

Lichtzeichenanlage

Projektabschlussbericht

Thema der Projektarbeit:	Planung und Erstellung von Fertigungsunterlagen für ein Unterrichtsmodell 'Lichtzeichenanlage' zum direkten Betrieb an den Parallelports eines Mikrocontrollers. Fertigung eines Prototypen."
Vollständiger Name:	Manfred Hayen und Matthias Titze
Datum der Abgabe:	07.05.2007 – 11.05.2007
Ausbildungsberuf:	Staatlich geprüfter Elektrotechniker Informations- und Kommunikationstechnik
Ausbildungsbetrieb	Berufsbildende Schulen Friedenstraße Wilhelmshaven Friedenstraße 60-62 26386 Wilhelmshaven

Letzte Änderung: 25.06.07, 10:28

Inhalt des Projektabschlussberichts

1 Deckblatt	1
2 Inhaltsverzeichnis	2
3 Aufgabenbeschreibung	4
4 Projektvorfeld	
4.1 Der „Ist-Zustand“	5
4.2 Aufgabenbeschreibung (Ziel)	6
4.3 Das Hauptproblem	6
5 Blockschaltbild	7
6 Schaltpläne	
6.1 Stromlaufplan	8
6.2 EAGLE Schaltplan, Ampel	9
6.3 EAGLE Schaltplan, Adapterplatine	10
7 Funktionsbeschreibung	
7.1 Grundsätzlicher Aufbau	11
7.2 EMES liefert High-Pegel	11
7.3 EMES liefert Low-Pegel	11
7.4 Funktionsweise der Taster	12
8 Aufbau Ampelgehäuse	
8.1 Frontplatte (schematisch)	13
8.2 Frontplatte-Bohrplan	14
8.3 Verdrahtung-Frontplatte	15
8.4 Innenaufbau (schematisch)	16
8.5 Innenaufbau Verdrahtung	17
8.6 Verdrahtung der Sub-D Buchse im Gehäuse	18
8.7 Seiten & Rückansicht (schematisch)	19
8.8 Rückwand-Bohrplan	20
8.9 Layout Ampelplatine, Oberseite & Unterseite	21
8.10 Bestückungsplan	22
9 Aufbau Adapterplatine	
9.1 Funktionsweise und Stromlaufplan	23
9.2 Bestückungsplan & Layout	24
10 GAL & Programmierung	
10.1 Funktionsweise GAL Torschaltung	25
10.2 Das Programm FGAL	26
10.3 Programmiergerät GALEP 4	27
10.4 Das Programmieren des GAL's	28
10.5 Das GAL-Programm	29

Inhalt des Projektabschlussberichts -Fortsetzung-

11 Aufgabenblätter	
11.1 Allgemeines	30
11.2 Zeitgeführte Steuerung	31
11.3 Zeit & Prozessgeführte Steuerung	32
12 EMES Software	
12.1 Das 8-Bit Programm	33
12.2 Das 16-Bit Programm	35
12.3 Das Delay Programm	38
13 Stückliste	
13.1 Fa. Reichelt Elektronik	39
13.2 Weitere Bauteile	40
14 Zusammenbau	
14.1 Verwendete Werkzeuge, Materialien und Software	41
14.2 Adapter und Ampel-Platine	42
14.3 Frontplatte	42
14.4 Gehäuse	43
15 Entscheidungen und ihre Gründe	44
16 Meilensteine	46
17 Fehlerbehebung	
17.1 Vorgehen bei der Problemlösung	47
17.2 Ein Fehler der gar keiner ist !	48
18 Technische Daten	49
19 Ansprechpartner	49
20 Abbildungsverzeichnis	50
21 Datenblattverzeichnis	51

3. Projekt TS-E 2 Schuljahr 2006/07 Aufgabenbeschreibung

Thema:

"Planung und Erstellung von Fertigungsunterlagen für ein Unterrichtsmodell 'Lichtzeichenanlage' zum direkten Betrieb an den Parallelports eines Mikrocontrollers. Fertigung eines Prototypen."

