

PROJEKT-DOKUMENTATION

Lehrgang:

IKA 8/08

Thema:

Digitales Codeschloss

Bearbeitungszeitraum:

Oktober 2009 bis Mai 2010

Verfasser:

Bott Andreas, Haltenberger Adrian,
Gantert Martin

Betreuer:

Dipl. Inf. Uwe Maulhardt

Inhaltsverzeichnis

1. Einleitung.....	4
1.1. Projekt-Idee.....	4
1.2. Projektdefinition (Lastenheft)	4
2. Projektplanung.....	5
2.1. Beschreibung des geplanten Projekts.....	5
2.2. vorraussichtlich benötigte Hardware	5
2.2.1. Auflistung der Hardware-Komponenten.....	5
2.3. Zeitplan.....	6
2.4. Kostenschätzung.....	6
3. Realisierung.....	7
3.1. Beschreibung des realisierten Projekts.....	7
3.2. Hardware (verbaut).....	8
3.2.1. Auflistung der Komponenten.....	8
3.2.2. Beschreibung der Hardware.....	9
3.2.3. Schaltplan.....	11
3.3. Zeitlicherverlauf.....	12
3.4. Projektkosten.....	12
3.5. Programmbeschreibungen.....	13
3.5.1. Admin Programm (C++)	13
3.5.1.1. Beschreibung.....	13
3.5.1.2. Programm Ablaufplan Adminprogramm.....	14
3.5.2. AVR PProgram (Assembler)	15
3.5.2.1. Beschreibung.....	15
3.5.2.2. Softwaredefinitionen.....	16
3.5.2.3. Porgrammablaufplan AVR.....	17
4. Bedienungsanleitung.....	18
4.1. Allgemein.....	18
4.2. Admin Programm (C++).....	18

<u>4.3. AVR Programm (Assembler).....</u>	<u>20</u>
<u>5. Fehlerbeschreibungen.....</u>	<u>21</u>
<u>6. Fazit / Schlussfolgerungen.....</u>	<u>23</u>
<u>7. Quellen.....</u>	<u>23</u>
<u>8. Begriffserklärungen.....</u>	<u>24</u>
<u>9. Anlagen.....</u>	<u>25</u>

1. EINLEITUNG

1.1. PROJEKT-IDEE

Die Idee entstand im Unterricht in Informationstechnik bei Herrn Hoch während der Entwicklung einer Codierschaltung. Der Hauptthemenblock des Fachs Informationstechnik beinhaltet die Programmierung von Mikrocontroller in Assemblersprache. Die Idee war für die Realisierung eines digitalen Codeschlusses einen Mikrocontroller der Baureihe RISC zu benutzen.

1.2. PROJEKTDEFINITION (LASTENHEFT)

Ein Mikrocontroller der Baureihe AVR Risc wird mit einem kleinen Zahlenblock (USB oder selbst entwickelt), LCD Anzeige und einem elektrischen Türöffner ausgestattet. Der Mikrocontroller wird so programmiert, dass alles zusammen wie ein Zahlenschloss funktioniert. Der Verwalter, der sich davor mit einer geheimen Tastenkombination und einem Sicherheitscode authentifiziert hat, kann einen Sicherheitscode eingeben, der danach benötigt wird um das elektrische Schloss zu öffnen. Die LCD Anzeige wird für die Eingabeaufforderung, Ausgaben, Fehleingaben-Hinweise und Anzahl der noch möglichen Eingabeversuche verwendet. Der Mikrocontroller wird mit dem Programm „SiSy DIE“ programmiert.

Mögliche Erweiterungen:

- Verschiedene Zugänge (Zugangcodes)
- Anzeige über LCD, welcher Zugang schon eingegeben wurde (Zugangskontrolle)
- Loggen der Eingabe-Uhrzeit eines Codes und Abfrage (Ausgabe) der Loggdatei durch den Verwalter
- Einbau in den Tresor (H14)
- Programmierung in verschiedenen Sprachen: Assembler, C++, PAP
- Zeit und Datum aller Eingaben und Abruf durch den Verwalter
- Stoßfestes Gehäuse und Montage entsprechend

2. PROJEKTPLANUNG

2.1. BESCHREIBUNG DES GEPLANTEN PROJEKTS

Zuerst wurden Informationen (Datenblätter) über die einzelnen geplanten steckbaren Komponenten von MyAVR und ihre Pinbelegungen eingeholt. Nachdem alle freien Pins des AVR definiert waren, fiel uns auf, dass nicht genügend Pins frei sind, um noch ein Tastenfeld und die Schaltung des elektrischen Türöffners zu realisieren, trotz Binär-Codierung der einzelnen Tasten des Tastenfeldes.

Nach Abwägung mehrerer Alternativen entschieden wir uns 2 AVR's zu verbinden, damit mehr Pins zur Verfügung stehen, um das Projekt realisieren zu können. Freie Pins für mögliche Erweiterungen bei frühzeitiger Fertigstellung standen jetzt auch noch zur Verfügung.

2.2. VORRAUSSICHTLICH BENÖTIGTE HARDWARE

2.2.1. AUFLISTUNG DER HARDWARE-KOMPONENTEN

- 2 x AVR Atmega8L (myAVR Board MK2 USB)
- Tastaturfeld (0-9,*,#)
- LC-Display (myAVR LCD Add-On)
- Relaiskarte (myDigitalOut)
- 10-to-4 Codierer (SN74LS147N DIL)
- Echtzeituhr (myTWI Add-On Echtzeituhr)
- USB-Typ B Buchse und USB Kabel von AVR zur USB Buchse
- Motor für Schließanlage
- MyAVR Laborkarte A (myAVR Rasterleiterplatte, Industriequalität)

2.3. ZEITPLAN

Planung:

Monat	Oktober					November				Dezember									
KW	40	41	42	43	44	45	46	47	48	49	50	51	52						
Planung																			
Hardware																			
Software																			
Einbau																			
Testphase																			
Doku																			
Monat	Januar				Februar				März				April		Mai				
KW	53	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Planung																			
Hardware																			
Software																			
Einbau																			
Testphase																			
Doku																			

