

Elektroniker für Informations- und Systemtechnik

RFID Zeiterfassungsterminal



Inhaltsverzeichnis

1. Auftragsanalyse.....	3
2. Arbeitsplan/ Arbeitsplan Tabelle.....	4-5
3. Struktogramm/ Blockschaltbild.....	6-7
4. Stromlaufplan/Layout/ Materiallisten.....	8-10
5. Test Ergebnisse und Lösungen.....	11
6. Inbetriebnahmeprotokoll.....	12
7. Produktbeschreibung/ Bedienungsanleitung.....	13-19
8. Übergabeprotokoll.....	20
9. Einweisungsprotokoll.....	21
10. Anlagen.....	22

1. Auftragsanalyse

1.1 In diesem Projekt ist zugeordnete Aufgabe einen RFID-Zeiterfassungsterminal mit Datenspeicher und Real Time Clock zu fertigen.

Folgende Spezifikationen werden vorgegeben:

- Controllertyp: Arduino Uno
- RFID-Modul MFRC522
- 16x4 LCD zur Bedinungsanzeige
- 3 Taster mit folgenden Funktionen: „Kommen“, „Gehen“, „Info“
- Die Arbeitsdaten sollen im Exel Format auf einer SD-Karte gespeichert werden
- Einsatz eines frei wählbaren RTC Moduls zur genauen Zeiterfassung
- Das Gerät soll in einem Gehäuse verbaut werden welches noch nicht spezifiziert ist
- 4 voreingestellte Benutzerprofile mit hinterlegter ID des zugehörigen RFID-Chips
- Spannungsversorgung über ein 9V Netzteil

1.2 Spezielle Kundenanforderungen

- Schutz gegen wiederholte Anmeldung
- SD- Karte Check.
- Mögliche Erweiterungen für „Info“.(Beschreiben)
- Sommerzeit/Winterzeit Lösung finden
- Markierung und Info für Personen die vergessen haben sich abzumelden

1.3 Soll-Zustand bei Projektabgabe

Als fertiges Produkt bekommen wir einen RFID-Zeiterfassungsterminal für bis zu 10 Karten mit Datenspeicher auf SD-Karte und RTC (Real Time Clock).

1. Arbeitsplan

- Arbeitsumfang und Auftragsziel analysieren
- Spezielle Kundenanforderungen klären
- Werkzeug und Material auswählen und beschaffen
- Stromlaufplan erstellen
- Layout-plan erstellen
- Programme entwickeln und Programmdokumentation erstellen
- Mechanische Fertigung/Gehäuse fertigen
- Platine bestücken
- Programme in Systeme einbinden, Kompatibilitätsprobleme analysieren und Lösungen entwickeln
- Teilsysteme in Gesamtsysteme integrieren
- Systemtests durchführen
- Signale an Schnittstellen prüfen, netzwerkspezifische Prüfungen durchführen
- Funktionsprüfung durchführen
- Dokumentation erstellen
- Übergabe an den Kunden durchführen
- Prüfprotokoll ausfüllen
- Übergabeprotokoll erstellen

Um Projekt zu realisieren, haben wir folgende Teile vorbereitet:

1. Teile, die man mechanisch bearbeiten soll (Gehäuse).
2. Elektronische Teile (Widerstände, Lichtdiode, Tasten, Module).
3. Spannungsversorgung : Netzteil 230V AC/9V DC.

Projekt-Arbeit besteht aus vier Teilen.

1. Mechanische Fertigung

- a) Messungen und Zeichnungen
- b) Die Bohrungen anreißen, bohren.

1. Elektronische Fertigung.

- a) Teile anpassen und löten
- b) Messungen (ohne Spannung und mit Spannung).
- c) Funktions-Prüfung
- d) Das fertige Gerät in das Gehäuse montieren.

3. Quellcode erstellen (C++)/Gerät testen

4. Dokumentation.

- a) Beschreibungen
- b) Zusammenfassung der Projektdokumentation

2.1 Arbeitsplan Tabelle:

	Arbeitsart	Soll Zeit	Ist Zeit
1	Mechanische Fertigung		
	1.Messungen und Zeichnungen	0,1	0,1
	2.Die Bohrungen anreißen, bohren.	0,5	0,5
2	Elektronische Fertigung		
	1.Platine bestücken	0,4	0,4
	2.Teile anpassen und löten	0,8	0,8
	3.Messungen (Signalerfassung).	1,5	1,5
	4.Funktions-Prüfung	0,4	0,4
	5.Das fertige Gerät in das Gehäuse montieren	0,8	0,8
3	Quellcode erstellen (C++)/Gerät testen		
	1.Programme entwickeln, prüfen und Fehler beheben	7,3	7,3
4	Dokumentation		
	1.Stromlaufplan erstellen/Layout-plan erstellen	1,3	1,3
	2.Programmdokumentation erstellen	0,7	0,7
	3.Beschreibungen/Auftragsziel analysieren	4,4	4,4
	4.Zusammenfassung der Projektdokumentation	1,6	1,6
	5.Übergabe und Protokollierung	0,2	0,2
	Gesamte Zeit in Stunden und Minuten	20	20

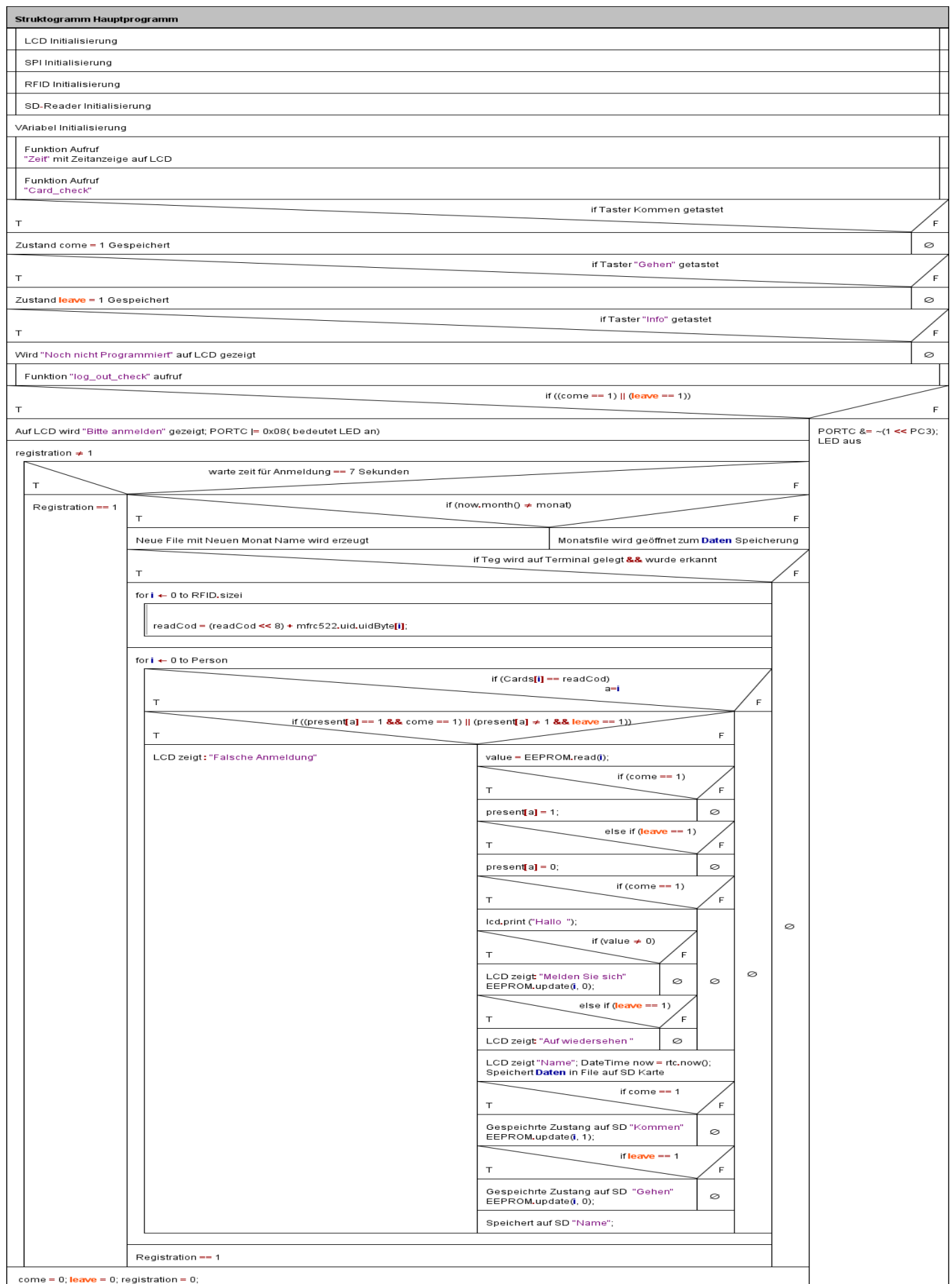
Für Mechanische Fertigung werden folgende Hilfsmittel benötigt

