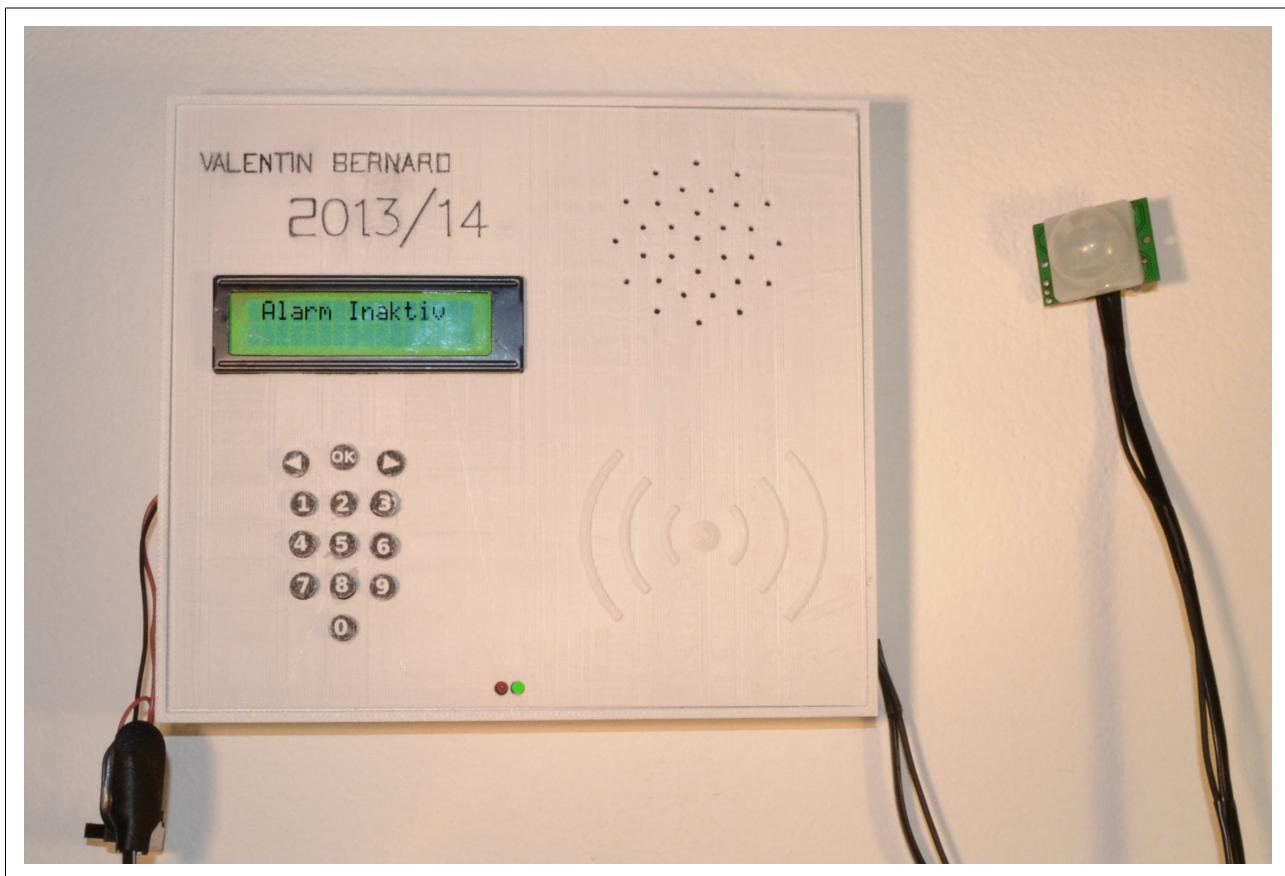


# Projektdokumentation:

## Alarmanlage

*Ein Projekt von Valentin Bernard  
GOB Meran 2013/14*



# Inhaltsverzeichnis

1) Projektbeschreibung.....	3
1.1) Ziel des Projekts.....	3
1.2) Erster Entwurf.....	3
1.2.1) Hardware.....	3
1.2.2) Software.....	3
1.2.3) Zusatzfunktionen.....	4
1.3) Realisiertes Projekt.....	4
1.3.1) Funktionsbeschreibung.....	4
1.3.2) Hardware.....	4
1.3.3) Software.....	5
2) Zeitplanung.....	6
2.1) Gantt-Diagramm.....	6
3) Technische Details.....	8
3.1) Hardware.....	8
3.1.1) Blockschaltbild.....	8
3.1.2) Schaltplan.....	9
3.1.3) Schaltung: Stromversorgung.....	10
3.1.4) Schaltung: Controller.....	10
3.1.5) Schaltung: Lautsprecher.....	12
3.1.6) Schaltung: Taster.....	13
3.1.7) Schaltung: LCD-Display.....	13
3.1.8) Schaltung: RFID-Lesegerät.....	14
RFID-Lesegerät.....	14
3.1.9) Schaltung: Bewegungsmelders.....	15
Bewegungsmelder.....	15
3.1.10) Vollständige Bauteilliste.....	16
3.1.11) Board.....	17
Fehler in der ersten Version.....	17
Board Version zwei.....	18
3.1.12) Gehäuse.....	18
3.2) Software.....	19
3.2.1) Grundstruktur.....	19
3.2.2) Bewegungsmelder.....	20
3.2.3) Tonausgabe.....	20
3.2.4) Passwort.....	20
Ändern des Passwortes.....	22
3.2.5) RFID.....	22
SPI-Initialisierung.....	22
SPI-Kommunikation.....	23
RFID-Request.....	24
4) Kostenberechnung.....	25
4.1) Notwendige Verbesserungen und mögliche Erweiterungen.....	26
5) Anlagen.....	27

# 1) Projektbeschreibung

## 1.1) Ziel des Projekts

Im Rahmen des Fachs Technisches Zeichnen und Projektieren muss jeder Schüler der Elektronik-Fachrichtung der Gewerbeoberschule Meran in der fünften Klasse ein selbst gewähltes Abschlussprojekt realisieren. Ich habe mich für eine Alarmanlage entschieden.

Ziel des Projekts war es, eine Anlage zu entwickeln, welche das Eindringen einer Person in einen Raum erkennt und dies akustisch meldet. Dabei sollte sich die Anlage per Knopfdruck von jedem aktivieren lassen, aber nur die Eingabe eines Passcodes oder das Annähern eines RFID-Tags sollten sie deaktivieren können.

## 1.2) Erster Entwurf

### 1.2.1) Hardware

Im erstem Entwurf war eine Anlage, bestehend aus einer Zentrale und diversen angeschlossenen Sensoren vorgesehen. Als Sensoren waren handelsübliche Infrarot-Bewegungsmelder oder Lichtschranken vorgesehen..

Die Zentrale sollte aus einer Platine in einem 3D-gedruckten Gehäuse bestehen, welche mehrere Eingänge für Sensoren, ein Tastenfeld für die Eingabe des Deaktivierungscodes (1-9 und „Enter“), einen Taster für die Aktivierung der Anlage, ein RFID-Lesegerät, einen Lautsprecher sowie eine rote und eine grüne LED zur Statusanzeige beinhalten sollte. Ein LCD-Bildschirm war lediglich als Zusatzfunktion geplant.

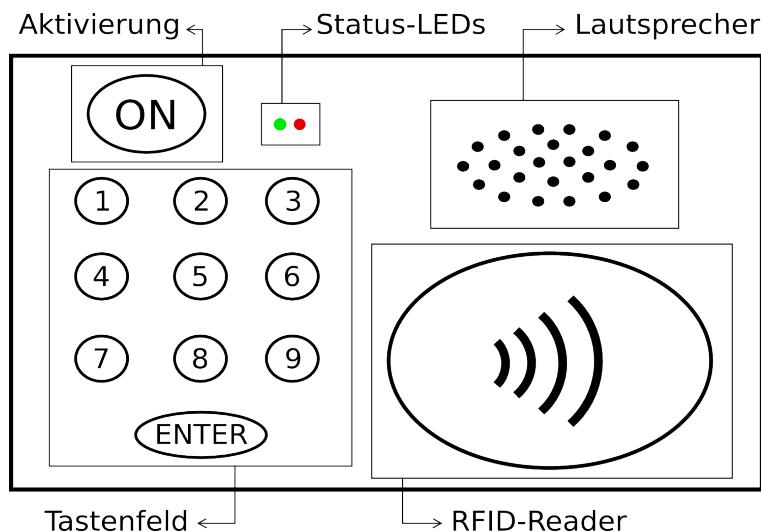


Abbildung 1: Erster Hardware-Entwurf

### 1.2.2) Software

Die Alarmanlage sollte, nach der Aktivierung durch einen Knopfdruck, ständig sämtliche Sensoreingänge kontrollieren. Sobald ein Sensor aktiviert wird, sollte ein Alarm ausgegeben werden. Nach 5 Minuten ohne Sensorsignal sollte sich der Alarm wieder ausschalten, um unnötige Lärmbelästigung zu vermeiden.

Die Deaktivierung der Anlage sollte über die Eingabe eines selbst gewählten Passworts oder über ein RFID-Tag erfolgen, welches vom eingebauten RFID-Lesegerät erkannt wird. Das gewählte Passwort sollte nach Ausfall der Stromversorgung erhalten bleiben.

Die Status-LEDs sollten den Zustand der Anlage anzeigen: Rot für „Aktiv“, grün für „Inaktiv“. Dies war im ersten Entwurf die einzige Form der Visualisierung des Zustands.

### 1.2.3) Zusatzfunktionen

Im ersten Entwurf war vorgesehen, nach Realisierung der Grundfunktionen weitere Zusatzfunktionen zu implementieren sofern dafür noch Zeit zur Verfügung stand. Die angedachten Funktionen waren: die Aktivierung des Alarms nach mehrmaliger Falscheingabe des Codes, Deaktivierung über Funk oder sogar die Versendung eines Alarm-SMS über ein GSM-Modul.

## 1.3) Realisiertes Projekt

Die realisierte Anlage entspricht in großen Teilen dem ersten Entwurf, jedoch unterscheidet sie sich in einigen Details, auf welche im Folgenden grob eingegangen wird.

### 1.3.1) Funktionsbeschreibung

Die Funktion der Anlage entspricht der Aufgabenstellung. Die Anlage erkennt, solange sie aktiviert ist, das Eintreten einer Person oder eines Tiers in einen Raum mit Hilfe eines Bewegungsmelders und gibt darauf hin über einen Lautsprecher einen relativ lauten akustischen Alarm aus. Die Aktivierung erfolgt, wie vorgesehen, per Knopfdruck, und die Deaktivierung erfolgt per Eingabe des selbst gewählten Passwortes. Die Deaktivierung mit RFID-Tag funktioniert aus bisher ungeklärten Gründen noch nicht, Näheres dazu im Kapitel „Software – RDIF“(S.??). Die Anlage benötigt eine externe Gleichstromversorgung von 9-12V.

Von den Zusatzfunktionen wurde bisher keine implementiert, jedoch könnte mit der bestehenden Hardware die Funktion „Aktivierung durch Falscheingabe“ realisiert werden. Nachdem nur ein Prototyp der Platine gefertigt wurde, ist es unmöglich weitere Funktionen hinzuzufügen, da diese eine Erweiterung der Hardware voraussetzen.

### 1.3.2) Hardware

Die Hardware der Anlage unterscheidet sich in einigen Punkten vom ersten Entwurf. Ein Unterschied besteht darin, dass lediglich ein Sensoranschluss auf der Platine vorgesehen ist, nämlich der für den Infrarot-Bewegungsmelder. Außerdem wurde Der LCD-Bildschirm bereits in die erste Version integriert, um eine zusätzliche Vereinfachung der Bedienung zu erzielen. Weitere kleinere Änderungen betreffen die Ergänzung einer Vor- und einer Zurück-Taste sowie die Verschmelzung des Aktivieren-Knopfs mit der OK-Taste. Der Rest der Hardware entspricht dem ersten Entwurf.