2.4. KOSTENSCHÄTZUNG

Menge	Artikelbeschreibung	Preis
2	MyAVR MK2 USB Board	49,00 €
1	MyAVR Laborkarte	6,45 €
1	MyAVR LC-Display Add-On	19,49 €
1	MyAVR Digital Out Add-On	15,95 €
1	MyAVR TWI Echtzeituhr Add-On	9,95 €
1	Tastenfeld	ca. 15,00 €
1	USB Buchse + USB Kabel	ca. 1,50 €
1	Motor	ca. 30,00 €
1	10-to-4 Codierer (SN74LS147N DIL)	ca. 0,50 €
	Gesamt	147,84 €

Bott Andreas
Haltenberger Adrian
Gantert Martin

Bernd-Blindow- Schule
Freidrichshafen

3. REALISIERUNG

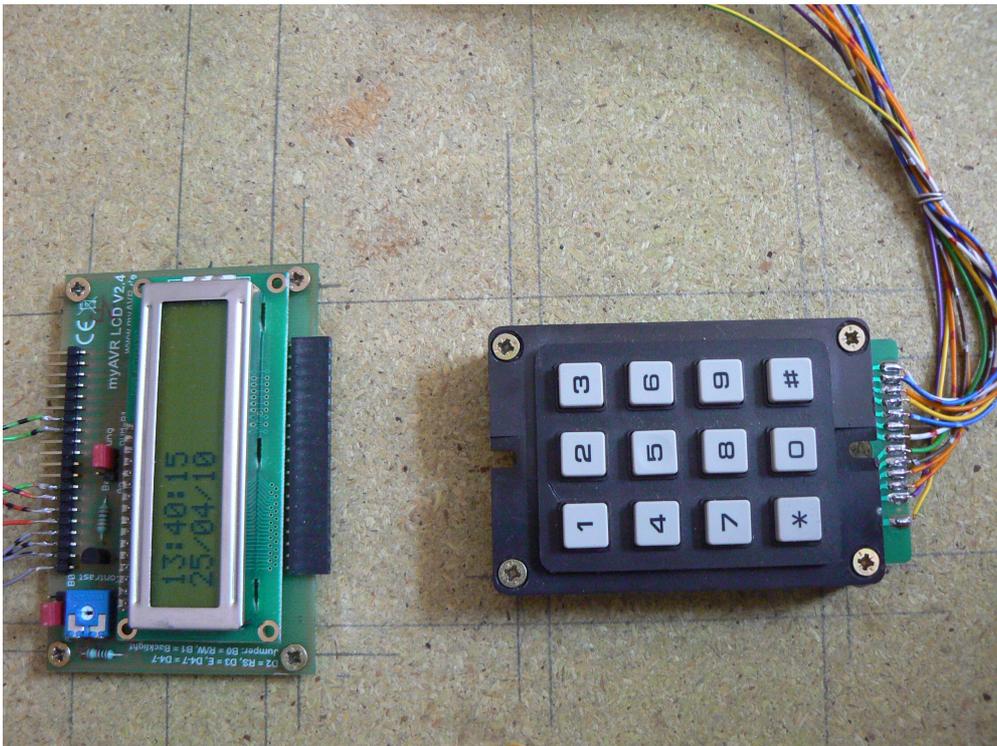
3.1. BESCHREIBUNG DES REALISIERTEN PROJEKTS

Bei der Entwicklung der Hardware und einigen Programmtests fiel uns auf, dass die Realisierung mit 2 AVR's nicht nur teurer, sondern auch viel aufwändiger ist, als geplant. Aus diesen Gründen suchten wir nach einer anderen Lösung des Problems. Mit einer Schließenanlage, welche nur 1 Pin benötigt (unidirektional), würden die Pins ausreichen und alle definierten Erweiterungen trotzdem möglich sein. Die Schließenanlage musste dementsprechend konzipiert werden, damit Sie nicht nur öffnet, sondern auch wieder schließt. Dies wurde durch eine Feder gelöst, welche den Schließmechanismus wieder in den Ursprungszustand zieht.

Pinbelegungen 1 bis 20:

- PIN 1 bis 8 LC-Display
(interne PORTs D2-D7, B0-B1)
- PIN 9 bis 12 Tastenfeld: Tasten 1-9 binär codiert über 10-to-4
Line Encoder (interne PORTs B2-B5)
- PIN 13 und 14 Stromversorgung +5V und Masse
- PIN 15 bis 17 Tastenfeld: Tasten *,0,# uncodiert
(interne PORTs C0-C2)
- PIN 18 Schließenanlage
(interner PORT C3)
- PIN 19 bis 20 Echtzeitmodul TWI
(interne PORTs C4-C5)

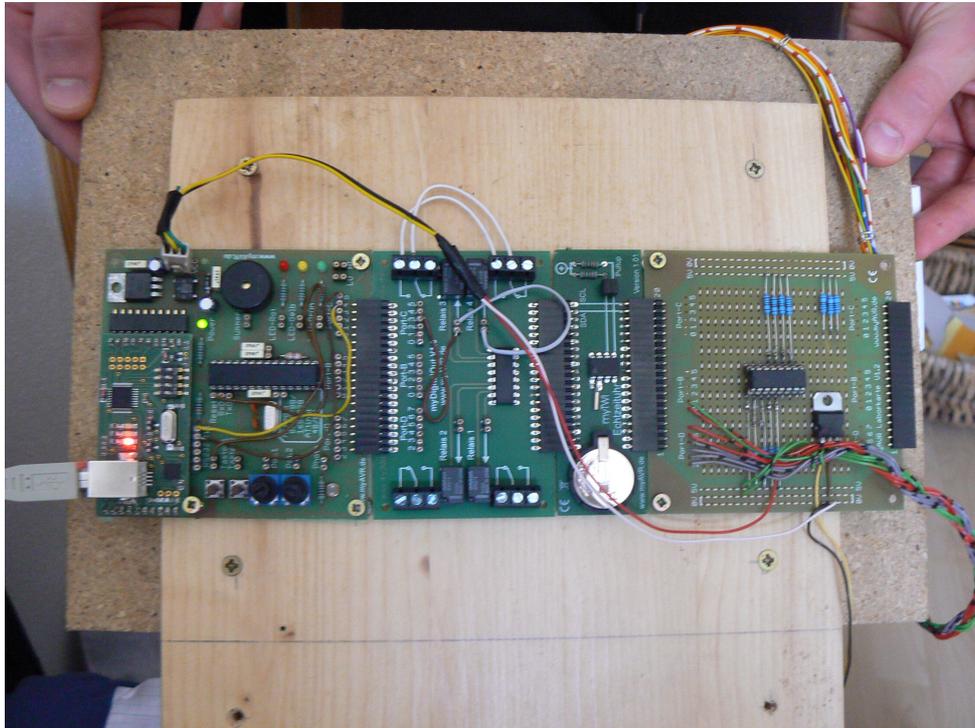
3.2. HARDWARE (VERBAUT)