- 1.Schleifpapier grob und fein.
- 2.Bohrmaschine mit Bohrer
- 3.Universalmesser

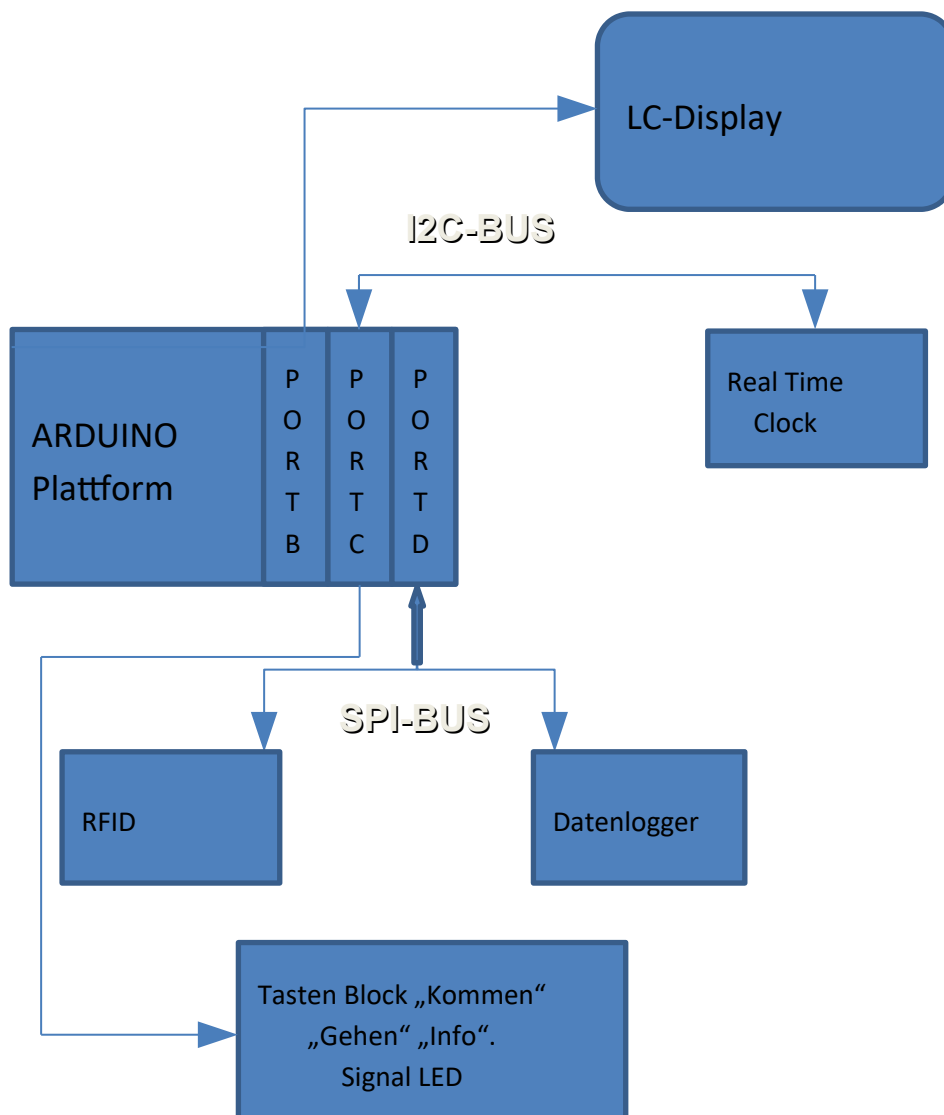
Für Elektronische Fertigung werden folgende Hilfsmittel benötigt.

- 1.Lötkolben-Set, Platinen Halter, Lötdraht, Feilkloben, Messer
- 2.Multimeter, Netzteil, Oszilloskop.
- 3.Schraubenzieher