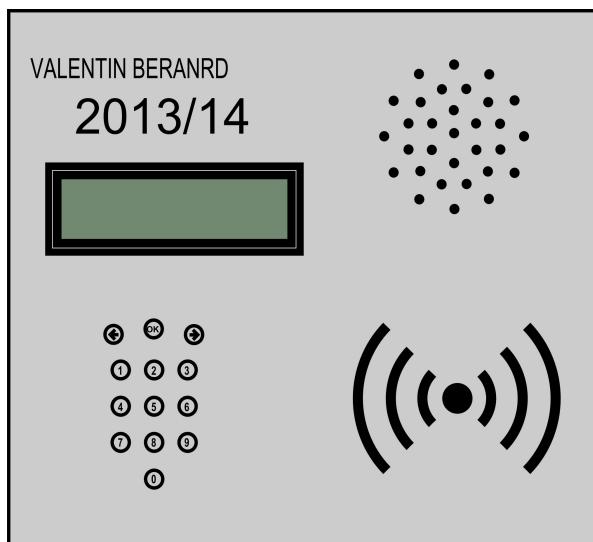


Abbildung 3: Gehäusedeckel

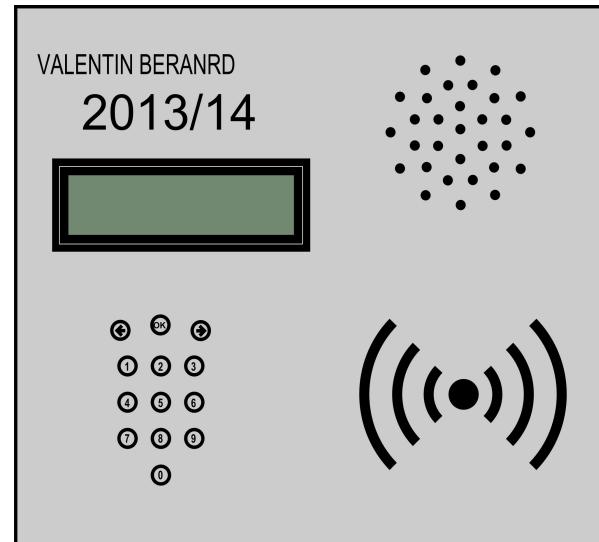


Abbildung 3: Platinen und Lautsprecher

### 1.3.3) Software

Die Alarmanlage wertet, solange sie aktiv ist, ständig den angeschlossenen Bewegungsmelder aus. Wird dieser ausgelöst wird über den eingebauten Lautsprecher ein Alarm ausgegeben welcher wie geplant nach 5 Minuten ohne Bewegung verstummt.

Die Deaktivierung über das Passwort sowie die Möglichkeit der Abänderung desselben wurde erfolgreich implementiert, jedoch wird das Passwort bei Unterbrechung der Stromversorgung auf den Standartwert zurückgesetzt. Die Deaktivierung über ein RFID-Tag bereit einige Probleme, doch die Kommunikation mit dem Lesegerät über die SPI-Schnittstelle funktioniert einwandfrei. Das Problem liegt nur noch beim Übermitteln der richtigen Befehle an das Lesegerät.

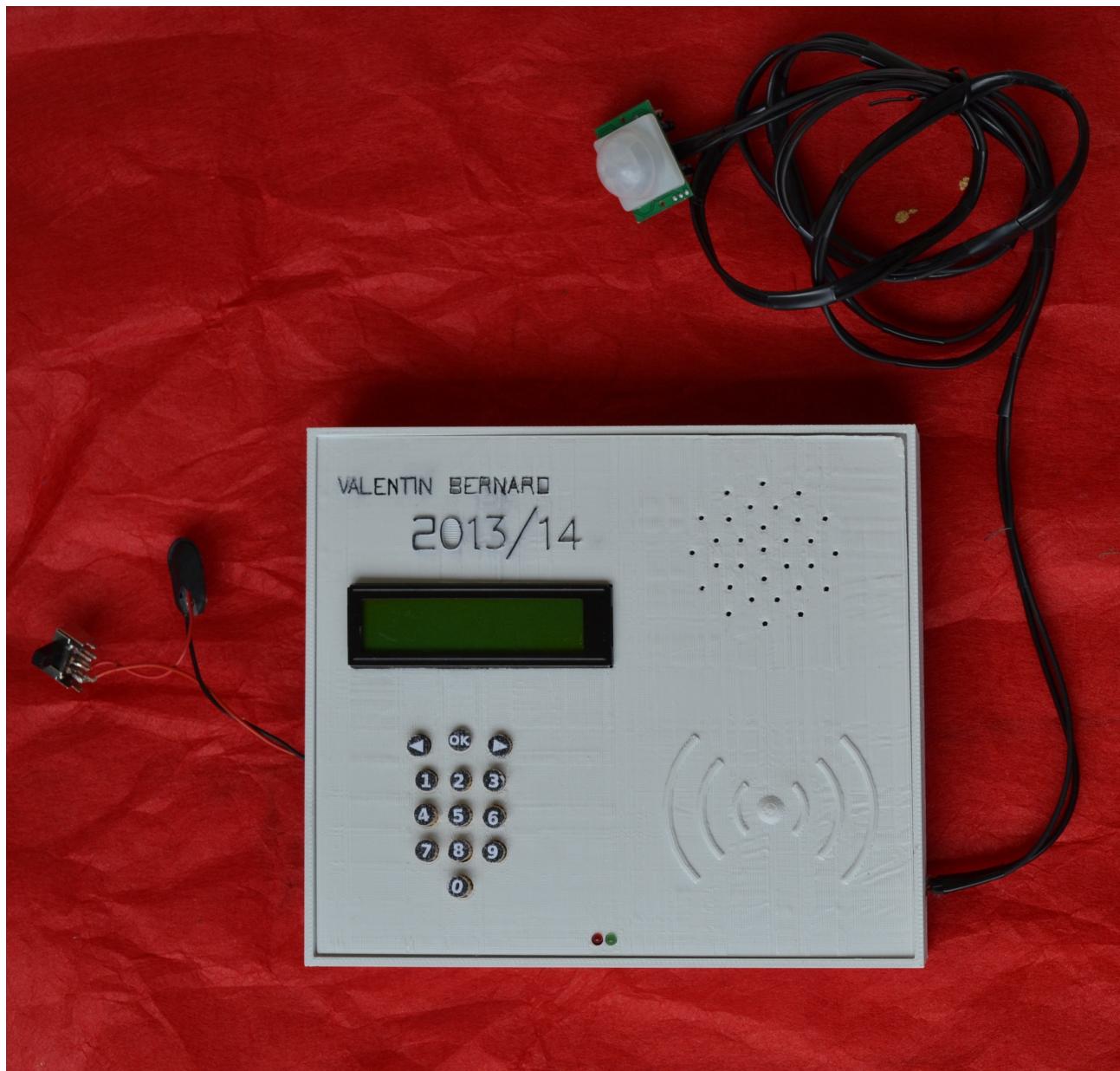


Abbildung 4: Endresultat in 3D-geruckter Hülle

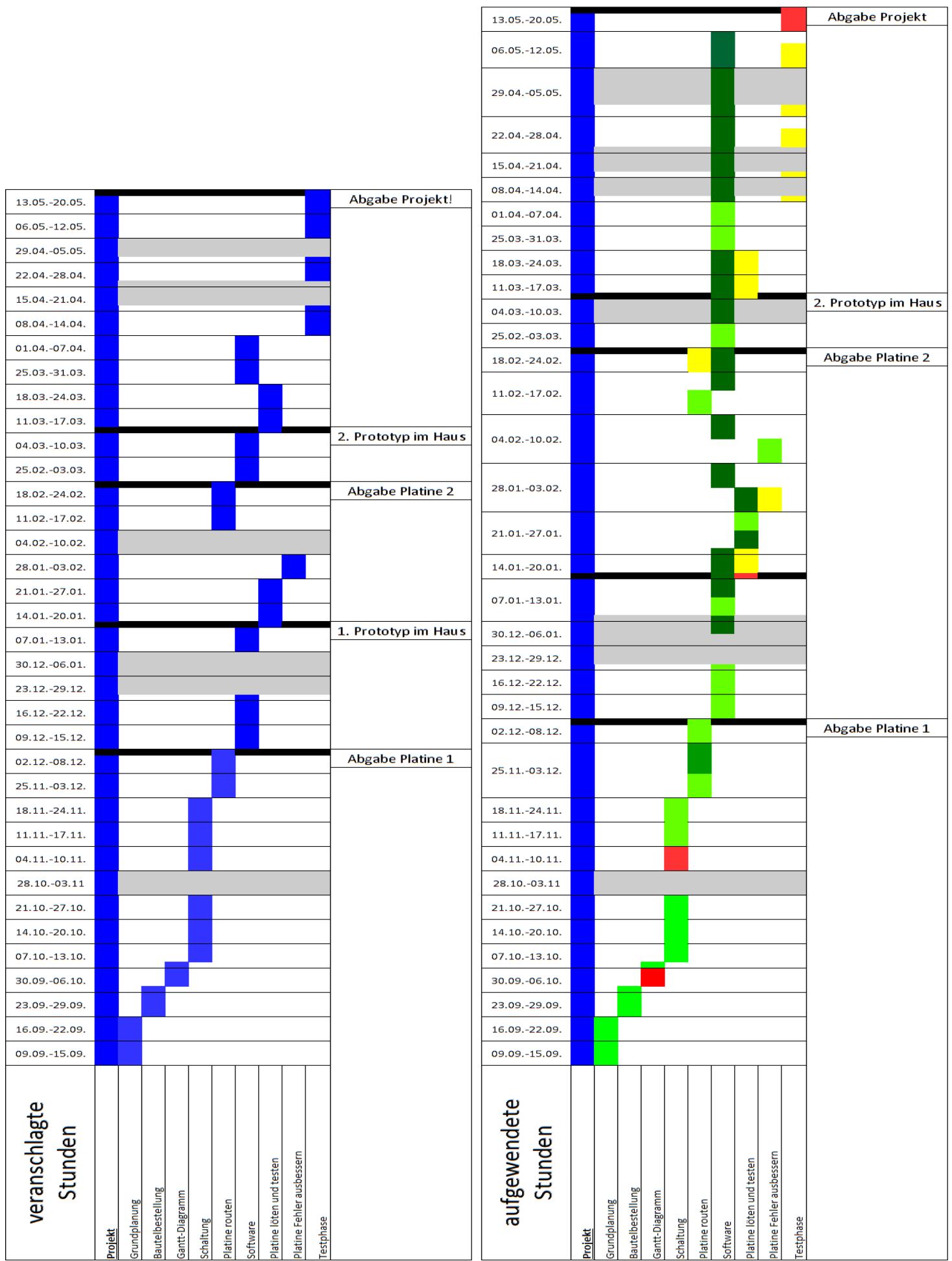
## 2) Zeitplanung

An Anfang des Projekts wurde zur Zeitplanung ein Gantt-Diagramm erstellt, in welchem die veranschlagten Stunden für die einzelnen Teile des Projekts eingetragen wurden. Im Laufe der Projektarbeit ergaben sich einiger Verschiebungen, und es mussten an manchen stellen Überstunden eingefügt oder Stunden gekürzt werden. Insgesamt wurden 121 Stunden Veranschlagt und etwa 163 benötigt. Es wurde viel mehr Zeit als angenommen für die Software aufgewendet, davon flossen über 80% in die letztendlich erfolglose Aktivierung des RFID-Lesegeräts. Das Projekt hätte ansonsten bereits im März abgeschlossen werden können.