3.2.1. AUFLISTUNG DER KOMPONENTEN

- AVR Atmega8L (myAVR Board MK2 USB)
- Tastaturfeld (0-9,*,#)
- LC-Display (myAVR LCD Add-On)
- Relaiskarte (myDigitalOut)
- 10-to-4 Line Endocder (SN74LS147N DIL)
- Echtzeituhr (myTWI Add-On Echtzeituhr)
- Spannungsregler (L78S09CV)
- USB-Typ B Buchse und USB Kabel von AVR zur USB Buchse
- MyAVR Laborkarte A (myAVR Rasterleiterplatte, Industriequalität)

- Schiebemotor + diverse Metallteile (Kugellager, Stange, Blech, Schrauben, Muttern)



3.2.2. BESCHREIBUNG DER HARDWARE

Es wurde ein **AVR Experimentierboard** (myAVR Board MK2 USB) von MyAVR (myAVR.de) verbaut, welches mit einem RISC **Atmega8L** bestückt ist.

Dieses beinhaltet auch den notwendigen **USB Programmer**, um den Atmega8L zu beschreiben.

Zur **Eingabe** der Codes wurde ein Tastaturfeld (0-9,*,#) installiert. Diese wurde an einen dezimal zu binär Codierer (10-line-to-4 Encoder SN74LS147N DIL) angeschlossen. Damit das Tastenfeld nur 7 PINS (4 Codiert, 3 Uncodiert) anstatt ohne Codierung 12 PINS belegt, da der AVR nur eine begrenzte Anzahl an PINS zur Verfügung stellt. Die Tasten 0,* und # wurden direkt ohne Codierung angeschlossen. Alle Tasten wurden mit einem Pull-Up-Widerstand versehen.

Für die **Ausgabe** wurde ein anschlussfertiges LC-Display (myAVR LCD Add-On) von MyAVR mit 2 Zeilen a 16 Zeichen verbaut.

Um genaue **Zeit- und Datum** Angaben zu erhalten, wurde ein Echtzeitmodul (myTWI Add-On Echtzeituhr) von MyAVR verbaut. Durch

eine Knopf-batterie wird auch bei komplettem Stromausfall sichergestellt, dass die Uhrzeit und Datum weiterhin aktuell sind und bleiben.

Die **Ansteuerung der Schließanlage** wurde durch eine Relais-Experimentierkarte (myDigitalOut) von MyAVR realisiert. Diese Karte ist mit 4 Relais a 30 W Maximalbelastung ausgestattet.

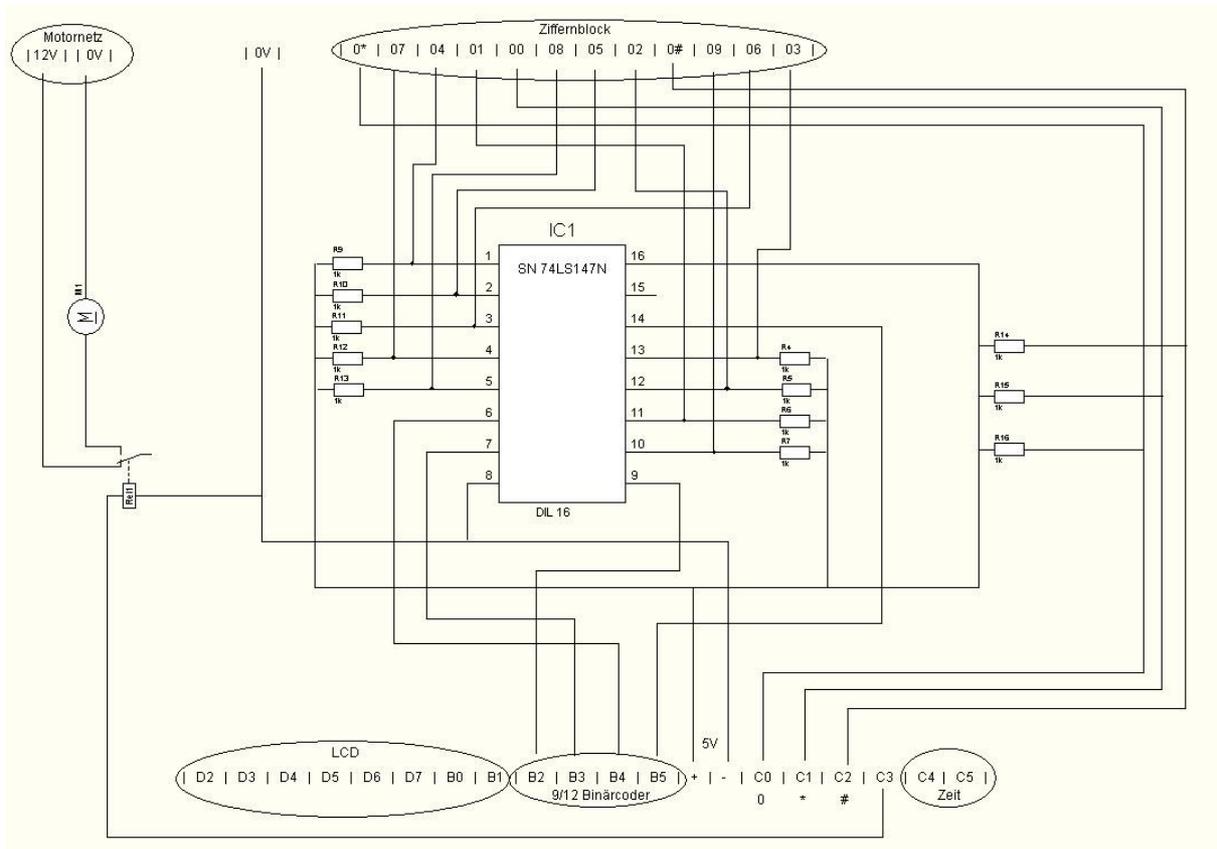
Die oben genannte Hardware wurde in die Türe des **Tresors eingebaut**, damit Display und Tastenfeld gut sichtbar an der Vorderseite der Tür zu bedienen sind. Die Stromversorgung wurde nicht mit in die Türe eingebaut.