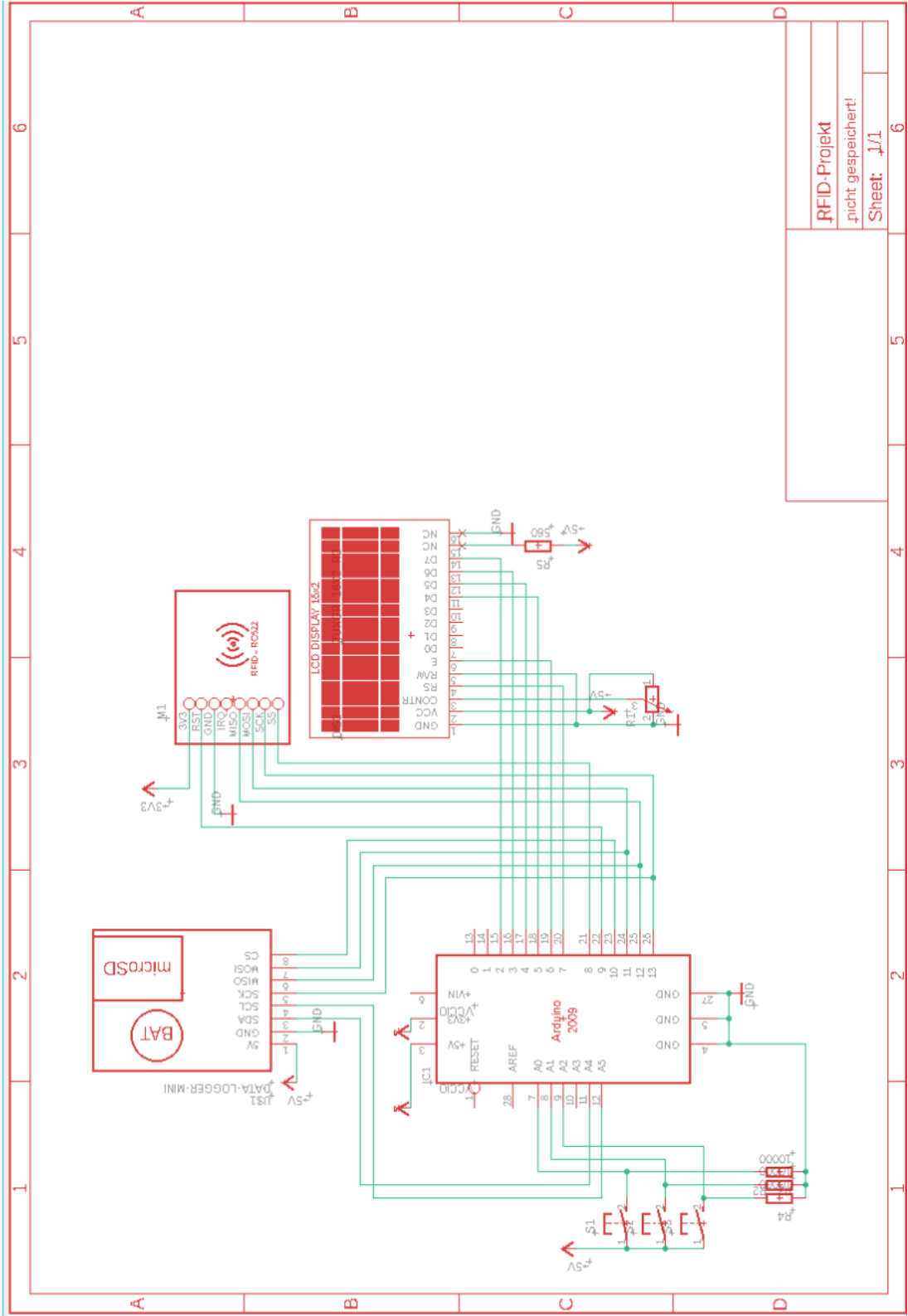
3.1 Struktogramm



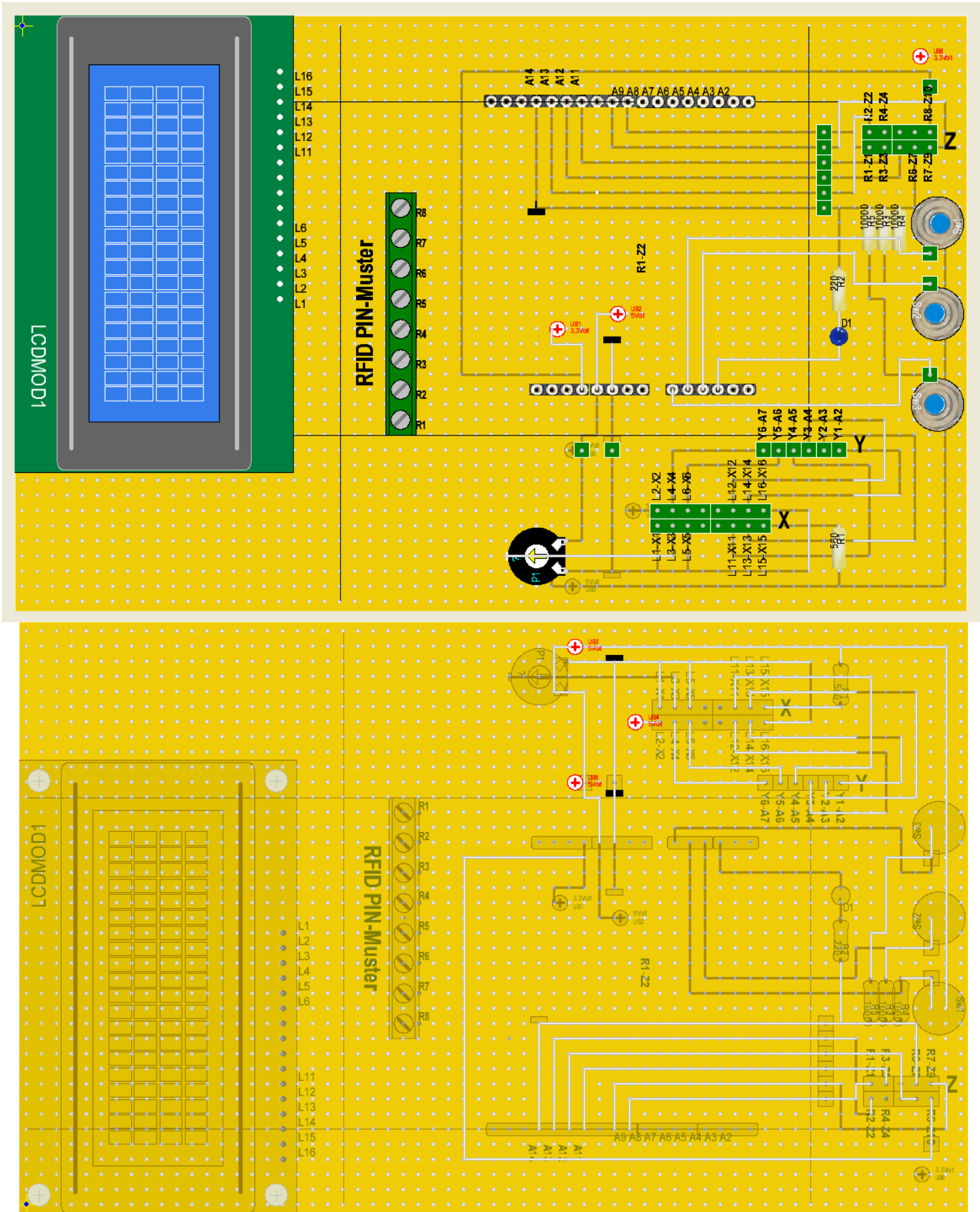
3.2 Blockschaltbild



4. Stromlaufplan



4.1 Layout



4.2 Materiallisten

Nº	Name	Hersteller	Menge	Preis(ohne MwSt.)	Gesamt Price In €
1.	Kunststoffgehäuse Basteln Projekt Anschlussdose ABS Pult Tisch Gehäuse Elektronik		1		
2.	RFID-Modul Set		1		
3.	Erweiterungsmodul Datenlogger Modul		1		
4.	Drucktaster 48 V/DC 2 A		3		
5.	ARDUINO UNO REV 3		1		
6.	Character 16x4 LCD Display Module 1604 White on Blue 5V		1		
7.	Netzteil 9V 1A		1		
8.	Präzisionspotentiometer, Trimmer, liegend, 10 kOhm		1		
9.	Widerstand 560 Ω		1		
10.	Widerstand 10000 Ω		3		
11.	Widerstand 220 Ω		1		
12.	LED Weiß		1		
13.	Flachbandkabel		50 cm		
14.	Pfosten-Steckverbinder Polzahl Gesamt: 10		1		
15.	Stiftleiste ohne Auswurfhebel Polzahl Gesamt: 10		1		
16.	Pfosten-Steckverbinder Polzahl Gesamt: 16		1		
17.	Stiftleiste ohne Auswurfhebel Polzahl Gesamt: 16		1		
Gesamt					

5. Test Ergebnisse und Lösungen

Kompatibilitätsprobleme, die in der Entwicklungsphase aufgetreten sind :

RFID und Datenlogger funktionieren nicht im zusammengebundenen System, trotz perfekt funktionierendem Teilsystem.