### 2.1) Gantt-Diagramm

Legende	
Geplante Zeit	
Eingehalener Zeitplan	
Überstunden	
Ausgefallene Stunden	
Gekürzte Stunden	
Meilensteine	
Ferien	

Wenn man die beiden Gantt-Diagramme, jenes mit den veranschlagten Stunden und jenes mit den tatsächlich aufgewendeten Stunden gegenübergestellt, lässt sich neben den bereits erwähnten längeren Entwicklungszeit auch ablesen, wie viel mehr Zeit in die Entwicklung der Software gesteckt wurde. Die Testphase, eigentlich eine Pufferzone, wurde vollständig mit Softwareentwicklung ersetzt, und es wurden in den letzten Monaten eine Menge Überstunden für die Software aufgewandt. Umso frustrierender ist die Tatsache dass das Projekt trotzdem nicht vollständig funktioniert.



### 3) Technische Details

#### 3.1) Hardware

##### 3.1.1) Blockschaltbild

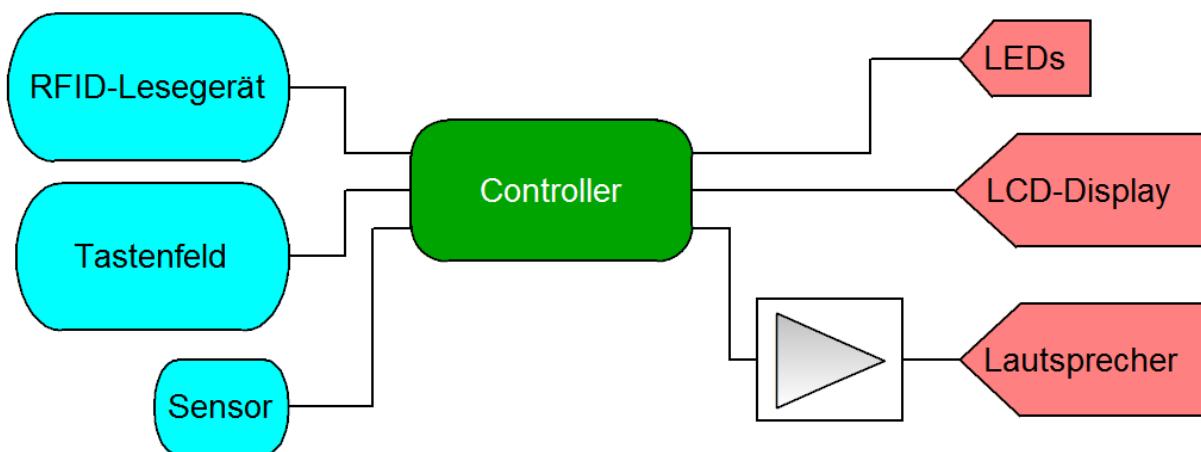


Abbildung 5: Blockschaltbild

Die Schaltung besteht im Wesentlichen aus sechs Schaltblöcken: Einem Tastenfeld, einem PIR-Bewegungssensor und einem RFID-Lesegerät als Eingänge und einer Verstärkung mit Lautsprecher, zwei Status-LEDs und einem LCD-Display als Ausgabe. Die Blöcke sind alle über den Mikrocontroller miteinander verbunden.

RFID-Lesegerät und Sensor wurden angekauft, die anderen Schaltblöcke wurden selbst entworfen. Dazu kommt noch die Schaltung für die Stromversorgung, welche eine Eingangsspannung von 12 Volt in eine 5V und eine 3,3V-Spannung umwandelt. Auch die 12V Eingangsspannung werden weiter verwendet.



Abbildung 6: Blockschaltbild Versorgung

Die 12 Volt werden extern von einem Netzteil bezogen. Da Anlage ist bei Eingangsspannungen von 6-18 Volt voll funktionsfähig, wurde jedoch für 9-12 Volt ausgelegt.

### 3.1.2) Schaltplan

Der Schaltplan für das Projekt wurde mit Eagle™ 6.5.0 erstellt. Die meisten Bauteile waren in der Standardbibliothek zu finden, für den Mikrocontroller wurde zusätzlich eine Bibliothek von Motorola verwendet.

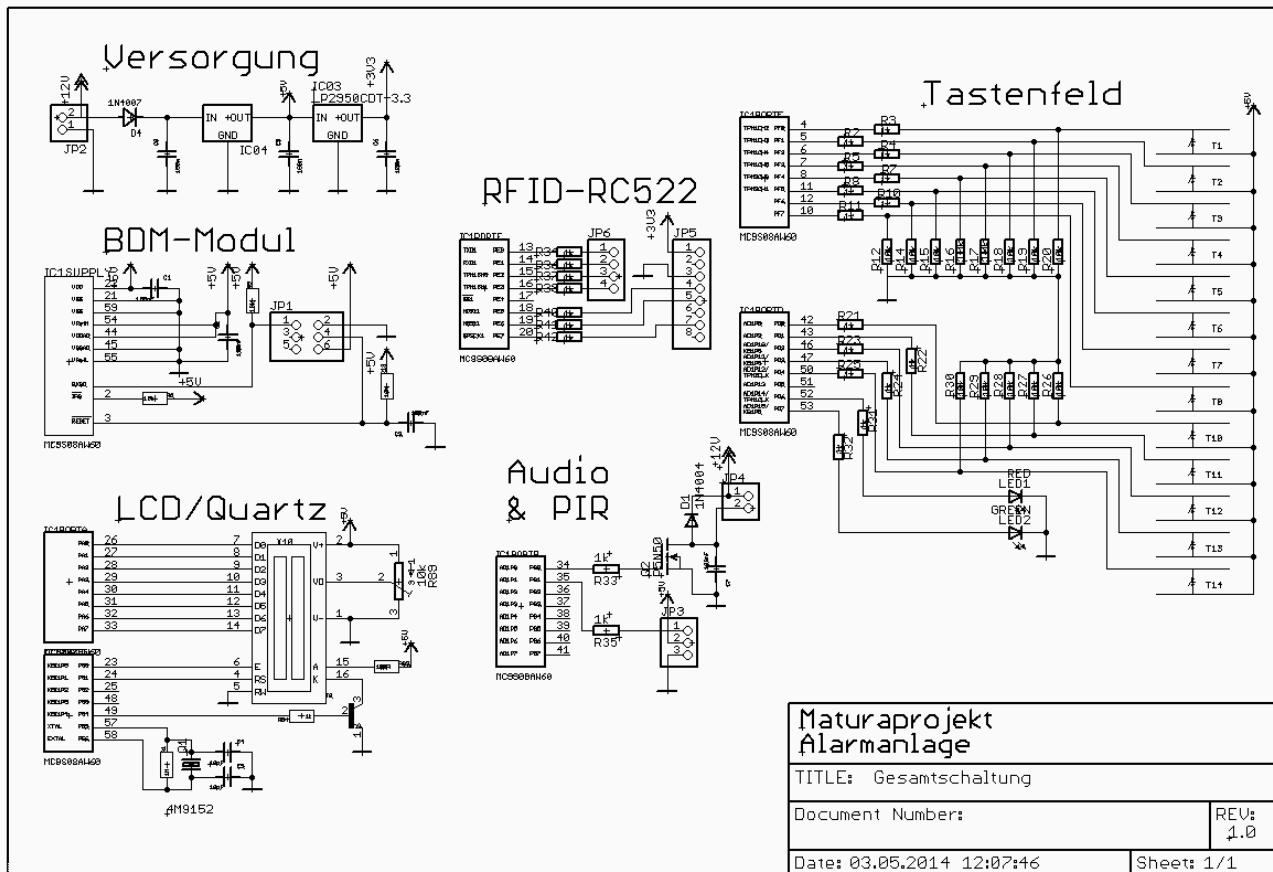


Abbildung 7: Vollständiger Schaltplan Ein Ausdruck in besserer Qualität befindet sich in den Anlagen

Die Planung der Schaltung erwies sich bei diesem Projekt als moderat aufwendig, da der selbe Controller und das selbe LCD-Display verwendet werden wie auf dem Demoboard, weshalb die entsprechenden Schaltblöcke direkt vom Demoboard übernommen werden konnten. Der Rest der Schaltung besteht großteils aus Tastern und Steckerleisten. Der einzige Schaltblock der einige Entwicklungsarbeit erforderte war jener zur Ausgabe des Alarms über einen Lautsprecher.

### 3.1.3) Schaltung: Stromversorgung

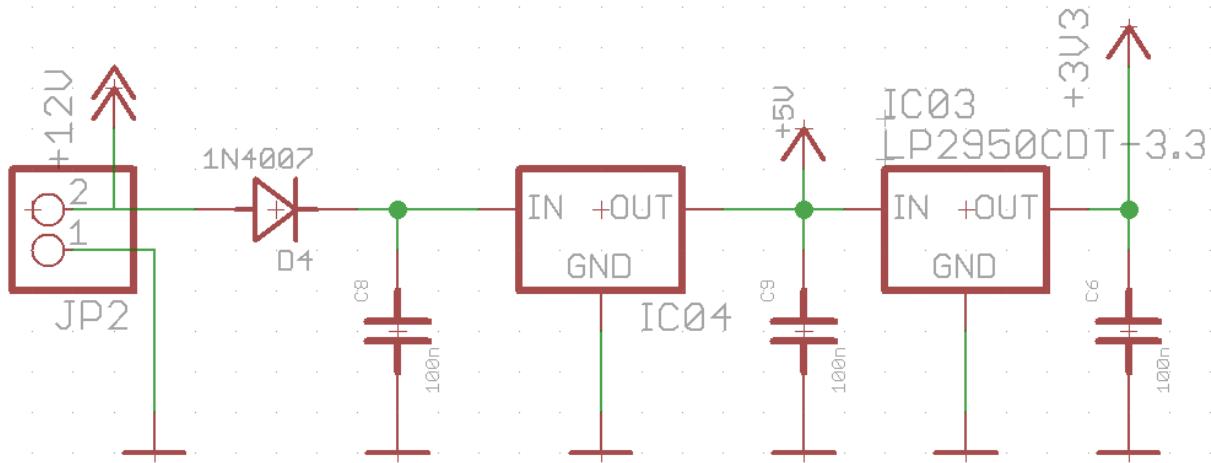


Abbildung 8: Schaltung zur Stromversorgung

Die Stromversorgung wird über einen 5V und einen 3,3V-Spannungsregler geregelt. Dazu wird eine externe 9-12V-Gleichspannungsquelle benötigt. Als 5V-Regler wird der 78S05-Regler im TO220-Gehäuse verwendet, während der 3,3V-Regler LM1112 in SMD eingebaut wurde.

Die 9-12V Eingangsspannung werden zur Steuerung des Lautsprechers verwendet, der Controller, der LCD und der Bewegungsmelder nutzen die 5V, die 3,3V werden lediglich für das RFID-Lesegerät benötigt.