Die **Schließanlage** selbst wurde mit einem **Schiebemotor** der Schließanlage aus einem alten Auto (Schrottplatz) realisiert. Dieser zieht zum Öffnen der Tür einen Stift, an einer kurzen Stange auf einem Kugellager aus einer Kerbe, damit die ursprüngliche Schließanlage des Tresors von der Blockierung gelöst wird und somit wieder wie ursprünglich verbaut (Schlüssel und Drehhebel) funktioniert. Das Kugellager und notwendige Bleche wurden mit der Tresortüre verschweißt. Wird die Stromzufuhr des Motors wieder entzogen, zieht eine eingebaute Feder die Stange mit Stift wieder in die Kerbe, sobald der Drehhebel wieder in Schließstellung gebracht wurde.

Als **Stromquelle** wurde ein altes **PC-Netzteil** verwendet, welches die nötige Leistung für die Schließanlage liefert. Außerdem wurde mit einem 12zu9 Volt Spannungsregler (L78S09CV) die Stromversorgung für den AVR, LCD, Echtzeit-modul, Tastenfeld und Relaiskarte realisiert.

Außerdem wurde eine **USB-Typ-B Buchse** an der Innenseite der Tresortür befestigt, um einen PC für das Admin-Programm anschließen zu können.

3.2.3. SCHALTPLAN



Bott Andreas
 Haltenberger Adrian
 Gantert Martin

Bernd-Blindow- Schule
 Freidrichshafen

3.3. ZEITLICHER VERLAUF

Realisierung:

Monat	Oktober					November				Dezember									
KW	40	41	42	43	44	45	46	47	48	49	50	51	52						
Planung	[Blue]					[Blue]				[Blue]									
Hardware						[Yellow]				[Yellow]									
Software						[Orange]				[Orange]									
Einbau																			
Testphase										[Cyan]									
Doku						[Yellow]				[Yellow]									
Monat	Januar				Februar				März				April				Mai		
KW	53	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Planung		[Blue]																	
Hardware	[Yellow]				[Yellow]														
Software	[Orange]				[Orange]														
Einbau									[Yellow]				[Yellow]				[Yellow]		
Testphase	[Cyan]				[Cyan]				[Cyan]				[Cyan]				[Cyan]		
Doku	[Yellow]				[Yellow]				[Yellow]				[Yellow]				[Yellow]		

3.4. PROJEKTKOSTEN

Menge	Artikelbeschreibung	Preis
1	MyAVR MK2 USB Board*	49,00 €
1	MyAVR Laborkarte	6,45 €
1	MyAVR LC-Display Add-On*	19,49 €
1	MyAVR Digital Out Add-On	15,95 €
1	MyAVR TWI Echtzeituhr Add-On	9,95 €
1	Tastenfeld	ca. 15,00 €
1	USB Buchse + USB Kabel *	ca. 1,50 €
1	Motor *	ca. 30,00 €
1	10-to-4 Codierer (SN74LS147N DIL) *	ca. 0,50 €
	Gesamt	99,89 €

* = Beschaffung durch ELO - Raum, Schrottplatz, Privatbestände

Bott Andreas
Haltenberger Adrian
Gantert Martin

Bernd-Blindow- Schule
Freidrichshafen

3.5. PROGRAMMBESCHREIBUNGEN

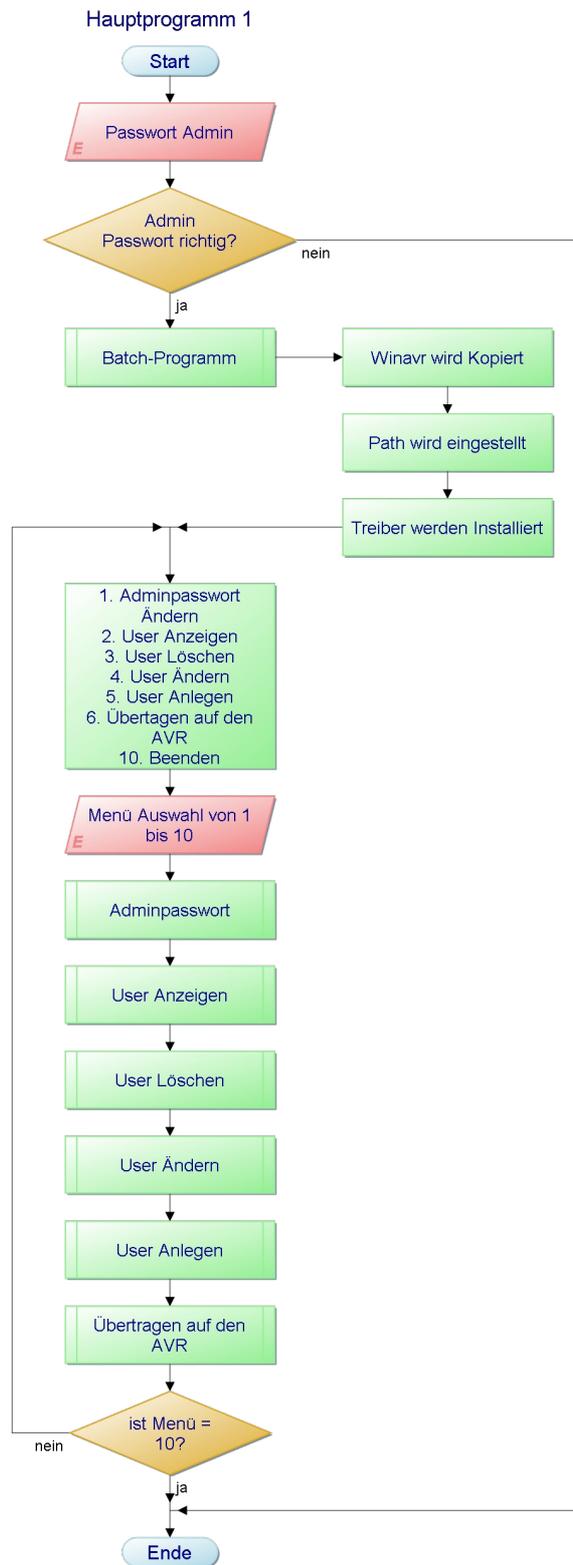
3.5.1. ADMIN PROGRAMM (C++)