Lösung: Problem war nach Anpassung des Codes behoben. Datenübertragung per SPI Schnittstelle wurde richtig zugeordnet.

Langzeittest hat 5 Tagen geführt werden mit der Echtzeit „Eingang“ „Ausgang“ Anmeldung und Fehler Simulation(Z.B. Mitarbeiter hat sich nicht abgemeldet.

Nach der Durchführung des Langzeittestes wurden folgende Mängel festgestellt:

- a) Zu kurze Erscheinungszeit der Information auf dem LCD Display
- b) Gespeicherte Daten informieren nicht deutlich, dass jemand sich nicht abgemeldet hat.
- c) Es gibt die Möglichkeit die Uhr zu jedem Zeitpunkt umstellen.

Lösungen:

- a) Erscheinungszeit wurde von 1 s auf 2,5 s verlängert
- b) Bei der fehlenden Abmeldung wird im Datenfile die Zeit „17.00“ mit Markierung „Fehler“ gespeichert.
- c) Programm geändert mit folgende Logik: wenn mindestens ein Mitarbeiter schon angemeldet ist, ist es nicht möglich die Uhrzeit umzustellen.

6. Inbetriebnahmeprotokoll

Sichtprüfung

	Soll	Ist	i.O.	n.i.O.	Erl.
Bauteile vollständig bestückt, polungsrichtiger Einbau alle Teile	ja	ja	ja		x
Lötstelle	ja	ja	ja		x
Mechanische Bearbeitung	ja	ja	ja		x
Montage ordnungsgemäß	ja	ja	ja		x

Elektrische Prüfung

	Soll	Ist	i.O.	n.i.O.	Erl.
Kurzschluss test	Ok	Ok	Ja		x
Durchgangsprüfung	Ok	Ok	Ja		x
Verbindungstest	Ok	Ok	Ja		x
Spannung Prüfung in V					
Ubetrieb VCC	9	9	ja		x
Datalogger Pin1	5	5	ja		x
RFID RC522 Pin1	3,3	3,3	ja		x
LCD Display Pin3	5	5	ja		x
Massen Prüfungen					
Datalogger Pin2	GND	GND	ja		x
RFID Pin3	GND	GND	ja		x
LCD Display Pin1 Pin3 Pin16	GND	GND	ja		x

Ort, Datum

Unterschrift

7. Produktbeschreibung

Als fertiges Produkt bekommen wir einen RFID-Zeiterfassungs-Terminal für 7 Karten mit einem Datenspeicher auf der SD-Karte und RTC (Real Time Clock).

Auf der SD-Karte soll die Arbeitszeit gespeichert werden. Um das zu realisieren verwenden wir folgendes:

1. Arduino Uno mit vorgeschriebenem Programm
2. RFID Modul – der die Kode von der ID-Karte liest. Die ID-Karte mit einer Nummer, die bei der Produktion gespeichert wurde und zwar UID (Unique Identifier)
3. Diese Nummer wird einer bestimmten Person zugewiesen.
4. Beim Drücken einer aus drei Tasten, die man zur Auswahl hat („Kommen“ „Gehen“ „Infos“), hat man 7 s, um die ID-Karte zu legen.
5. Wenn die Karte schon in der Datenbank gespeichert wurde, erscheint die Angabe der Zeit und des Namens auf dem LCD-Display
6. Gleichzeitig wird Logg-File auf der SD-Karte gespeicherte. Dieser File erhält folgende Info: Datum, Zeit, Name, Eingang/Ausgang
7. Als mögliche Erweiterungen kann man für Taster „Info“ folgende Funktionen programmieren:
 - a) Info über Überstunde
 - b) Info über gesamten Arbeitszeit seit Monatsanfang
 - c) Bestätigung Überstunden durch der Karte des Leiters

Daten sind im folgenden Format auf SD-Karte Gespeichert. (Beispiel)

Datum	Zeit	Anmeldung	Name
22/6/2020	07:38:00	Kommen	Person2
22/6/2020	07:38:00	Kommen	Person3
22/6/2020	07:39:00	Kommen	Person1
22/6/2020	15:10:00	Gehen	Person1
22/6/2020	15:57:00	Gehen	Person3
22/6/2020	16:30:00	Fehler	Person2
23/6/2020	07:24:00	Kommen	Person2
23/6/2020	07:25:00	Kommen	Person1
23/6/2020	15:00:00	Gehen	Person2
23/6/2020	16:30:00	Fehler	Person1
24/6/2020	07:23:00	Kommen	Person2
24/6/2020	07:24:00	Kommen	Person3
24/6/2020	07:24:00	Kommen	Person1
24/6/2020	15:12:00	Gehen	Person2
24/6/2020	16:30:00	Fehler	Person1
24/6/2020	16:30:00	Fehler	Person3

7.1 Bedienungsanleitung

Um Gerät in den laufenden Zustand zu bringen lesen Sie bitte vollende Beschreibung.

Gerät ist nicht von Stromausfall geschützt.

Vor der Benutzung soll das Gerät durch den Facharbeiter vorprogrammiert werden.

Name des Mitarbeiters und ihm zugehörige Kartenummer muss im Programmbereich

„//***** Employees assignment *****//“

festgelegt werden.

Arbeit mit den Daten auf der SD-Karte

1. Zum Datenspeicher ist SDHC Kard verwendet. Wenn beim betrieb Fehler mit SD-Karte passieren soll, dann kommt Meldung auf Display. Die SD-Karte muss mit FAT16 oder FAT32 Dateisystem formatiert werden.
2. Die Daten, die auf der Karte gespeichert sind, kann man mit Microsoft Excel und LibreOffice öffnen und in tabellarische Form formatieren. Daten Format „.CSV“
3. Die Karte soll erst dann entfernt werden, wenn alle Mitarbeiter sich abgemeldet haben z.B. am Ende des Arbeitstages.
4. Nach der Übertragung der Daten auf PC wird empfohlen, die Daten von der Karte zu löschen.
5. Nach der Setzung der Karte ins Gerät soll die Taste „Kommen“ oder „Gehen“ gedrückt werden. Die Karte wird dann automatisch innerhalb von 10 sec. initialisiert.