### 3.1.4) Schaltung: Controller

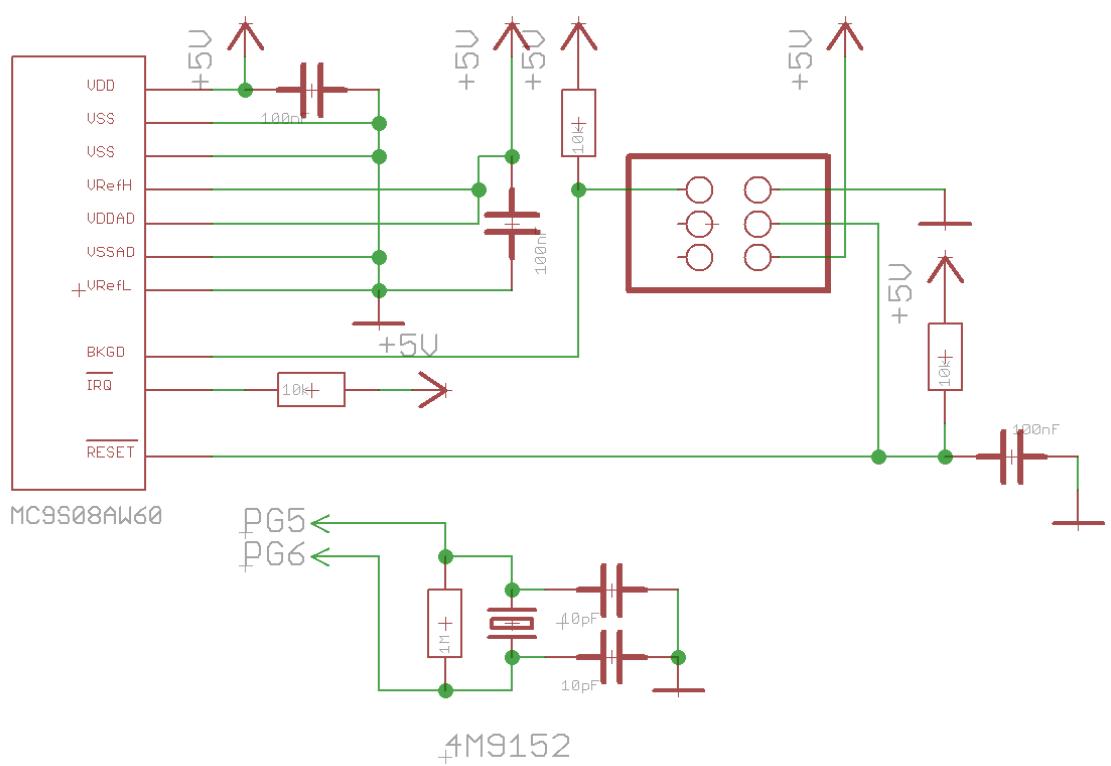


Abbildung 9: Mikrocontroller Beschaltung

Die Beschaltung, die für das korrekte Funktionieren des verwendeten Controllers benötigt wird, ist hier abgebildet und schließt die Stromversorgung und die Quarz-Oszillatorschaltung und ebenso

den Anschluss für das Programmiergerät ein.

Für dieses Projekt wurde der Mikrocontroller Motorola MC9S08AW32 zusammen mit einem 4,9252MHz Schwingquarz verwendet. Der Controller hat eine ausreichende Anzahl von Ports und seine Rechenleistung ist für dieses Projekt mehr als ausreichend.

### 3.1.5) Schaltung: Lautsprecher

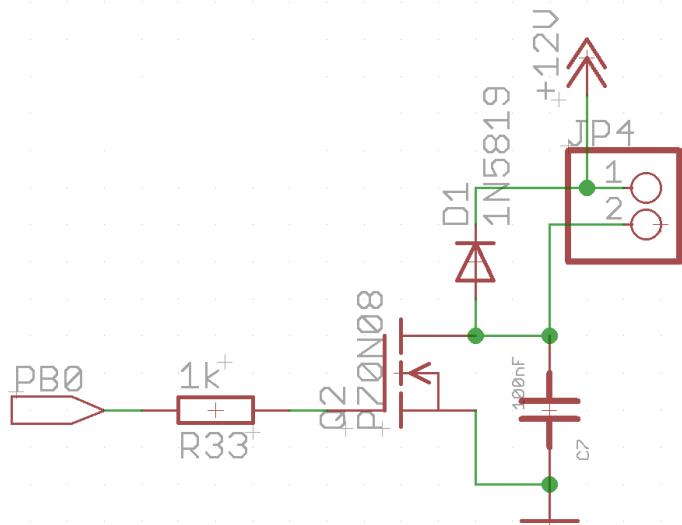
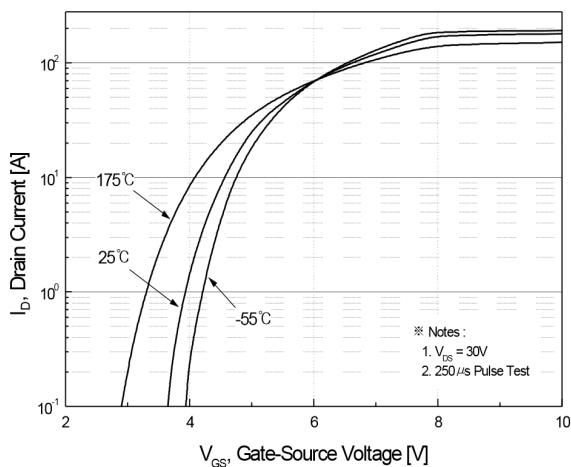


Abbildung 10: Schaltung Lautsprecher

Die Schaltung für die Ausgabe des Alarms besteht aus einem Mosfet, einem Freilaufdiode und einer Klemme mit zwei Anschlüsse für den Lautsprecher. Außerdem wurde ein Kondensator zur Abrundung des Rechtecksignals eingebaut.

Abbildung 12: P70N08



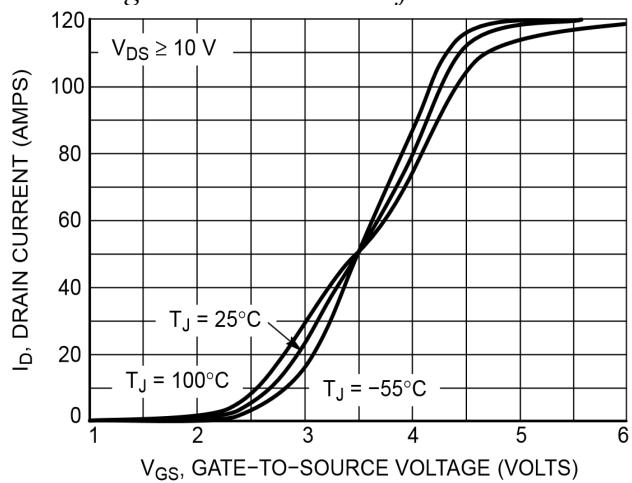
#### Transfercharakteristik

Als **Mosfet** wird der P70N08 verwendet, da dieser bereits bei Gate-Source-Spannungen von 5V beinahe vollständig durchschaltet (Logic Level). Andere Logic-Level-Mosfets, wie der P60N06, könnten ebenfalls verwendet werden, sofern sie die selbe Pinbelegung wie der P70N08 aufweisen.

Aus den Transfer-Charakteristiken ist ganz klar ersichtlich dass beide Transistoren für die Schaltung geeignet sind, da der Mikrocontroller eine Spannung von 5V am Gate anlegt. Da der P70N08 kommerziell nicht mehr erhältlich ist, wird auf der Bauteilliste der P60N06 angeführt.

Die eingezeichnete **Diode** 1N5819 ist eine Shottky-Diode welche ausgezeichnete Eigenschaften für

Abbildung 11: P60N06L Transfercharakteristik



den hörbaren Frequenzbereich aufweist.

Der verwendete **Lautsprecher** ist ein alter ausgebauter Lautsprecher mit einen Innenwiderstand von  $8\Omega$  und einer Nennleistung von 3W. Eigentlich kann aber jeder beliebige Lautsprecher verwendet werden, wenn seine Kennwerte in ungefähr dem selben Bereich liegen.

### 3.1.6) Schaltung: Taster

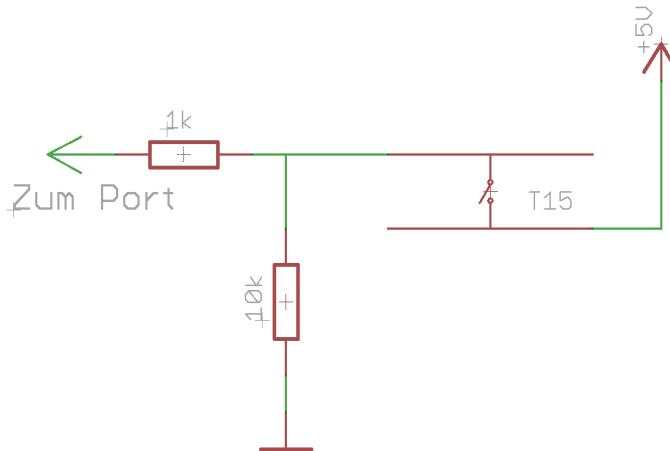


Abbildung 13: Tasterbeschaltung mit Pull Down

Sämtliche Taster auf der Platine sind Schließer, das heißt solange sie nicht gedrückt sind werden sie auf durch einen Pull-Down-Widerstand von  $10k\Omega$  auf Masse gelegt. Erst das Betätigen des Tasters schaltet den Eingang an dem der Taster hängt auf 'High'. Auf der Platine werden insgesamt 13 Taster verwendet.

### 3.1.7) Schaltung: LCD-Display

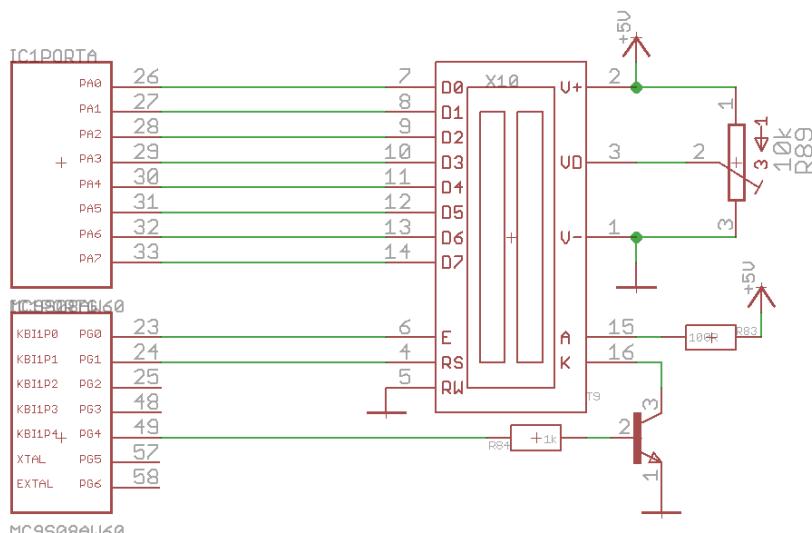


Abbildung 14: Beschaltung des LCD-Displays

Das verwendete LCD-Display MBCF162D6B ist zweizeilig und alphanumerisch und wird über acht Datenleitungen, 2 Versorgungsleitungen und 4 Steuerleitungen angesteuert. Für diese Schaltung wird der Port A des Controllers als Datenport verwendet, während die Pins G0, G1 und G4 als Steuerausgänge verwendet werden.