3.5.1.1. BESCHREIBUNG

Das Administrations-Programm ist in der Programmier Sprache C++ geschrieben. Es dient zur einfacheren Administration des Code-Schlusses. Dieses ist Passwort geschützt, gleichzeitig ist das Admin-Passwort auch das Passwort für das AVR Programm (Codeschloss), damit kein Unbefugter User die Konfiguration ändern kann. Mit Hilfe dieses Programms können maximal 10 User oder weniger angelegt werden, die das Codeschloss öffnen dürfen. Jeder User wird mit Vornamen und Nachnamen in das Programm eingefügt und gespeichert. Damit der AVR die User kennt, müssen die Einstellungen auf den AVR übertragen (gebrannt) werden.

Siehe Bedienungsanleitung.

3.5.1.2. PROGRAMM ABLAUFPLAN ADMINPROGRAMM



3.5.2. AVR PROGRAMM (ASSEMBLER)

3.5.2.1. BESCHREIBUNG

Zu Beginn wird das Display langsam dunkel gedimmt bis keine Hintergrundbeleuchtung mehr vorhanden ist. Es wird die Uhrzeit und das Datum auf dem Display angezeigt bis eine Taste gedrückt wird. Nun fragt das in Assembler-Sprache programmierte Programm für den AVR vom Benutzer einen 4-stelligen Usercode ab. Es vergleicht diesen eingegeben Code mit den gespeicherten Codes, welche zuvor mit dem Admin Programm auf den AVR gebrannt wurden. Findet eine Übereinstimmung mit einem Usercode statt, wird ein Signal an das verbaute Relais gegeben, welches die Schließanlage der Tresortür betätigt (öffnet). Gibt es keine Übereinstimmung mit einem gespeicherten Code, hat der Benutzer 3 Fehlversuche bis das Programm eine Strafzeit verhängt. In dieser Zeit es nicht möglich das Codeschloss zu benutzen. Es werden Strafzeiten bei 3,6 ,9 oder 10 und mehr Falscheingaben verhängt, welche entsprechend den Falscheingaben lang sind.

Anzahl der Falscheingaben	Strafzeitlänge
3	15 Sekunden
6	30 Sekunden
9	1 Minute
10 oder mehr	2 Minuten

Ist der eingegebene Code der Admincode, gelangt der Benutzer in das **Adminmenü**:

Menüpunkt	Aktion
1. Türe öffnen	Betätigen der Schließanlage (Türe auf)
2. letzte 10 Zugänge mit Usernamen	Zeigt Zeit, Datum und Username der letzten 10 richtigen Eingaben
3. letzte 3 Falscheingaben	Zeigt Zeit und Datum der 3 letzten Falscheingaben
4. letzter Adminlogin	Zeigt Zeit und Datum des letzten Adminlogins (Login vor aktuellem Login)
5. letzter Reboot	Zeitstempel des letzten Reboots. Admin-Programm (brennen) oder bei kurzer Trennung jeglicher Stromversorgung

Nachdem ein richtiger Usercode eingegeben wurde oder der Admin Menüpunkt 1 ausgewählt hat, also die Tür geöffnet wird, startet das Programm wieder von Anfang an.

Bott Andreas
Haltenberger Adrian
Gantert Martin

Bernd-Blindow- Schule
Freidrichshafen

3.5.2.2. SOFTWAREDEFINITIONEN

UserIDs:

Admin: FF
User 1-9: 01-09
User10: 0A

Speicherbelegungen:

Benutzercodes im Flash/SRAM:

Benutzercodes : 0x0060 – 0x008B (Admin + 10 User je 4 Byte)
Aktueller Code: 0x008C - 0x008F (4 Byte)

Zeitstempel im EEprom:

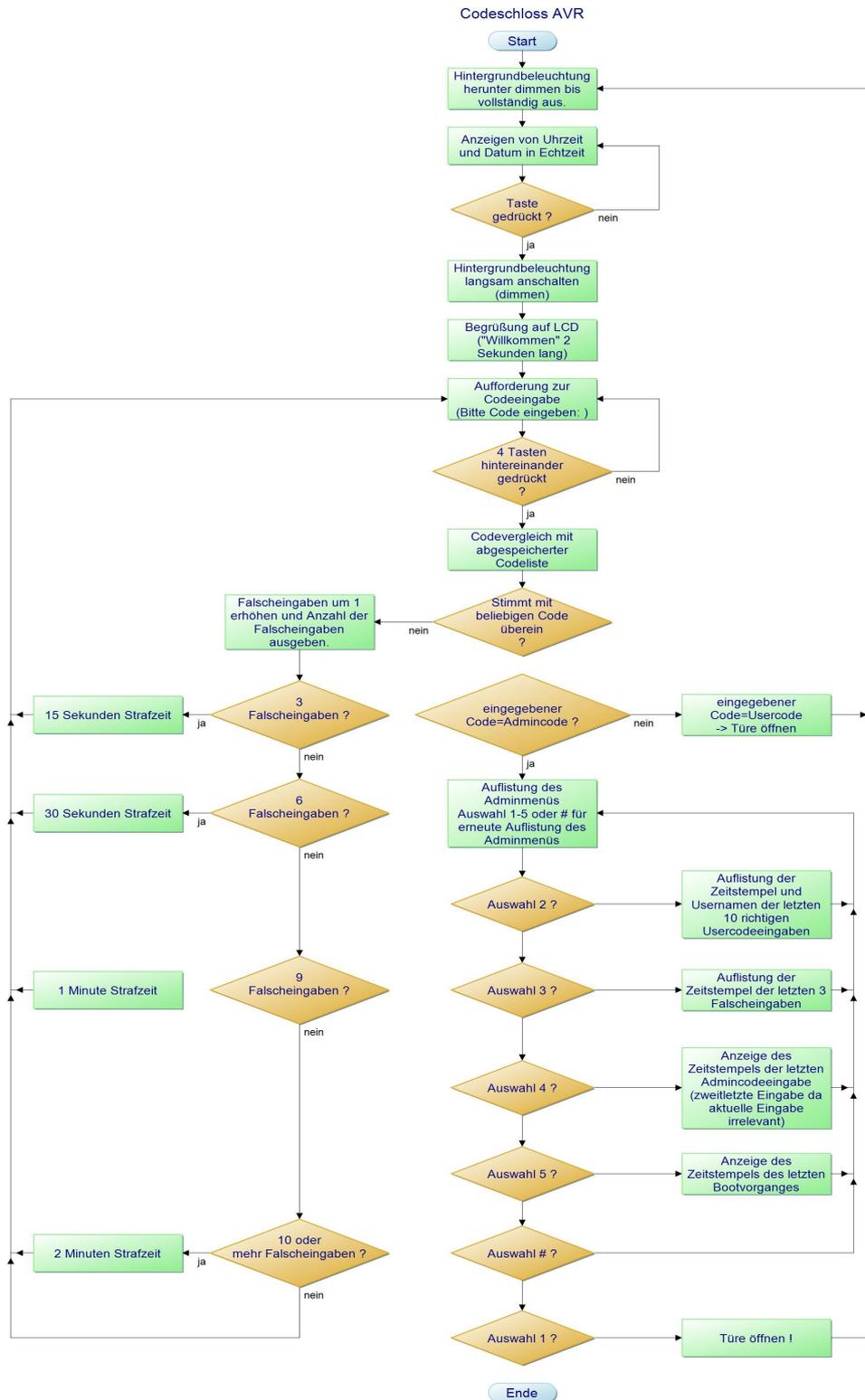
Letzte 10 richtige Eingaben mit UserID :
(3 Byte Zeit, 3 Byte Datum, 1 Byte User ID = 7 Byte)
Adresse 0x0090 – 0x00D5 (10 Stempel je 7 Byte)

Letzte 3 Falscheingaben:
Adresse 0x00D6 – 0x00E7 (3 Stempel je 6 Byte)

Letzter Adminlogin:
Adresse 0x00E8 – 0x00ED und 0x00F6 – 0x00FB als Puffer

Letzter Reboot:
Adresse 0x00EF – 0x00F5

3.5.2.3. PORGRAMMABLAUFPLAN AVR



Bott Andreas
Haltenberger Adrian
Gantert Martin

Bernd-Blindow- Schule
Freidrichshafen

4. BEDIENUNGSANLEITUNG

4.1. ALLGEMEIN

Das Codeschloss benötigt 2 Programme. Ein Adminprogramm und ein AVR Programm. Das Adminprogramm modifiziert das AVR Porgramm mit den angelegten Benutzern und brennt es auf den AVR. Das Adminprogramm und alle notwendigen Dateien, Treiber und Scripte sind auf dem mitgelieferten USB-Stick. Um das Codeschloss verwenden zu können, muss mindestens 1 Mal nach Auslieferungszustand im Adminprogramm Menüpunkt 6 (brennen) ausgeführt werden.

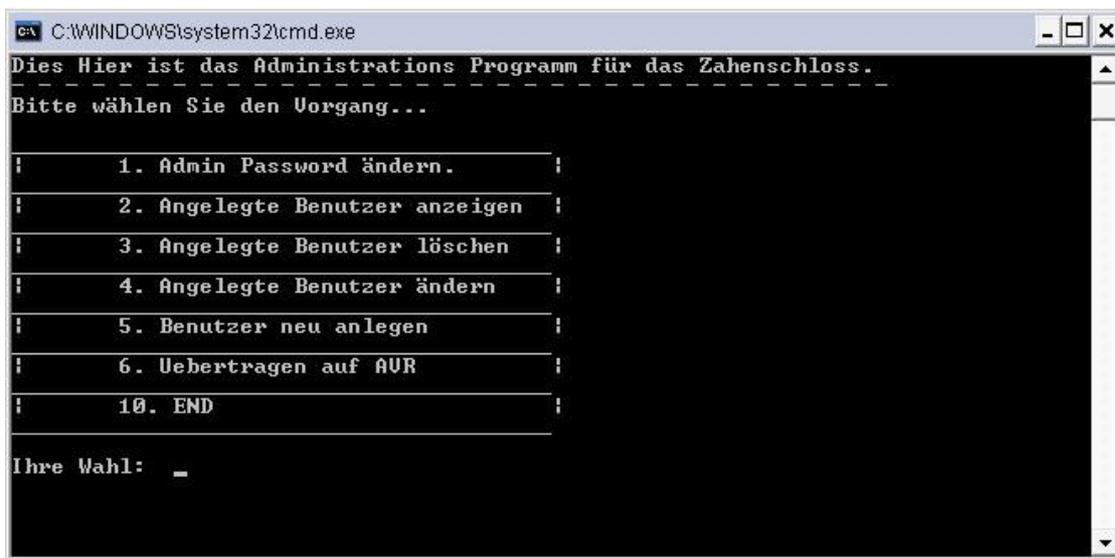
4.2. ADMIN PROGRAMM (C++)

Als Voraussetzung, um das Adminprogramm korrekt ausführen zu können, wird ein Microsoft Betriebssystem (Windows XP mit SP2 empfohlen) benötigt.