An- und Abmeldungsprinzip

Bei der Ab-oder Abmeldung soll die Taste „Kommen“ oder „Gehen“ ca. 1-2 sec. gedrückt gehalten. Auf dem LCD-Display erscheint „Bitte anmelden“. Innerhalb von 7 sec. soll die ID-Karte zum Gerät angebracht werden, bis Ende der An/Abmeldung.

Zeit Umstellung

Sommer-Winterzeit Einstellung

Für die Zeiteinstellung soll man die Taste „Info“ Drücken und halten, dann mit der Taste „Kommen“ kann man die Stunden anpassen und mit der Taste „Gehen“ – die Minuten.

Zeit Umstellung verfügbar nur wenn alle Mitarbeiter abgemeldet sind.

7.2 Funktionsbeschreibung

Für alle Funktionen zu realisieren sind vier Modulen verwendet.

Modul_1 – Arduino UNO R3

Modul_2 – DatenLogger mit RTC 1307 (Real Time Clock)

Modul_3 – RFID MFRC-522 Card Reader

Modul_4 – LSD Display 16x4 HD44780

Alle Modulen sind in einem Gehäuse zusammen verbunden und mit eine Arduino Uno Plattform mit gesteuert.

7.2.1 Modul_1

Arduino UNO R3 basiert auf ein Mikrocontroller von ATMEL, der ATMEGA 328P. Auf ihm befindet sich die Arduino-Bootloader-Software. Sie ermöglicht es, Arduino-Programme auszuführen.

Das Arduino UNO verfügt über 14 digitale I/O Pins, wovon sechs als PWM Kanäle (ermöglichen z.B. das Dimmen von LEDs) genutzt werden können. Darüber hinaus sind sechs analoge Input Pins verfügbar.

Das Board beherrscht die Kommunikation per UART, SPI und I2C (TWI).

Zum Datenaustausch und zur Programmierung wird es per USB mit dem Computer verbunden. Da der Mikrocontroller nicht direkt per USB kommunizieren kann, befindet sich ein zweiter Mikrocontroller (ATmega8U2) auf dem Arduino UNO, der die „Übersetzung“ übernimmt. Arduino UNO R3 basiert sich auf ATmega328P und hat 14 digitalen input/output Pins, Keramikresonator 16 MHz, USB-Verbindung, Stromanschluss, und Reset-Taste.

7.2.2 Modul_2

DatenLogger mit RTC 1307 (Real Time Clock)

Dieser Shield verfügt über einen Slot für eine SD-Karte, eine Echtzeituhr (Real-Time-Clock). Echtzeituhr-Chip DS1307. Externe Batterie versorgt RTC mit dem Spannung, wenn der Shield nicht an eine Spannungsquelle angeschlossen ist.

Die SD-Karte ist über den SPI-Bus des Arduino angesteuert. Unterstützt werden dabei die Dateisysteme FAT16 und FAT32.

Für I2C ist Arduinobibliothek *<Wire.h>* verwendet.

Mit dieser Bibliothek kann man mit I2C / TWI-Geräten (Two-Wire-Interface, englisch für Zweidraht-Schnittstelle) kommunizieren. Auf dem Arduino befinden sich SDA PinA4 (Datenleitung) und SCL PinA5(Taktleitung). Pins A4 (SDA), A5 (SCL)

Die Wire-Bibliothek verwendet 7-Bit-Adressen.

I²C benötigt zwei Signalleitungen: Takt – (SCL = Serial Clock) und Datenleitung – (SDA = Serial Data). Der I²C-Bus arbeitet mit positiver Logik, ein High-Pegel auf der Datenleitung entspricht einer logischen „1“, der Low-Pegel einer „0“.



Adressierung - 7-Bit

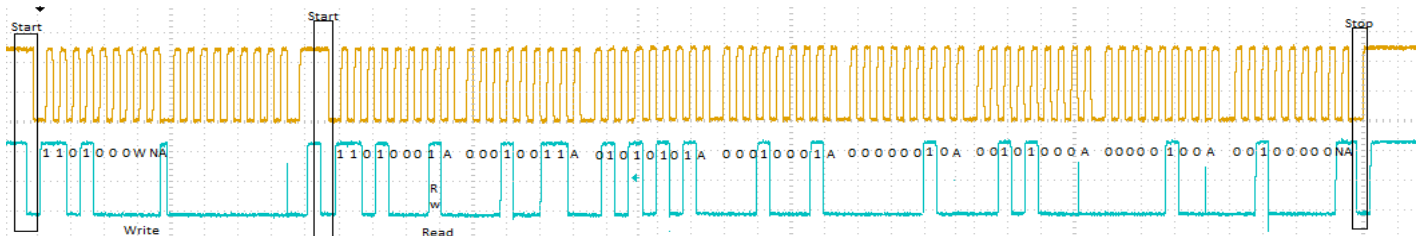
Bei Verwendung der I2C-Bibliothek Wire.h ist die korrekte Slaveadresse dadurch errechnet, dass die 7-Bit Adresse (1100_000 ohne das R/W Bit) des Slave um eine Position nach rechts (in Richtung des R/W Bits) verschoben wird.

Wie man auf Bild entnehmen kann, wird aus der Slaveadresse 0xC0 (dezimal: 192) durch die Verschiebung 0x60 (dezimal: 96).

C				0			
1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
6				0			

Signaldiagramm

Arduino – DS 1307



DATENBUS

Der DS1307 unterstützt den I2C-Protokoll und arbeitet als Slave auf der I2C Bus.

Das Gerät, das die Nachricht steuert, ist Master (Arduino Uno). Der Bus ist von einem Master-Gerät gesteuert, das generiert die serielle Clock(Takt) (SCL) PINA5, steuert den Buszugriff und generiert die START- und STOP-Bedingungen.

Dementsprechend wurden die folgenden Busbedingungen definiert:

Bus nicht besetzt: Sowohl Daten- als auch Taktleitungen bleiben HOCH.

START-Datenübertragung: Eine Änderung des Status der Datenleitung von HIGH nach LOW, während die Uhr auf HIGH steht, definiert eine START-Bedingung.

STOP-Datenübertragung: Eine Änderung des Status der Datenleitung von LOW nach HIGH, während die Taktleitung HIGH ist, definiert die STOP-Bedingung.

Daten gültig: Der Zustand der Datenleitung stellt gültige Daten dar, wenn die Datenleitung nach einer START-Bedingung für die Dauer der HIGH-Periode des Taktsignals stabil ist. Die Daten auf der Leitung müssen während der LOW-Periode des Taktsignals geändert werden.

Es gibt einen Takt pro Datenbit.

Jede Datenübertragung wird mit einer START-Bedingung eingeleitet und mit einer STOP-Bedingung beendet. Die Informationen werden byteweise übertragen und jeder Empfänger bestätigt mit einem neunten Bit.

Der DS1307 arbeitet nur im Standardmodus (100 kHz).

Bestätigung: Jedes empfangende Gerät ist verpflichtet, nach dem Empfang jedes Bytes eine Bestätigung zu generieren, wenn es angesprochen wird. Das Master-Gerät muss einen zusätzlichen Takt erzeugen, der diesem Bestätigungsbit zugeordnet ist.