### 3.1.8) Schaltung: RFID-Lesegerät

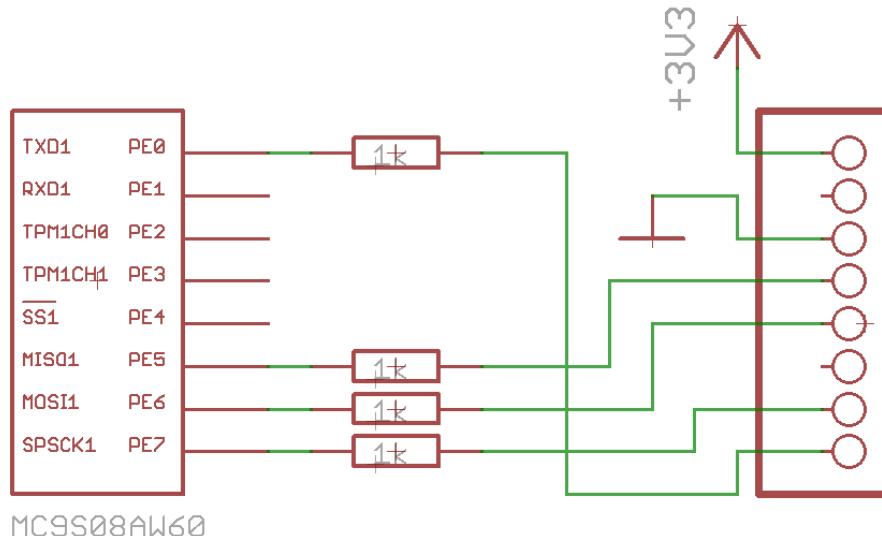


Abbildung 15: Steckleiste zum RFID-Lesegerät

Die Schaltung für das RFID-Lesegerät beschränkt sich auf eine Stiftleiste mit 8 Anschlüssen, da ein gekauftes Lesegerät darüber angeschlossen wird. Von den acht Anschlüssen werden vier zum Controller geführt, zwei dienen der Versorgung und zwei bleiben frei. Als Vorwiderstände zum Controller werden  $1\text{k}\Omega$ -Widerstände eingefügt.

#### RFID-Lesegerät

Das verwendete RFID-Lesegreät basiert auf dem RFID-Chip MFRC522 von MIFARE®. Auf der verwendeten Reader-Platine ist die Antenne bereits aufgebaut und die SPI-Schnittstelle des RFID-Chips kann direkt verwendet werden.

Das Gerät wurde gewählt da es in einem für Arduino konzipierten Kit erhältlich ist, welches neben der Reader-Platine auch noch ein RFID-Tag sowie eine RFID-Karte beinhaltet. Das ganze Kit kostet je nach Bezugsquelle 4-7€, also inklusive Lieferung um die 11€.

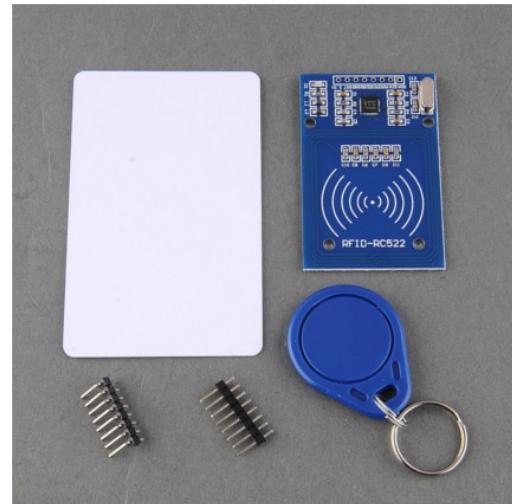


Abbildung 16: Inhalt des RFID-Kits

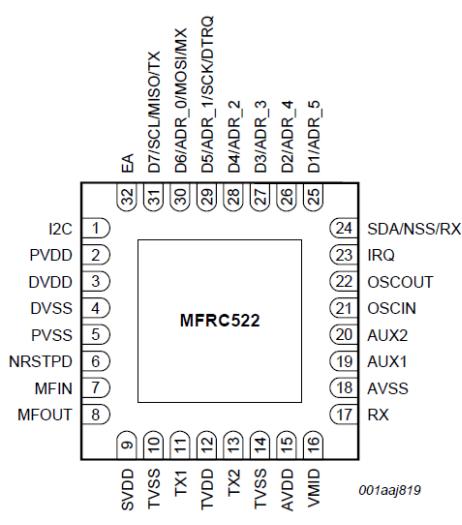


Abbildung 17: Pinbelegung MFRC522

Der verwendete Chip, der MFRC522, ist ein read/write-IC für 13,56MHz-RFID. Er benötigt eine 3,3V-Stromversorgung und ist in der Lage mit wenig externen Beschaltung eine RFID-Antenne zu betreiben. Der Chip unterstützt diverse Interfaces, nämlich I<sup>2</sup>C, UART und SPI, jedoch sind auf der gekauften Platine nur für SPI die Aus- und Eingänge an die Steckleiste geführt.

Die Vollständige Schaltung des gekauften RFID-Lesegeräts ist im Anhang beigelegt.

### 3.1.9) Schaltung: Bewegungsmelders

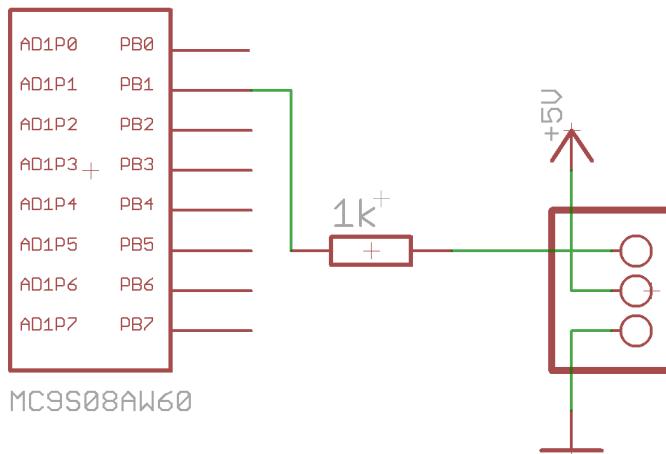


Abbildung 18: Steckleiste zum Bewegungsmelder

Die Schaltung zum Anschluss des Bewegungsmelders besteht lediglich aus einer Steckleiste mit drei Pins: einer Verbindung zum Controller für das Sensorsignal und zwei Versorgungspins. Der Bewegungsmelder ist am Pin PB1 des Controllers angeschlossen, als Vorwiderstand wird  $1\text{k}\Omega$  verwendet.

#### Bewegungsmelder

Der verwendete Bewegungsmelder ist ein Passiv-Infrarot-Bewegungsmelder vom Modell BOT-06643. Der Sensor erkennt eine menschliche Bewegung aufgrund der vom Körper ausgestrahlten Infrarotstrahlung. Er benötigt eine Versorgungsspannung von 5V und gibt bei jeder Aktivierung einen 5V-Puls mit fester Länge aus.

Das Produkt wird unter dem Namen „PIR Body Movement Sensor Module“ auf der Seite [www.geeetech.com](http://www.geeetech.com) um 8,80€ verkauft und ist, wie das RFID-Lesegerät, für Arduino konzipiert.

Das Modul hat eine Verzögerungszeit von 5 Sekunden bis zu 18 Minuten, welche sich über ein Potentiometer einstellen lässt. Bei dieser Alarmanlage ist die Verzögerungszeit auf das Minimum eingestellt.

Das Gerät sollte laut Angaben 'Low'-Pulse ausgeben, es gibt jedoch 'High'-Pulse aus. Dennoch funktionierte das Gerät bis vor kurzem einwandfrei.

Am Freitag den 09.05., eine Woche vor der Abgabe dieser Dokumentation wurde festgestellt, dass das Gerät die Funktion eingestellt hatte. Die Zeit reichte nicht mehr zur Ersetzung des Geräts weshalb in der abgegebenen Endversion die Alarmanlage ihre Basisfunktion nicht mehr wahrnehmen kann.

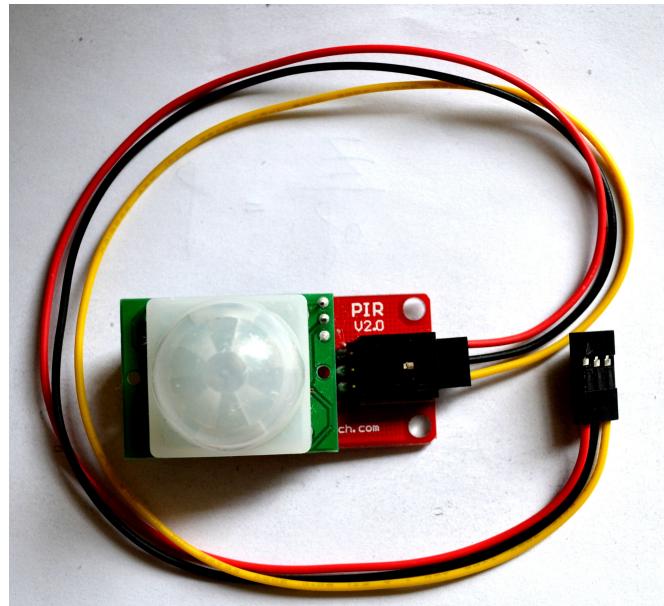


Abbildung 19: PIR-Sensormodul

### 3.1.10) Vollständige Bauteilliste

Sämtliche für die Erstellung des Projekts nötigen Bauteile können online bestellt werden. Zum Großteil sind es Standardbauteile wie Widerstände, Kondensatoren, Taster, Steckerleisten usw. Es folgt eine vollständige Auflistung sämtlicher Bauteile mit Bezugsquelle.

R100	1	Widerstand 100Ω	SMD 0805	farnell.com	distrelec.it
R1k-Traditionell	1	Widerstand 1kΩ	Traditionell	farnell.com	distrelec.it
R1k	24	Widerstand 1kΩ	SMD 0805	farnell.com	distrelec.it
R10k	16	Widerstand 10kΩ	SMD 0805	farnell.com	distrelec.it
R1M	1	Widerstand 1MΩ	SMD 0805	farnell.com	distrelec.it
Poti-10k	1	Potentiometer 10kΩ	S63X	farnell.com	distrelec.it
C10p	2	Kondensator 10pF	SMD 0805	farnell.com	distrelec.it
C100n	7	Kondensator 100nF	SMD 0805	farnell.com	distrelec.it
1N4007	1	Diode	Traditionell	farnell.com	distrelec.it
1N5819	1	Diode	Traditionell	farnell.com	distrelec.it
78S05CV	1	5V Regler	TO220	farnell.com	distrelec.it
LM1117	1	3,3V Regler	SMD	farnell.com	distrelec.it
LED rot	1	Leuchtdiode	Traditionell 3mm	farnell.com	distrelec.it
LED grün	1	Leuchtdiode	Traditionell 3mm	farnell.com	distrelec.it
FSM2JH	13	Taster	Traditionell	farnell.com	distrelec.it
1-826629-6	2	Stiftleisten 16 Pin 2,54mm	Traditionell	farnell.com	distrelec.it
BC817-40 6CW	1	NPN Transistor	SMD	farnell.com	distrelec.it
P60N06	1	MosFet	TO220	farnell.com	distrelec.it
49USMX, 4,9152MHz	1	4,9152MHz Quarz	SMD	farnell.com	distrelec.it
MC9S08AW60	1	Mikrocontroller	64-Pin QFP/LQFP	farnell.com	distrelec.it
n.a.	1	LCD-Display	n.a	Prof. De Tomaso	n.a.
BOT-06643	1	PIR-Sensor	n.a	myarduinoshop.com	geeeetech.com
SanSmart	1	RFID-Lesegerät	n.a	electrodragon.com	sainsmart.com

Die Bauteile sind großteils Standartbauteile und sind daher bei in größeren Elektronikkatalogen vorhanden. Diese Standardbauteile sind genauso bei Digikey erhältlich, wo es für Privatpersonen wesentlich einfacher zu bestellen ist. Die Besonderen Bauteile, wie der PIR-Sensor oder das RFID-Lesegerät sind im Internet auf verschiedenen Seiten erhältlich, und je nach Region sind unterschiedliche Bezugsquellen am billigsten.

Der LCD-Display für das Projekt wurde von Prof. De Tomaso bezogen und im Internet konnte keine Bezugsquelle gefunden werden.

### 3.1.11) Board

Ebenfalls mit Eagle™ 6.5.0 wurde aus dem Schaltplan das Board, also die Platine erstellt. Diese misst 4025x3300mil, also 10,2235x8,382 cm. Die Board-Datei wurde anschließend eingeschickt und maschinell gefertigt.