1. Gehen Sie an das Codeschloss und geben Sie den **Code 0000** ein, um in das Adminmenü zu gelangen und wählen Sie dort Menüpunkt 1. (Türe öffnen) , um das USB-Kabel später anschließen zu können. Öffnen Sie nun die Türe.
2. Führen sie die Datei: **Adminprogramm.exe** aus , welche sich auf dem USB Stick befindet.
3. Geben Sie das voreingestellte (**Auslieferungspasswort: 0000**) oder geänderte Adminpasswort ein.

Jetzt gelangen Sie in das Hauptmenü des Adminprogramms (siehe Bild).

Adminprogramm-Menü:



```
C:\WINDOWS\system32\cmd.exe
Dies hier ist das Administrations Programm für das Zahenschloss.
Bitte wählen Sie den Vorgang...

| 1. Admin Password ändern. |
| 2. Angelegte Benutzer anzeigen |
| 3. Angelegte Benutzer löschen |
| 4. Angelegte Benutzer ändern |
| 5. Benutzer neu anlegen |
| 6. Uebertragen auf AUR |
| 10. END |

Ihre Wahl: _
```

4. Bitte verbinden Sie Ihren Computer und das Codeschloss mit dem beigelegten USB-Kabel. Die USB-Buche des Codeschlusses befindet sich an der Innenseite der Tresortüre.

5. Jetzt sind Sie im Hauptmenü. Hier können Sie Benutzer und die zugehörigen Gehimcodes anlegen:

Menüpunkt 1. Ändern des Adminpassworts (Wichtig)

Menüpunkt 2. alle Benutzer anzeigen lassen

Menüpunkt 3. einzelne Benutzer löschen

Menüpunkt 4. einzelne Änderungen der Benutzerdaten vornehmen.

Menüpunkt 5. Benutzer anlegen

6. Jetzt **müssen** Sie den **Menüpunkt 6.** auswählen, damit alle **Änderungen auf dem Codeschloss wirksam** werden. Dies kann einige Sekunden in Anspruch nehmen, warten Sie bis das Menü wieder sichtbar ist.

7. Zum Beenden des Adminprogramm **Menüpunkt 10.** auswählen.

Das Zahlenschloss ist nun einsatzbereit!

Bott Andreas
Haltenberger Adrian
Gantert Martin

Bernd-Blindow- Schule
Freidrichshafen

4.3. AVR PROGRAMM (ASSEMBLER)

Nach dem Hinzufügen der Stromversorgung wird die Uhrzeit auf dem LC-Display angezeigt. Bei Tastendruck wird die Hintergrundbeleuchtung gedimmt und das Programm startet.

1. Es wird auf dem Display eine Aufforderung zur Code-Eingabe dargestellt: **„Bitte Code eingeben:“**
2. **Geben** Sie jetzt Ihren **Usercode / Admincode ein**.
 - Bei einer **Falscheingabe** werden Sie erneut aufgefordert einen Code einzugeben. Geben die mehrmals einen falschen Code ein, müssen Sie gewisse Strafzeiten abwarten. Je öfter die Falscheingaben, desto länger die Strafzeiten
 - Bei der Eingabe eines **Usercodes** wird die Schließanlage betätigt (Tresortüre geöffnet).
 - Bei der Eingabe des **Admincodes** gelangen Sie in das Adminmenü.

Adminmenü:

- **1. Türe öffnen**
Hier wird die Schließanlage betätigt und die Tresortüre kann geöffnet werden. Danach startet das Programm wieder von Anfang an.
- **2. Letzten 10 Zugänge anzeigen**
Hier werden die Zeitstempel der letzten 10 richtigen Codeeingaben angezeigt. Jeweils 2 Sekunden lang.
- **3. Letzte 3 Falscheingaben anzeigen**
Hier werden die Zeitstempel der letzten 3 Falscheingaben angezeigt.
- **4. Letzter Adminlogin anzeigen**
Hier wird ein Zeitstempel der letzten Eingabe (vorletzte, da Aktuelle unwichtig) des Admincodes angezeigt.

- **5. Letzten Reboot anzeigen**

Hier wird der Zeitstempel des letzten Reboots angezeigt. Ein Reboot wird bei jedem Brennen (Ändern der Usercodes über das Adminprogramm) oder bei Stromausfall des AVR durchgeführt.

5. FEHLERBESCHREIBUNGEN

- Verbinden von 2 AVRs

Die Synchronisation beider AVRs war schwieriger zu realisieren als geplant. Durch nachträgliche Recherchen und Überlegungen, haben wir aus Kostengründen und Aufwandsersparungen die Realisierung mit nur einem AVR entwickelt.

- Spannungsspitzen des Schiebemotors

Beim Einbau des Motors in Verbindung mit der AVR Hardware entstand beim Zurückziehen des Schiebemotors durch die Rückzugfeder eine Spannungsspitze. Diese Spannungsspitze schlug durch das Relais auf den AVR und seine Komponenten. Die Echtzeituhr und das LCD Display wurden dadurch beschädigt und funktionsunfähig gemacht. Nachdem die 2 Komponenten wieder bestellt und eingetroffen waren, haben wir das Problem durch das Parallelschalten eines Varistors am Schiebemotor gelöst. Jetzt treten keine Spannungsspitzen mehr auf.

- Stringverarbeitung in Assembler Sprache

Beim Ansprechen des LC-Displays über Strings kam beim Compilieren manchmal eine Fehlermeldung:

„(.text+0x4c): warning: internal error: out of range error“

Nach langer Recherche im Internet und einigen Testversuchen fiel uns auf, dass die angegebenen Strings im Bootvorgang initialisiert werden müssen und alle Zeichen zusammen eine gerade Zahl ergeben müssen.