Ein Gerät, das bestätigt, muss die SDA-Leitung während des Bestätigungstaktimpulses so herunterziehen, dass die SDA-Leitung während der HIGH-Periode des bestätigungsbezogenen Taktimpulses auf LOW stabil ist.

Ein Master muss dem Slave ein Datenende signalisieren, indem er auf dem letzten Byte, das aus dem

Slave getaktet wurde, kein Bestätigungsbit erzeugt. In diesem Fall muss der Slave die Datenleitung HIGH verlassen, damit der Master die STOP-Bedingung erzeugen kann.

Abhängig vom Zustand des R / W-Bits sind zwei Arten der Datenübertragung möglich:

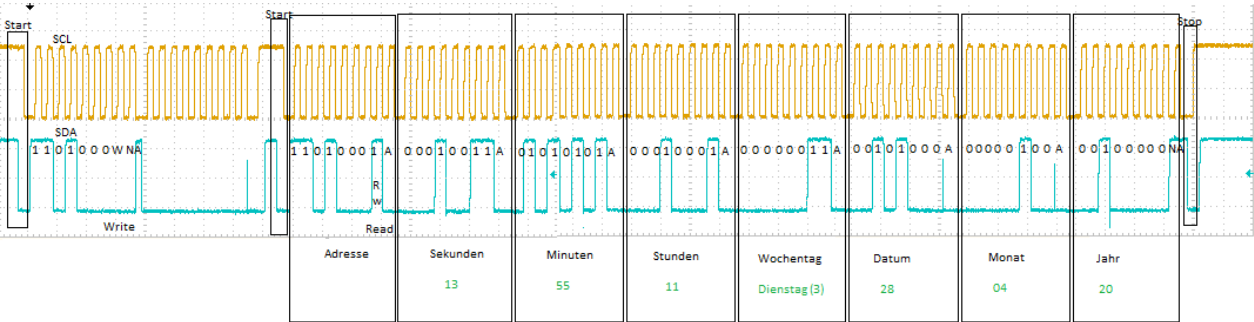
1. Datenübertragung von einem Master-Sender zu einem Slave-Empfänger. Das erste vom Master übertragene Byte ist die Slave-Adresse. Als nächstes folgt eine Anzahl von Datenbytes. Der Slave gibt nach jedem empfangenen Byte ein Bestätigungsbit zurück. Daten werden zuerst mit dem höchstwertigen Bit (MSB) übertragen.
2. Datenübertragung von einem Slave-Sender zu einem Master-Empfänger. Das erste Byte (die Slave-Adresse) wird vom Master übertragen. Der Slave gibt dann ein Bestätigungsbit zurück. Anschließend sendet der Slave eine Anzahl von Datenbytes. Der Master gibt nach allen empfangenen Bytes außer dem letzten Byte ein Bestätigungsbit zurück. Am Ende des zuletzt empfangenen Bytes wird eine "Nichtbestätigung" zurückgegeben.

In
der

Table 2. Timekeeper Registers

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

Tabelle 2 sind Datenrheinfoolge erkennbar und wir können Signaldiagramm dechiffrieren. Die Zeit- und Kalenderinformationen werden durch Lesen der entsprechenden Registerbytes erhalten. Tabelle 2 zeigt die RTC-Register. Die Zeit und der Kalender werden durch Schreiben der entsprechenden Registerbytes festgelegt oder initialisiert. Der Inhalt der Zeit- und Kalenderregister liegt im BCD-Format vor. Das Wochentagesregister wird um Mitternacht erhöht. Die Werte, die dem Wochentag entsprechen, sind benutzerdefiniert, müssen jedoch sequentiell sein (d. h. wenn 1 gleich Sonntag ist, dann sind 2 gleich Montag usw.)



Datenlogger ist mit SPI Interface gesteuert

Das Serial Peripheral Interface, ist ein Bussystem bestehend aus drei Leitungen für eine serielle synchrone Datenübertragung zwischen verschiedenen ICs.

Der Bus besteht aus folgenden Leitungen

- MOSI (Master Out → Slave In)
- MISO (Master In ← Slave Out)
- SCK (Serial Clock) – Schiebetakt

Zusätzlich zu diesen drei Leitungen wird für jeden Slave eine Slave Select (SS) (Chip Select (CS)) Leitung verwendet, in diesem Projekt sind 2 Slaves mit SPI gesteuert.

Slave Selekt selektiert der Master benötigten Slave zur aktuellen Kommunikation.

Übertragungsprinzip

Das Übertragungsprinzip funktioniert so, dass gleichzeitig 1 Bit vom Master zum Slave und 1 Bit vom Slave zum Master übertragen wird. Die Übertragung funktioniert mit Prinzip "Bitaustausch" und nicht Sender Empfänger. Man kann auch sagen, dass sowohl Master als auch Slave jederzeit sowohl Sender als auch Empfänger sind. Der Unterschied zwischen den Busteilnehmern besteht darin, dass beim SPI Bus Kommunikation immer vom Master initiiert. Er steuert SCK Leitung und wenn der Master keine SCK Pulse generiert, dann überträgt ein Slave auch nichts. Die Generierung der SCK Pulse wird dabei dadurch angestoßen, dass der Master angewiesen wird, 1 Byte über SPI auszugeben. Wenn vom Master die SCK Pulse kommen, dann muss ein Slave die MISO Leitung bedienen.

Erfolgt eine Abfrage eines Masters an einen Slave dergestalt, dass der Master ein Kommando an den Slave sendet, auf das der Slave mit einem Ergebnis zu antworten hat, dann sind dazu immer mindestens 2 Übertragungen notwendig. Im ersten Byte-Austausch übermittelt der Master sein Kommando an den Slave und im zweiten Byte-Austausch überträgt der Master einfach nur 1 Byte um dem Slave seinerseits Gelegenheit zu geben, das zuvor angefragte Ergebnis zum Master zu übertragen. Ein Slave braucht etwas Zeit, um ein gerade erhaltenes Kommando auszuwerten und die entsprechenden Ergebnisse bereit zu stellen. Ein Master soll nach dem Byteaustausch zum Zwecke der Kommandoübermittlung eine kleine Pause einzulegen, ehe er dann mit dem zweiten Byteaustausch das Ergebnis bekommt.

7.2.3 Modul_3

RFID / NFC Modul mit MFRC522 Chip, SPI Schnittstelle, 3,3V Versorgungsspannung

Das RFID- oder Radiofrequenz-Identifikationssystem besteht aus zwei Hauptkomponenten, einem Transceiver/Reader und einem Transponder / Teg, der an einem Mitarbeiter zugeordnet ist .