Eine Detailansicht des Boards mit Bestückungsliste befindet sich in den Anlagen.

#### Fehler in der ersten Version

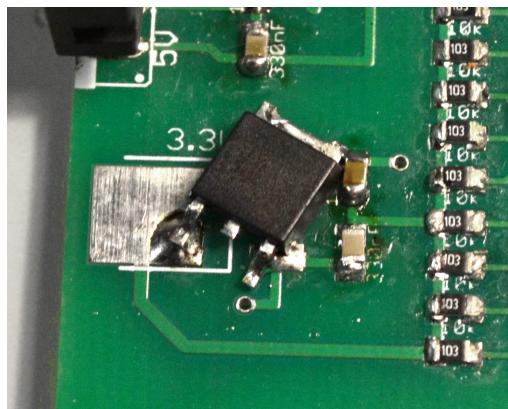


Abbildung 20: Kreativ aufgelöteter  
3,3V-Regler

Diese Version hat einige minimale Fehler, welche auf der Platine durch geschicktes Löten und Verkabeln ausgebessert wurden. Es wurde ein falsches Packpage für den 3,3V-Spannungsregler verwendet, wodurch dieser schräg aufgelötet werden musste.

Auch wurde wegen eines Fehlers in dem im Internet gefundenen Schaltplan des RFID-Lesegeräts die Stiftleiste die zum Gerät führt falsch verbunden. Daher konnte zur Verbindung mit dem Lesegerät nicht wie vorgesehen ein Flachbandkabel verwendet werden. Stattdessen musste das Lesegerät durch ein selbst erstelltes überkreuztes Kabel verbunden werden.

Außerdem wurde für den Chip-Select-Anschluss des Lesegeräts kein Anschluss auf dem Stecker vorgesehen da beabsichtigt wurde diesen direkt auf 'Low' zu legen (Masse). Später stellte sich heraus dass der Chip-Select-Pin zwischen jeder Datenübertragung auf 'High' gesetzt werden muss. Zum Glück konnte jedoch einer der sicherheitshalber zusätzlich ausgeführten Ausgänge des Mikrocontrollers verwendet werden.

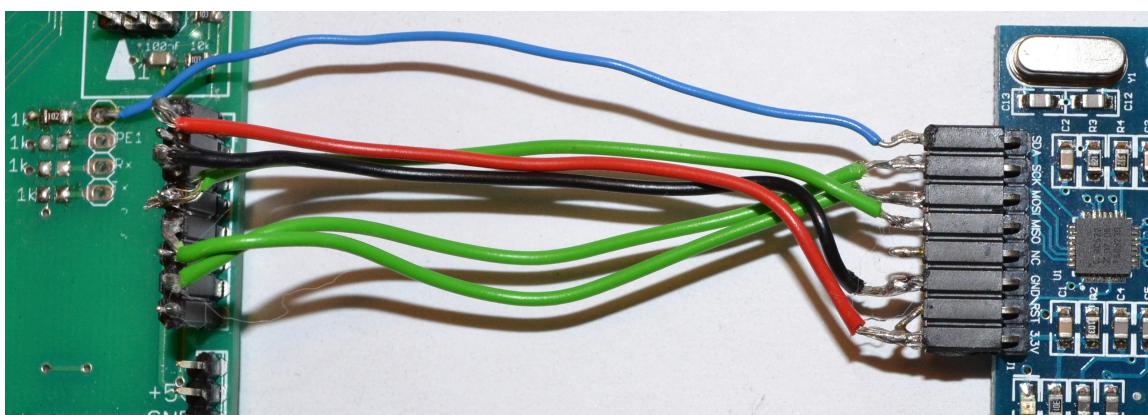


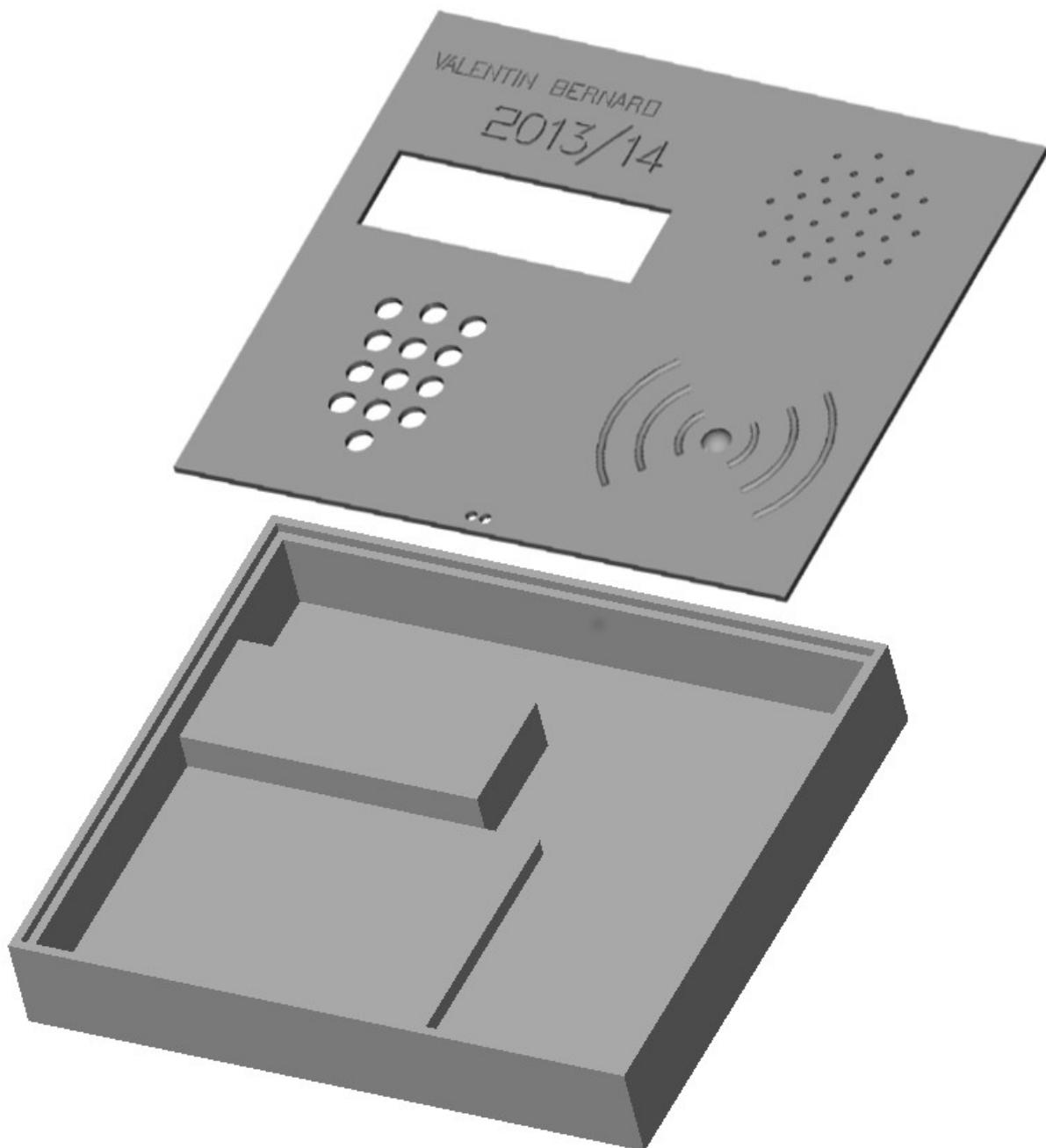
Abbildung 21: Verkabelung und Zusatzleitung für Chip-Select (Blau)

## **Board Version zwei**

Aufgrund des geringen Einflusses der Fehler auf die Funktionstüchtigkeit des Projekts wurde entschieden, keine zweite Version der Platine zu fertigen. In einer zweiten Version hätten neben der Behebung der beschriebenen Fehler die Schaltung um ein GSM- oder ein Funkmodul erweitert werden können, um eine der im ersten Entwurf vorgesehenen Zusatzfunktionen zu realisieren. Aus Zeitmangel wurde darauf verzichtet.

### **3.1.12) Gehäuse**

Das Gehäuse der Alarmanlage wurde mit der CAD-Software VariCAD™ gezeichnet und mit dem schuleigenen 3D-Drucker gefertigt. Innerhalb der Hülle befinden sich die Platine, der Lautsprecher und das RFID-Lesegreät. Der Bewegungsmelder wird zusätzlich angeschlossen.



*Abbildung 22: 3D-Zeichnung der Hülle*

## 3.2) Software

Die gesamte Software für die Anlage wurde in der Programmiersprache Assembler geschrieben. Dazu wurde die Programmierumgebung Codewarrior™ von Freescale verwendet. Natürlich wäre eine Programmierung in C ebenso möglich gewesen.

### 3.2.1) Grundstruktur

Das Programm besteht aus einer Schrittkette mit drei Schritten: „Inaktiv“, „Aktiv“ und „Alarm“. Zusätzlich kann vom Schritt „Inaktiv“ in den Schritt „Change\_PW“ gesprungen werden, um das Passwort zu ändern.

Im Schritt „Inaktiv“ kann die Schaltung per Knopfdruck aktiviert werden. Außerdem kommt man durch drücken der Vor-Taste in den Schritt „Change\_PW“. Solange die Anlage inaktiv ist blinkt die grüne LED.

Sobald die Anlage in den Schritt „Aktiv“ wechselt, wird ununterbrochen der Bewegungsmelder ausgemessen, um bei eventueller Auslösung desselben in den Schritt „Alarm“ zu springen. Außerdem kann man durch eine Passworteingabe oder durch das Annähern des RFID-Tags in den Schritt „Inaktiv“ zurückkehren. Im aktiven Zustand blinkt die rote LED.

In letzten Schritt, „Alarm“, wird über den eingebauten Lautsprecher der Alarm ausgegeben. Auch hier kann über das Passwort oder das RFD-Tag zu „Inaktiv“ zurückgesprungen werden. Zusätzlich kehrt die Anlage automatisch in den Schritt „Aktiv“ zurück wenn der Bewegungsmelder 5 Minuten lang kein Signal gibt. Die rote LED leuchtet bei ausgelöstem Alarm durchgehend.

