1_Click(object sender, EventArgs e)
{
    // Lösche den Inhalt der Textboxen
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
}

private void write_modus_btn_Click(object sender, EventArgs e)
{
    // Fordere den Arduino auf, in den Schreibmodus zu wechseln
    if (!Port.IsOpen)
    {
        Port.Open();
    }
    Port.WriteLine("W");

    // Aktiviere den Schreibmodus
    isWritingMode = true;
}

private void read_modus_btn_Click(object sender, EventArgs e)
{

```

```
// Fordere den Arduino auf, in den Lese-Modus zu wechseln
if (!Port.IsOpen)
{
    Port.Open();    // Öffnen der seriellen Schnittstelle
}
Port.WriteLine("R"); // Senden eines Befehls an den Arduino, in den Lesemodus zu wechseln
Port.Close();     // Schließen der seriellen Schnittstelle

// Deaktivieren des Schreibmodus
isWritingMode = false;
}

private void label5_Click(object sender, EventArgs e)
{
}

private void label4_Click(object sender, EventArgs e)
{
}
}
}
```