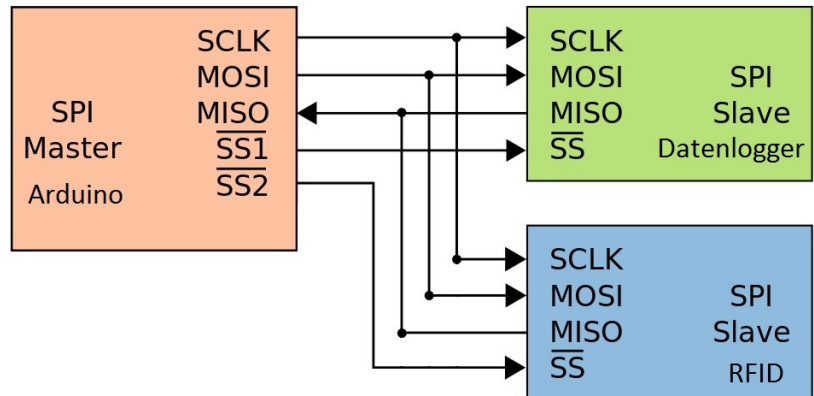
Ein Reader besteht aus einem Hochfrequenzmodul und einer Antenne, die ein hochfrequentes elektromagnetisches Feld erzeugt. Andererseits ist das Teg ein passives Gerät, das keine Batterie enthält. Stattdessen enthält es einen Mikrochip, der Informationen speichert und verarbeitet, sowie eine Antenne zum Empfangen und Senden eines Signals.

Um die auf einem Teg codierten Informationen zu lesen, soll ein Teg in unmittelbarer Nähe des Lesegeräts gelegt werden. Ein Reader erzeugt ein elektromagnetisches Feld, das bewirkt, dass sich Elektronen durch die Antenne des Tegs bewegen und anschließend den Chip mit Strom versorgen.

Der mit Strom versorgte Chip im Tag antwortet dann, indem er seine gespeicherten Informationen in Form eines anderen Funksignals an den Leser zurücksendet. Dies wird als Rückstreuung bezeichnet. Die Rückstreuung oder Änderung der elektromagnetischen / HF-Welle wird vom Lesegerät erfasst und interpretiert, das die Daten dann an einen Computer oder Mikrocontroller sendet.

Für Programmierung ist Bibliothek <MFRC522.h> verwendet.

Auf Grund, dass der RFID-Modul gleiche Kommunikationsschiene (SPI) wie Datenlogger hat, ist SPI-Verbindung durch Kaskadierung der Slaves gewählt.



7.2.4 Modul_4

Das LCD und sein Controller

LCD Display verwendet den Controller HD44780.

Für Programmierung ist Bibliothek LiquidCrystal.h verwendet

Display ist mit 4-bit-Modus angesteuert. Der **4-bit-Modus** verwendet nur die oberen vier Datenleitungen (**DB4-DB7**). Um ein Byte zu übertragen braucht man somit zwei Zugriffe, wobei zuerst das höherwertige "**Nibble**" (= 4 Bits), also Bit 4 bis Bit 7 übertragen wird und dann das niederwertige, also Bit 0 bis Bit 3. Die unteren Datenleitungen des LCDs, die beim Lesezyklus Ausgänge sind, sind offen gelassen.

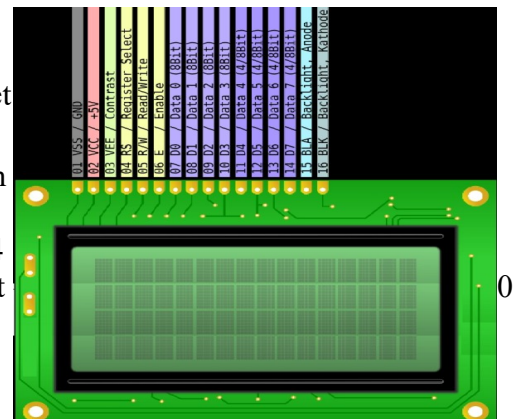
Der 4-bit-Modus hat den Vorteil, dass nur 4 IO-Pins benötigt sind.

Neben den vier Datenleitungen (DB4, DB5, DB6 und DB7) werden noch die Anschlüsse **RS**, **RW** und **E** benötigt.

Über **RS** wird ausgewählt, ob man einen Befehl oder ein Datenbyte an das LCD schicken möchte. Ist RS Low, dann wird das ankommende Byte als Befehl interpretiert, ist RS high, dann wird das Byte auf dem LCD angezeigt.

RW legt fest, ob geschrieben oder gelesen werden soll. High bedeutet lesen, low bedeutet schreiben. In unserem Fall ist RW auf GND verbinden und befindet sich immer in „schreiben“ Zustand.

Der **E** Anschluss schließlich signalisiert dem LCD, dass die übrigen Datenleitungen jetzt korrekte Pegel angenommen haben und es die gewünschten Daten von den Datenleitungen bzw. Kommandos von den Datenleitungen übernehmen kann.



8. Übergabeprotokoll

Auftraggeber

Auftragnehmer

Vladimir

97421 Schweinfurt

Ist-Zustand des Projektes bei Abgabe

Funktionsfähiger Zeiterfassungsterminal mit vollständiger Dokumentation.

Fehler die bei der Übergabe noch relevant sind

Keinen.

Änderungswünsche des Kunden/Nachtbesserungstermine

Übergebene Sachen

Der Auftragnehmer hat dem Auftraggeber einen vollständig funktionierenden RFID-Zeiterfassungsterminal inklusive Dokumentation mit

- Auftragsanalyse
 - Arbeitsplanung
 - Blockschaltbild
 - Materialliste mit Kosten
 - Layout
 - Stromlaufplan
 - Inbetriebnahmeprotokoll
 - Funktionsbeschreibung
 - Einweisungsprotokoll
 - Übergabeprotokoll
 - Wichtige Datenblättern
- übergeben.