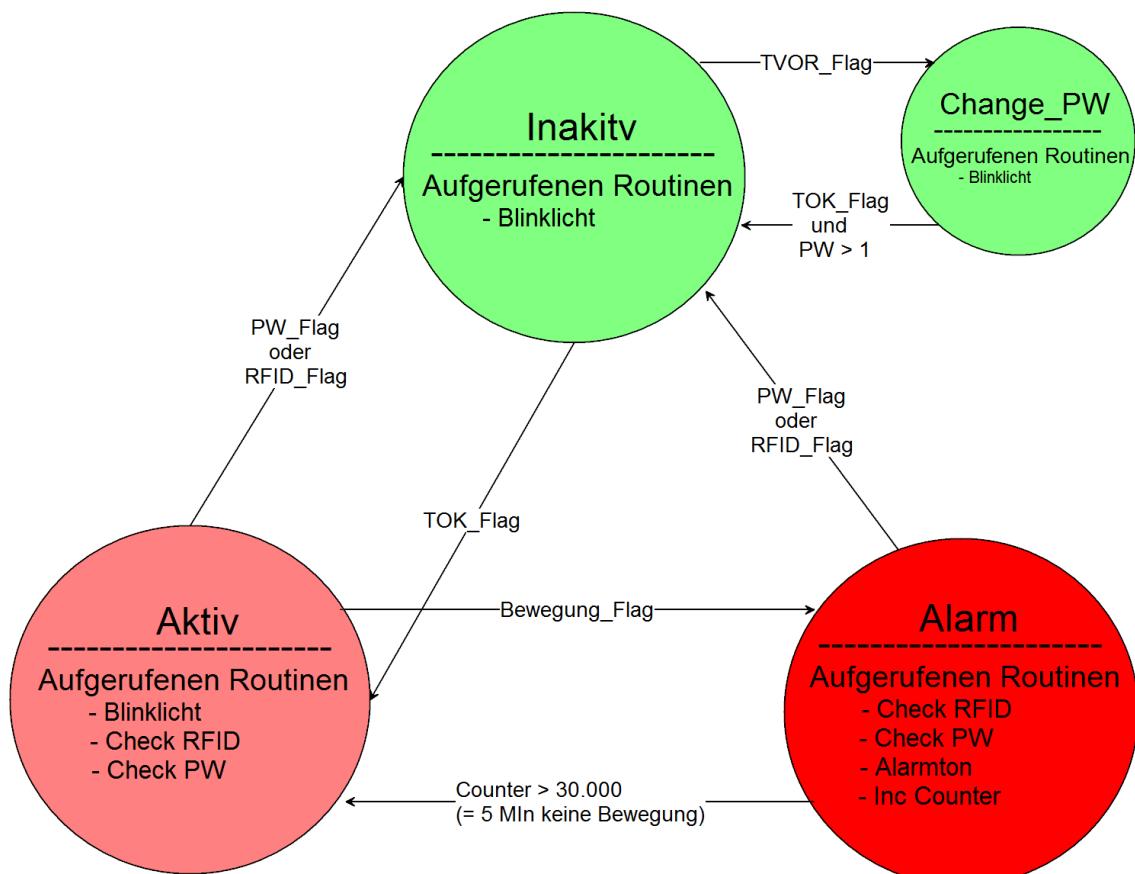


Abbildung 23: Schrittkette

### **3.2.2) Bewegungsmelder**

Die Routine zum Auslesen des Bewegungsmelders wird in den Schritten „Aktiv“ und „Alarm“ aufgerufen. Sie wertet die vom Sensor ausgegebenen Pulse aus und gibt durch Setzen von drei Flags bekannt, wenn eine Bewegung erfasst wurde.

Damit der Alarm nicht durch einen eventuell fehlerhaft ausgegebenen Puls des Bewegungsmelders ausgelöst wird, ist es zur Aktivierung des Alarms nötig, dass mindestens 3 Pulse in Folge auftreten. Dabei wird bei jedem hereinkommenden Puls eines der drei Flags gesetzt. Wenn alle drei Flags gesetzt sind, löst die Software den Alarm aus. Vergeht zu viel Zeit zwischen zwei Pulsen werden alle Flags wieder gelöscht.

### **3.2.3) Tonausgabe**

Der Alarmton wird über einen Rechteckimpuls ausgegeben, welcher einen Mosfet durchschaltet. Die Frequenz wird von einem Timer-Interrupt festgelegt, in dem die Wartezeit (in Clock-zyklen) bis zum nächsten Interrupt festgelegt werden kann. Bei gegebener Clockfrequenz von 4,9152 MHz lässt sich die Frequenz des Rechteckimpulses aus der eingegebenen Zahl der Warte-Zyklen folgendermaßen berechnen:

$$f = \frac{f_c}{2 * \text{Zahl}} = \frac{4,9152}{2 * \text{Zahl}}$$

Daher ergibt sich die Zahl, welche man zum Erreichen einer bestimmten Frequenz im Timerinterrupt als Wartezeit eingeben muss, aus der Formel

$$\text{Zahl} = \frac{f_c}{2 * f} = \frac{4,9152}{2 * f}$$

Der Faktor zwei im Nenner kommt daher, dass das Programm im Timer-Interrupt dem eines Blinklichts entspricht, d.h. bei jedem Interrupt wird das Output-Pin abwechselnd ein- und ausgeschaltet. Die Frequenz des Rechtecks ist damit halb so groß wie die Interrupt-Frequenz.

```
;Programm im Interrupt
    brset 0,PORTB,Aus
    bset  0,PORTB
    bra   Ein
Aus
    bclr 0,PORTB
Ein
    pulh
    rti
```

Die Software zur Tonausgabe arbeitet in einem einstellbaren Rhythmus fortlaufend eine Look-up-Table mit Zahlen ab, welche vom Timerinterrupt als Wartezeiten verwendet werden. Als Alarm werden lediglich zwei Töne abwechselnd ausgegeben, doch das selbe Programm kann auch zur Ausgabe von einfachen Melodien verwendet werden.

(Flussdiagramm?)

### **3.2.4) Passwort**

Das Programm für die Passwort-Eingabe kann in den Schritten „Aktiv“ und „Ausgelöst“ durch die Taste „OK“ aufgerufen werden. Auf dem LCD-Bildschirm wird bei Eingabe einer Zahl ein Sternchen ausgegeben. Mit einem weiteren klick auf OK bestätigt der Nutzer die Vollendung der Eingabe, worauf entweder in den Schritt „Inaktiv“ zurückgesprungen wird, oder für einige Sekunden der Textzug „FALSCH EINGABE“ erscheint.

Die Passwort-Überprüfung geht softwaremäßig folgendermaßen von statthen: zehn in der E-RAM aufeinander folgende Variablen beinhalten das Passwort. Dabei beinhalten beispielsweise bei einem Passwort mit sechs Stellen die ersten sechs Variablen je eine Zahl des Passwortes, während die restlichen vier \$FF enthalten. Bei jeder Zahleneingabe überprüft das Programm die Übereinstimmung der Eingabe mit den Variablen an der jeweiligen Stelle. Sobald ein Fehler festgestellt wird, wird ein Fehler-Flag gesetzt. Beendet der Nutzer durch das OK die Eingabe wird nun überprüft, ob die nächste Variable \$FF enthält. Tut sie dies nicht, war die Passworteingabe zu kurz und es wird ebenfalls das Fehler-Flag gesetzt. Ist an dieser Stelle im Programm nun kein Fehler-Flag gesetzt, bedeutet das, dass die Eingabe korrekt war. Die Anlage springt dann in den inaktiven Zustand.

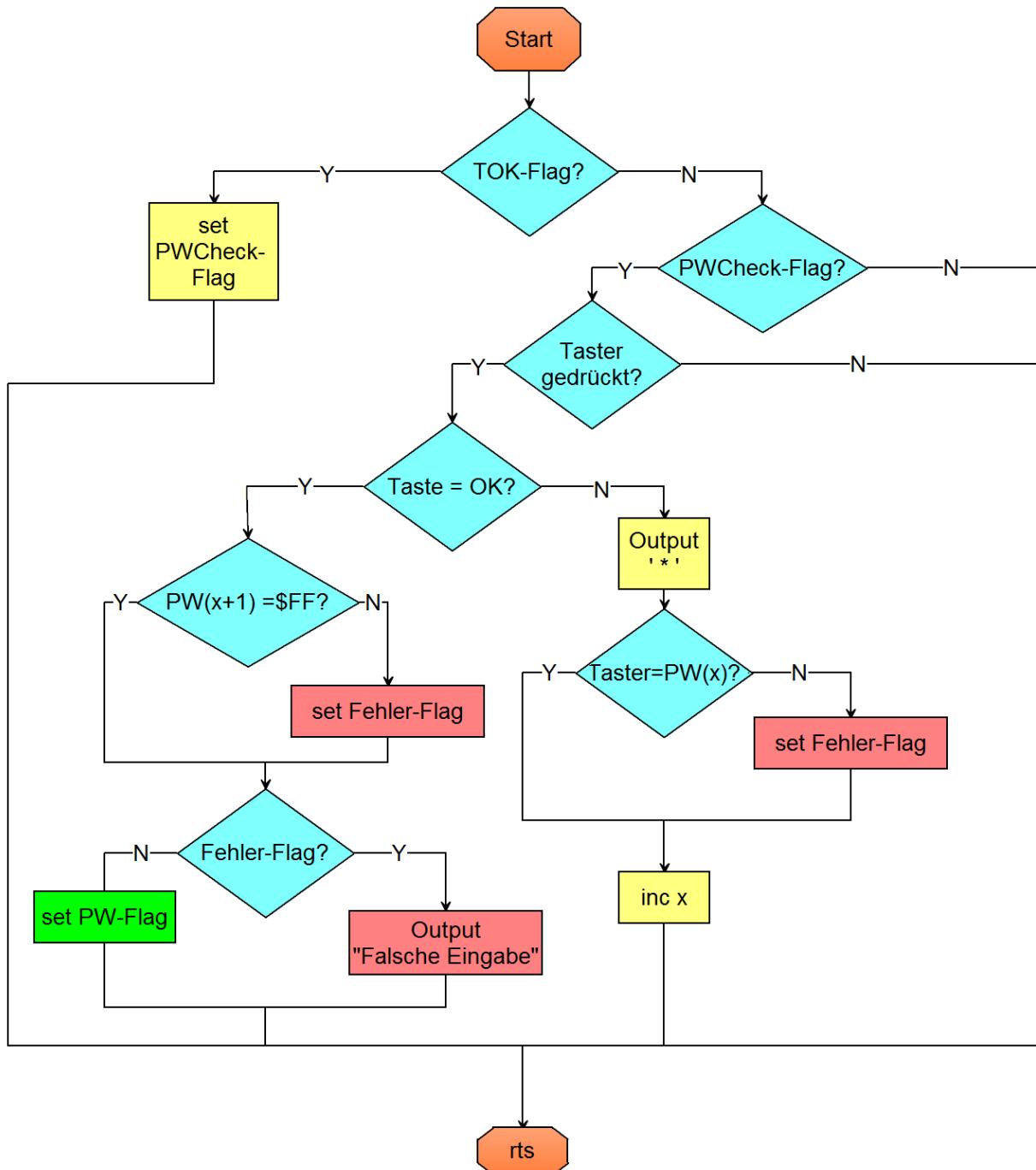


Abbildung 24: Flussdiagramm Passworterkennung

## Ändern des Passwortes

Zum Ändern des Passwortes muss im „Inaktiv“-Schritt die Vorwärts-Taste gedrückt werden. Die Schrittkette springt daraufhin in den „Passwort ändern“-Schritt, den der Nutzer erst verlassen kann wenn er mindestens ein Passwort der Länge '1' eingegeben hat. Mit OK bestätigt er die Eingabe und kehrt in den „Inaktiv“-Schritt zurück.

Die Passwort-Änderung funktioniert ähnlich wie die Passwort-Kontrolle, nur dass bei jeder Eingabe des Benutzers nicht der eingegebene Wert mit der Variablen verglichen wird, sondern die Variable mit dem eingegebenen Wert überschrieben wird. Beim Klick auf das OK werden die restlichen Variablen mit \$FF überschrieben. Damit kann das neue Passwort verwendet werden.

### 3.2.5) RFID

Die Erkennung eines RFID-Tags erfolgt über das angeschlossene RFID-Lesegerät, welches mit dem Mikrocontroller über SPI kommuniziert. Dabei wird in den Schritten „Aktiv“ und „Alarm“ an das Gerät eine Anfrage gesendet um zu erfahren ob sich ein RFID-Tag im Bereich der Antenne befindet. Ist dies der Fall, wird die Anlage auf „Inaktiv“ gesetzt.