- Entprellung des Tastenfelds

Beim Anschluss des Tastenfelds musste nicht nur durch ein internen Pull-Up Widerstand (softwareseitig) und einen verlöteten Pull-Up Widerstand (1 k Ohm) die einzelnen Tasten entprellt werden. Es war auch noch nötig eine Zeit von 40 ms nach drücken einer Taste zu warten und die Taste noch einmal einzulesen, um ein entprelltes Signal zu bekommen.

- Anpassung des Adminprogramms an das AVR Programm

Bei der Zusammenführung (Modifizierung des AVR Programmcodes durch das Adminprogramm) gab es Probleme bei der Pointersetzung. Das Problem lag in der Berechnung des Pointerwertes, um die einzelnen Zeichen zu ändern. Durch längere Überlegungen und mehreren Testphasen haben wir das Problem gelöst.

Natürlich sind noch viel mehr Fehler und Probleme aufgetreten, die wir an dieser Stelle nicht alle erwähnen möchten, da es sich meist um Kleinigkeiten handelte, welche schnellstens behoben werden konnten.

6.FAZIT / SCHLUSSFOLGERUNGEN

Die am Anfang erstellte Zeitplanung konnte nicht eingehalten werden, da Probleme auftraten, welche den zeitlichen Verlauf stark beeinflussten. Die Organisation des Projektteams und die zugehörige Aufgabenverteilung war eine Herausforderung, doch kein Hindernis um das Projekt erfolgreich abzuschließen. Der Lerneffekt für jeden Projektteilnehmer war sehr groß und stärkte die Kenntnisse in verschiedenen Bereichen unseres zukünftigen Berufsfeldes. Folgende Bereiche unserer Ausbildung wurden durch das Projekt vertieft:

- Assemblerprogrammierung (Informationstechnik)
- C++ Programmierung objektorientiert (Programmiertechnik)
- Elektronische Kenntnisse (Elektrotechnik)
- Office Kenntnisse (Anwendungssoftware)
- Batchscripte und DOS Befehle (Kommunikationstechnik)
- Organisation (Projektmanagement)

Trotz Höhen und Tiefen hat uns allen das Projekt viel Spaß bereitet und viele Erfahrungswerte vermittelt.

7.QUELLEN

- Wikipedia (www.Wikipedia.de)
- MyAVR Webseite (www.myAVR.de)
- Unterprogrammbeispiele des MyAVR Workpad
- Roboter Netz Webseite (www.RN-Wissen.de)
- Atmel Webseite (www.Atmel.com)
- C++ Webseite (www.CPlusPlus.com)
- Conrad Katalog (www.Conrad.de)
- Verschiedene Unterrichtsaufschriebe und hilfsbereite Lehrer

8. BEGRIFFERKLÄRUNGEN

TWI:

TWI (**T**wo-**w**ire Serial **I**nterface) ist eine Bezeichnung von Atmel für die auf vielen AVR Megas vorhandene I²C-Hardware.

Assembler:

Ein Assembler (nach DIN 44300) ist ein Hilfsprogramm der Programmierung, das ein in maschinennaher Assemblersprache geschriebenes Computerprogramm in Maschinensprache (auch *Maschinencode* genannt) übersetzt. *Assembler* wird häufig auch als Synonym für Assemblersprache benutzt.

Batchscripte:

Stapelverarbeitung oder auch Batchverarbeitung (Batchscript) ist ein Begriff aus der Datenverarbeitung und bezeichnet die sequentielle Bearbeitung von Aufgaben.

C++:

C++ ist eine von der ISO standardisierte höhere Programmiersprache. C++ ermöglicht sowohl die effiziente und maschinennahe Programmierung als auch eine Programmierung auf hohem Abstraktionsniveau.

Objektorientiert:

Die objektorientierte Programmierung (kurz OOP) ist ein auf dem Konzept der Objektorientierung aufgebaut. Die Grundidee dabei ist, Daten und Funktionen möglichst eng in einem so genannten Objekt zusammenzufassen und nach außen hin zu *kapseln*, so dass Methoden fremder Objekte diese Daten nicht versehentlich manipulieren können.

myAVR:

Eine Firma, welche Mikrocontrollerlösungen für Ausbildung und Selbststudium anbietet. Die Webseite beinhaltet einen Shop, sowie ein nützliches Forum mit Codebeispielen.

AVR:

"AVR" steht angeblich für *Advanced Virtual RISC* (laut Entwickler des AVR-Kerns Alf Egin Bogen und Vegard Wollan). Laut [Atmel](#) hat es keine Bedeutung.

Varistor:

Ein Varistor ist ein elektronisches Bauteil. Es handelt sich dabei um einen spannungsabhängigen Widerstand. Varistor ist ein aus den englischen Begriffen „variable resistor“ zusammengesetztes Kofferwort. Varistoren werden auch als VDR bezeichnet. VDR steht für *Voltage Dependent Resistor*, also spannungsabhängiger Widerstand.

Comilieren:

Unter Kompilierung (auch *Compilierung*, *Übersetzung* oder *Wandlung*) versteht man in der EDV die Anwendung eines Compilers auf den Quelltext eines Computerprogramms. Dabei wird das in einer Quellsprache geschriebene Programm in ein semantisch äquivalentes Programm in der Zielsprache übersetzt.

9. ANLAGEN

1. Adminproramm Programmcode in C++
2. AVR Programmcode in Assembler
3. Eidesstattliche Erklärung
4. Datenblatt myAVR USB MKII
5. Datenblatt myAVR LCD Add-On
6. Datenblatt myAVR TWI Echtzeit Add-On
7. Datenblatt myAVR Digital Out Add-On
8. Datenblatt 10-to-4-Line Encoder SN74LS147N DIL
9. weitere Projektbilder
10. Rechnungen

Das Dattenblatt des ATmeg8 von Atmel und das Datenblatt des Spannungsreglers L78S09CV wurde auf dem mitgelieferten USB-Stick gespeichert.