Ort, Datum, Unterschrift Antragsnehmer

Ort, Datum, Unterschrift Auftragsgeber

9. Einweisungsprotokoll

Auftraggeber

Auftragnehmer
Vladimir

Eingewiesene Personen

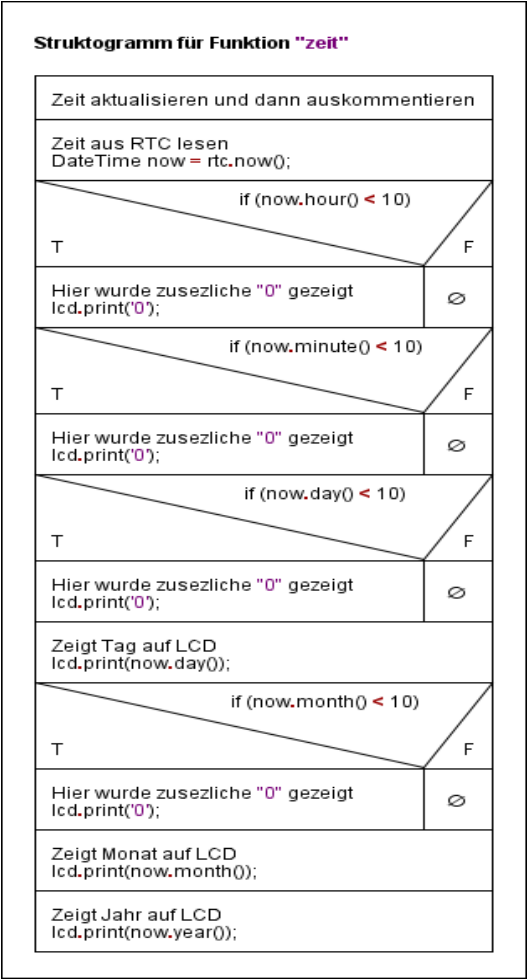
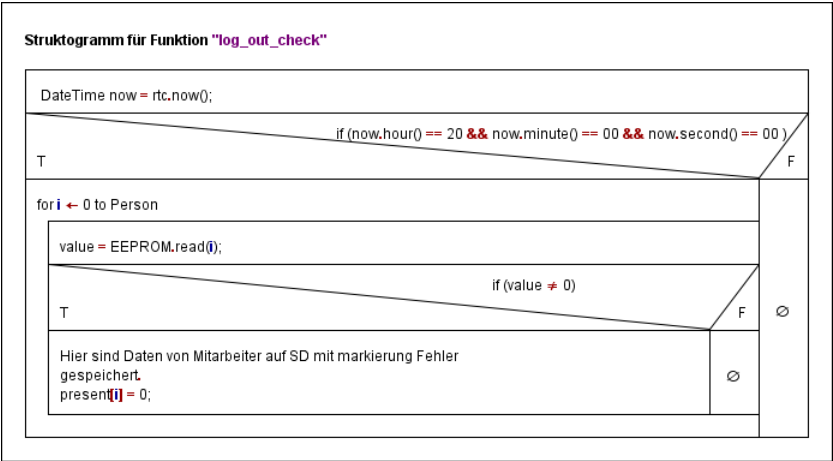
Die eingewiesene Person wurde in die Funktion und in die ordnungsgemäße Benutzung des Gerätes eingewiesen.

Ort, Datum, Unterschrift Einweiser Ort, Datum, Unterschrift eingewiesene Person

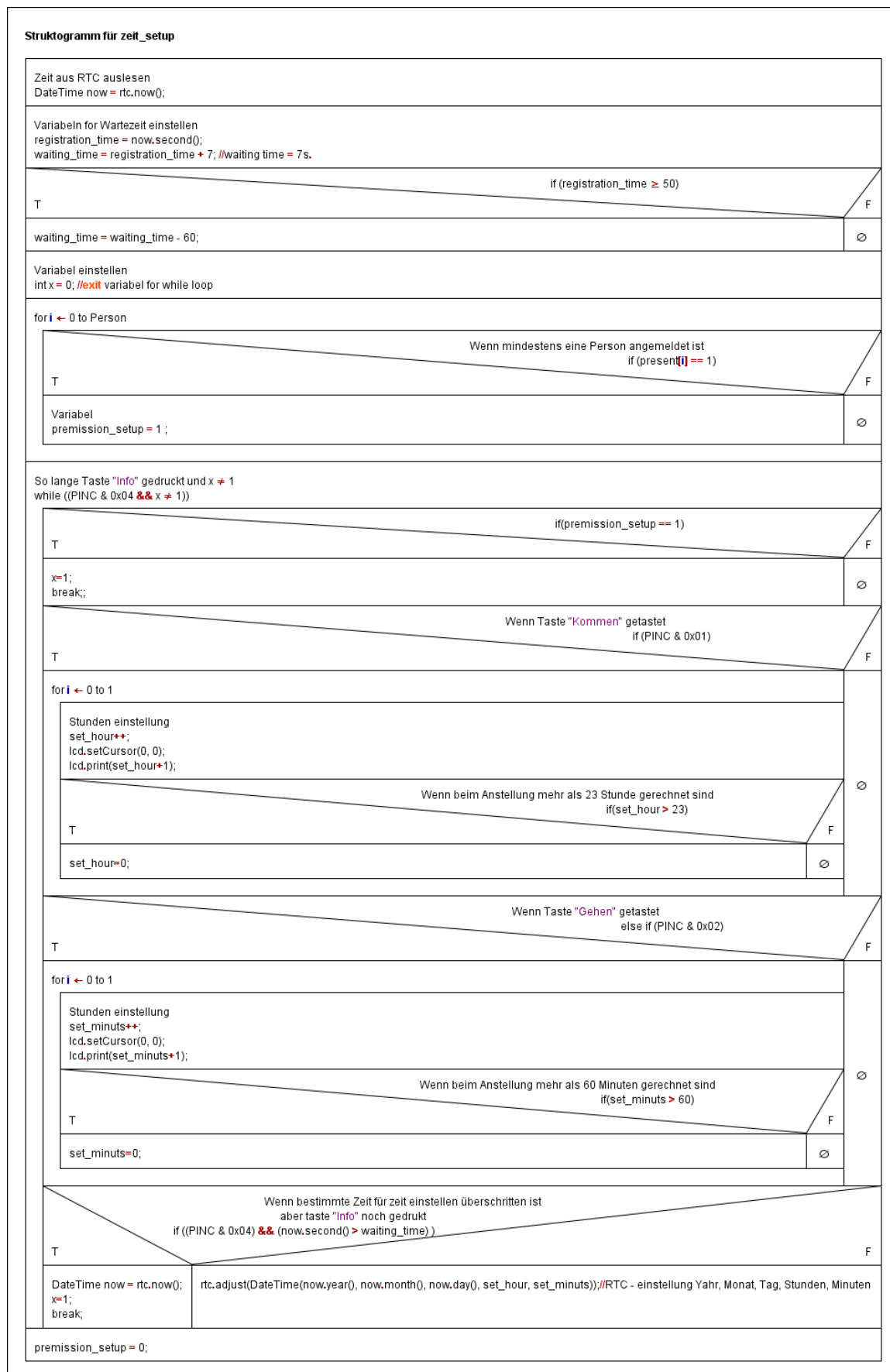
10. Anlagen

- 1.Struktogramm für Funktionen „zeit“ und „log_out_check“
- 2.Struktogramm für Funktion „zeit_setup“
- 3.Arduino Technische Daten
- 4.RFID-Modul Set
- 5.Datenlogger Bedienungsanleitung
- 6.VDE Protokol

Anlage 1



Anlage 2



Anlage 3

Technische Daten

- SRAM: 2 KB
- EEPROM: 1 KB
- Digitale I/O: 14 davon 6 als PWM verwendbar
- analoge Eingänge: 6
- Empfohlene Eingangsspannung: 7-12 V
- Maximale Eingangsspannung: 20 V
- Maximaler Strom pro I/O-Pin: 40 mA
- Belastbarkeit des 3.3-V-Ausgangs: 50 mA.

Modell

Arduino UNO

Taktfrequenz

16 MHz

Produkt-Art

Mikrocontroller

Schnittstellen

USB

Ethernet

ICSP

Serie (Embedded Mikrocontroller)

ATMega328

Anzahl analoge Eingänge

6

Anzahl I/O

14

Ausführung

ATmega328

EEPROM

1 kB

Betriebsspannung

5 V/DC

Flash-Speicher

32 kB

Betriebsspannung (Details)

5 V

Highlights & Details

- Stromversorgung über USB
- I/O-Pins: 14 davon 6 als PWM verwendbar
- Analoge Eingänge: 6
- 16 MHZ Taktfrequenz