#### SPI-Initialisierung

Die SPI-Schnittstelle wird als Master bei XYZ kHz konfiguriert, die Übertragung erfolgt Active-High und das Most-Significant-Byte wird als erstes gesendet.

```
;SPI-Initialisierung
Init_SPI
    mov    #01011100,SPI1C1    ; SPI Aktiviert als Master
;           | | | | | |
;           | | | | | |----- Shifter direction. 0 = 7..0
;           | | | | | |----- SS Output enable = 0
;           | | | | | |----- Clock Pahse, 0 = Middle o. Cycle
;           | | | | | |----- Clock Polarity 0 = Active High
;           | | | | |----- MSTR, 1 = Master
;           | | | | |----- Transmit Interrupt enable
;           | | | | |----- SPI enable
;           | | | | |----- Interrupt enable

    mov    #00110000,SPI1BR    ; Stellt den SPI_Clock ein
;           | | | | | |
;           | | | | | |----- )
;           | | | | | |----- > Timer Scaler 2..256 (2)
;           | | | | | |----- )
;           | | | | |----- n.c
;           | | | | |----- )
;           | | | | |----- > Timer Prescaler 1..8
;           | | | | |----- )
;           | | | | |----- n.c

    rts
```

## SPI-Kommunikation

Die gesamte Kommunikation mit dem RFID-Chip läuft über zwei Software Routinen ab: eine Schreib- und eine Leseroutine.

Die Lese-Routine benötigt lediglich eine Adresse, welche sie entsprechend dem Datenblatt des RFID-Chips in das Lesen-Format umwandelt. Die umgewandelte Adresse wird an den Chip gesandt, worauf dieser den Wert an der gesendeten Adresse zurückschickt.

```
;RFID-Leseroutine

write_RFID

    brclr    5,SPI1S,Ende_RW
    lda      Adresse
    lsla
    AND      #%01111110
    sta      SPI1D
    bclr    3,PORTE

S1Delay
    brclr    7,SPI1S,S1Delay
    lda      SPI1D
    lda      Daten
    sta      SPI1D

S2Delay
    brclr    7,SPI1S,S2Delay
    lda      SPI1D
    bset    3,PORTE

Ende_RW
    rts
```

Die Schreibroutine hingegen benötigt eine Adresse und ein Datenbyte. Wieder wird die Adresse laut Datenblatt umgewandelt, diesmal in das Schreiben-Format. Die Adresse wird als erstes gesendet, anschließend folgt das Datenbyte, worauf im RFID-Chip der Wert an der Adresse mit dem Datenbyte überschrieben wird.

```
;RFID-Schreibroutine

Read_RFID

    brclr    5,SPI1S,Ende_RW
    lda      Adresse
    lsla
    AND      #%01111110
    ORA      #%10000000
    sta      SPI1D
    bclr    3,PORTE

S3Delay
    brclr    7,SPI1S,S3Delay
    lda      SPI1D
    lda      #$00
    sta      SPI1D
```

```

S4Delay
brclr    7, SPI1S, S4Delay

lda      SPI1D

sta      Daten
bset    3, PORTE

rts

```

Diese beiden Softwareteile, die das Kernstück der Kommunikation bilden, funktionieren einwandfrei. Jedoch ist es bisher nicht möglich vom RFID-Lesegerät irgendwelche Daten zu erhalten die auf die Erkennung des RFID-Tags hinweisen.

### **RFID-Request**

Zur Erkennung eines Tags wird  $26_H$  in den FOFO-Buffer des RFID-Lesegeräts gespeichert und daraufhin der Befehl PCD-TRANCEIVE in das Command-Register geschrieben. Daraufhin sendet der RFID-Chip den Befehl  $26_H$  über die Antenne. Dieser Befehl bringt sämtliche RFID-Tags nach dem Standard ISO/IEC 14443 dazu, eine Antwort zurückzusenden.

Da der RFID-Chip im TRANCEIVE-Modus ist erwartet er nun am Tx-Eingang eben diese Antwort. Sobald der Empfang der Antwort abgeschlossen ist wird, falls diese gültig ist, im CommIrq-Register das Tx-Bit gesetzt. Ist die Antwort hingegen „IDLE“ (aktiv), wird das IdleIrq-Bit gesetzt. Kommt gar keine Antwort, läuft ein Timer ab und das TimerIrq-Bit wird gesetzt.

Im Controller wird, sobald der TRANCEIVE-Befehl gesendet wurde, wird durchgehend das CommIrq-Register ausgelesen. Ist eines der beiden Bits welche auf ein Tag hinweisen auf 'High', wird entweder das RFID-Flag auf  $FF_H$  gesetzt um eine erfolgreiche Erkennung eines Tags zu kennzeichnen, oder die Übertragung wird wiederholt. Ist hingegen das TimerIrq-Bit gesetzt wird die Übertragung neu gestartet. Zur erneuten Übertragung werden die Interrupt-Flags gelöscht und dem Chip der Befehl IDLE übermittelt, bevor erneut der TRANCEIVE-Befehl gesendet wird. Die Tag-Erkennung geschieht fortlaufend in den Schritten „Aktiv“ und „Alarm“.

In dieser Routine liegt möglicherweise der Fehler, aufgrund dessen die Deaktivierung per RFID-Tag nicht funktioniert. Aus einem noch ungeklärtem Grund wir im CommIrq-Register immer das TimerIrq-Bit gesetzt bevor eines der Flags gesetzt wird welches die Anwesenheit eines Tags melden. Das heißt, es wird keine gültige Antwort vom RFID-Tag empfangen. Das Problem könnte genauso beim angekauften Gerät liegen, da es nicht unbedingt ein Gerät der höchsten Preisklasse ist und deshalb nicht vor Fehlfunktionen gefeilt ist. Zur Bestätigung dieser Theorie müsste ein zweites Gerät angekauft werden.

## 4) Kostenberechnung

### Bauteilkosten

Bauteil	Anz.	Beschreibung	Package	Bezugsquelle 1	Einzelpreis	Prototyp	1000er-Serie	Preis	Bezugsquelle 2
R100	1	Widerstand 100Ω	SMD 0805	farnell.com	0,041 €	0,04 €	21,00 €	0,021 €	distrelec.it
R11k-Traditionell	1	Widerstand 1kΩ	Traditionell	farnell.com	0,019 €	0,02 €	8,00 €	0,008 €	distrelec.it
R1k	24	Widerstand 1kΩ	SMD 0805	farnell.com	0,041 €	0,98 €	456,00 €	0,019 €	distrelec.it
R10k	16	Widerstand 10kΩ	SMD 0805	farnell.com	0,056 €	0,90 €	272,00 €	0,017 €	distrelec.it
R1M	1	Widerstand 1MΩ	SMD 0805	farnell.com	0,044 €	0,04 €	13,00 €	0,013 €	distrelec.it
Poti-10k	1	Potentiometer 10kΩ	S63X	farnell.com	1,190 €	1,19 €	510,00 €	0,510 €	distrelec.it
C10p	2	Kondensator 10pF	SMD 0805	farnell.com	0,041 €	0,08 €	64,00 €	0,032 €	distrelec.it
C100n	7	Kondensator 100nF	SMD 0805	farnell.com	0,100 €	0,70 €	350,00 €	0,050 €	distrelec.it
1N4007	1	Diode	Traditionell	farnell.com	0,188 €	0,19 €	100,00 €	0,100 €	distrelec.it
1N5819	1	Diode	Traditionell	farnell.com	0,141 €	0,14 €	101,00 €	0,101 €	distrelec.it
78S05CV	1	15V Regler	TO220	farnell.com	0,760 €	0,76 €	290,00 €	0,290 €	distrelec.it
LM1117	1	3,3V Regler	SMD	farnell.com	1,120 €	1,12 €	600,00 €	0,600 €	distrelec.it
LED rot	1	Leuchtdiode	Traditionell 3mm	farnell.com	0,310 €	0,31 €	158,00 €	0,158 €	distrelec.it
LED grün	1	Leuchtdiode	Traditionell 3mm	farnell.com	1,310 €	1,31 €	1.158,00 €	1,158 €	distrelec.it
FSM2JH	13	Taster	Traditionell	farnell.com	0,166 €	2,16 €	728,00 €	0,056 €	distrelec.it
1-826629-6	2	Stiftleisten 16 Pin 2,54mm	Traditionell	farnell.com	1,550 €	3,10 €	2.040,00 €	1,02 €	distrelec.it
BC817-40 6CW	1	NPN Transistor	SMD	farnell.com	0,189 €	0,19 €	34,00 €	0,034 €	distrelec.it
P60N06	1	MosFet	TO220	farnell.com	2,310 €	2,31 €	719,00 €	0,719 €	distrelec.it
49UJSMX_4.9152MHz	1	4.9152MHz Quarz	SMD	farnell.com	2,510 €	2,51 €	1.906,25 €	1,220 €	distrelec.it
MC9S08AW60	1	Mikrocontroller	64-Pin QFP/LQFP	farnell.com	4,450 €	4,45 €	2.750,00 €	2,750 €	distrelec.it
n.a.	1	LCD-Display	n.a.	Prof. Detomasso	3,500 €	3,50 €	3.500,00 €	3.500 €	n.a.
BOT-06643	1	PIR-Sensor	n.a.	myarduinoShop.com	7,200 €	7,20 €	7.200,00 €	7.200 €	geetech.com
SanSmart	1	RFID-Lesegerät	n.a.	electrodragon.com	7,500 €	7,50 €	7.500,00 €	7.500 €	sainsmart.com

### Platinenkosten

Platinenfläche in cm <sup>2</sup>	85,693377
Fixkosten	200,00 €
Kosten/cm <sup>2</sup> Prototyp	0,04 €
Gesamtkosten Prototyp	163 €
Kosten/cm <sup>2</sup> 1000 Stück	203,43 €
Gesamtkosten 1000 Stück	0,02 €
Stückkosten bei 1000	1.913,87 €
	1,91 €

### Entwicklungs kosten

Arbeitsstunden	163
Stundenlohn	25,00 €
Lohnkosten	4.075,00 €
	153

### Amortisierung

Verkaufspreis	60,00 €
gesamte Stückkosten	31,706 €
Gewinn	28,29 €
Amortisierung ab (Stück):	153

Die Kostenbrechung ergibt, dass bei einem Verkaufspreis von 60€ pro Anlage, die Entwicklungskosten ab 153 Verkauften Anlagen amortisiert sind. Nachdem hauptsächlich Privatanwender als Kunden in Frage kommen ist eine solche Verkaufszahl durchaus realistisch.

## 4.1) Notwendige Verbesserungen und mögliche Erweiterungen

Damit das Produkt einen Marktpreis von 60€ erzielen kann, was für eine Alarmanlage ein absolut akzeptabler Preis ist, sind einige Verbesserungen notwendig. In erster Linie müsste die RFID-Deaktivierung realisiert werden. Mögliche Schritte zu dessen Realisierung wären der Ankauf eines weiteren Exemplars des Lesegeräts oder die Verwendung eines völlig anderen Lesegerät. Mit einem geringen Aufwand an Finanziellen Mitteln wäre das Problem innerhalb von maximal zwei vollständigen Arbeitswochen lösbar.

Zudem müssten einige kleinere Erweiterungen an Soft-und Hardware vorgenommen werden, welche die Bedienbarkeit des Geräts erhöhen und die Nutzung des Geräts als sichere Alarmanlage ermöglichen. Dazu zählen die Integrierung der Stromversorgung in das Gerät, vielleicht durch ein integriertes Schaltnetzteil im inneren der Hülle und eine Wandmontage auf einer Anschlussdose. Softwaretechnisch müsste einerseits zur Änderung des Passworts eine Passworteingabe erforderlich sein, andererseits müsste das Passwort permanent gespeichert werden und möglicherweise über einen für jedes Gerät einzigartigen Sicherheitscode rücksetzbar sein.

Mögliche Erweiterung schließen natürlich jene Erweiterungen ein, welche von Anfang an geplant waren, also eine Deaktivierung und Aktivierung über Funk und ein Alarm über SMS. Außerdem könnte man die Anlage um einen Gassensor oder einen Rauchmelder erweitern, um dem Gerät zusätzliche Funktionalität zu geben.

## **5) Anlagen**

*Schaltpläne, Board, Bestückungsliste*

*Digitale Anlagen*