Vorgaben:

- **Zwei Betriebsarten** über einen Schalter wählbar
 - 8-Bit 8 Eingänge und Masse. Steuerung: Haupt-, Nebenrichtung, Fußgänger
 - 16-Bit 13 Eingänge, 3 Ausgänge 2x Fußgängertaster, 1x Fahrbahntaster und Masse.
- Steuerung: alle Ampeln separat (Portbelegungen sind dem Team bekannt)
- **Verbindung zu Mikrocontrollerboards** mit einer universellen handelsüblichen Verbindungsleitung (Adapter am Controllerboard im Eigenbau)
- **Gehäuseeinbau**
- **Internes Netzteil** (Anschluss an 240V AC)
- **Funktionen gut sichtbar** bei Sonnenlicht und Kunstbeleuchtung
- **Nur professionell** hergestellte Platinen innerhalb des Gerätes
- **Reparaturfreundlicher** Aufbau
- **Programme** zur Vorführung jeder Betriebsart (im EPROM, für den EMUF)

Zur Dokumentation:

- Zwei Exemplare, gedruckt (Ordner, Loseblatt)
- 2CD's mit: (*wird erst nach der Präsentation abgegeben*)
 - kompletter Dokumentation im 'Open Document Format' oder 'PDF'
 - CAD-Dateien
 - kleiner WEB-Präsentation (nur relative Links verwenden)
 - druckfertigen Aufgabenblättern zu jeder Betriebsart (PDF)
 - Präsentation

4. Projektvorfeld

Einleitend sei gesagt, dass sämtliche Begriffe, die in eckigen Klammern [...] geschrieben sind, am Ende dieses Dokuments ausführlich erklärt werden.

4.1 Der „Ist-Zustand“

Die Lichtzeichenanlagen, die zurzeit an der Berufsbildenden Schule Friedenstraße in Wilhelmshaven genutzt werden, stammen aus den 80er Jahren und sind dementsprechend veraltet.

Die Lichtzeichenanlagen werden durch einen Mikrocomputer angesteuert. Hier gibt es zwei Möglichkeiten der Ansteuerung. Die eine Möglichkeit ist eine rein zeitgeführte (8-Bit) Ansteuerung, die andere Möglichkeit ist eine prozessgeführte (16-Bit) Ansteuerung. Die zeitgeführte Steuerung wird mit Hilfe eines 8poligen Flachbandkabels und eines kleinen Adaptersteckers, der auf eine IC-Fassung gesteckt wird, angesteuert. Für die prozessgeführte Steuerung wird einfach der Adapterstecker abgenommen und ein zusätzliches Flachbandkabel angeschlossen.

Die Lichtzeichenanlagen verfügen über kein eigenes Netzteil und belasten dadurch das Netzteil des Mikrocomputers. Die Lichtzeichenanlagen besitzen für den Fahrbahnkontakt ein Reedrelais. Um dieses Reedrelais auszulösen ist immer ein Magnet notwendig.

Da die Adapterstecker recht klein und häufig in Benutzung sind, sind viele verloren oder kaputt gegangen.

Durch häufiges Einstecken des zusätzlichen Flachbandkabels sind die Anschlussbuchsen dementsprechend verschlissen und es kommt dadurch zu Wackelkontakten. Die fest angebrachten Flachbandkabel weisen deutliche Gebrauchsspuren und gelegentliche Kabelbrüche auf.

Von den zehn vorhandenen Lichtzeichenanlagen funktionieren nur zwei fehlerfrei.

4.2 Aufgabenstellung (Ziel)

- Die neue Lichtzeichenanlage soll in einem Gehäuse untergebracht werden.
- Die beiden Betriebsarten sollen mit Hilfe eines Schalters wählbar sein.
- Die Verbindung zum Mikrocontrollerboard soll mit einer universellen handelsüblichen Verbindungsleitung hergestellt werden.
- Die Lichtzeichenanlage soll außerdem ein eigenes Netzteil besitzen.
- Die Funktionen sollen bei Sonnenlicht, aber auch bei Kunstbeleuchtung gut sichtbar sein.
- Ein reparaturfreundlicher Aufbau sowie zwei Programme zur Vorführung werden gefordert.

4.3 Das Hauptproblem

Das Hauptproblem ist es, die in Abb. 4.3 abgebildete Schaltung in elektronischer Form umzusetzen. Wir haben uns für ein programmierbares Bauteil, einem GAL (Gate Array Logik) entschieden. Im Abschnitt 10 liefern wir unsere Lösung.

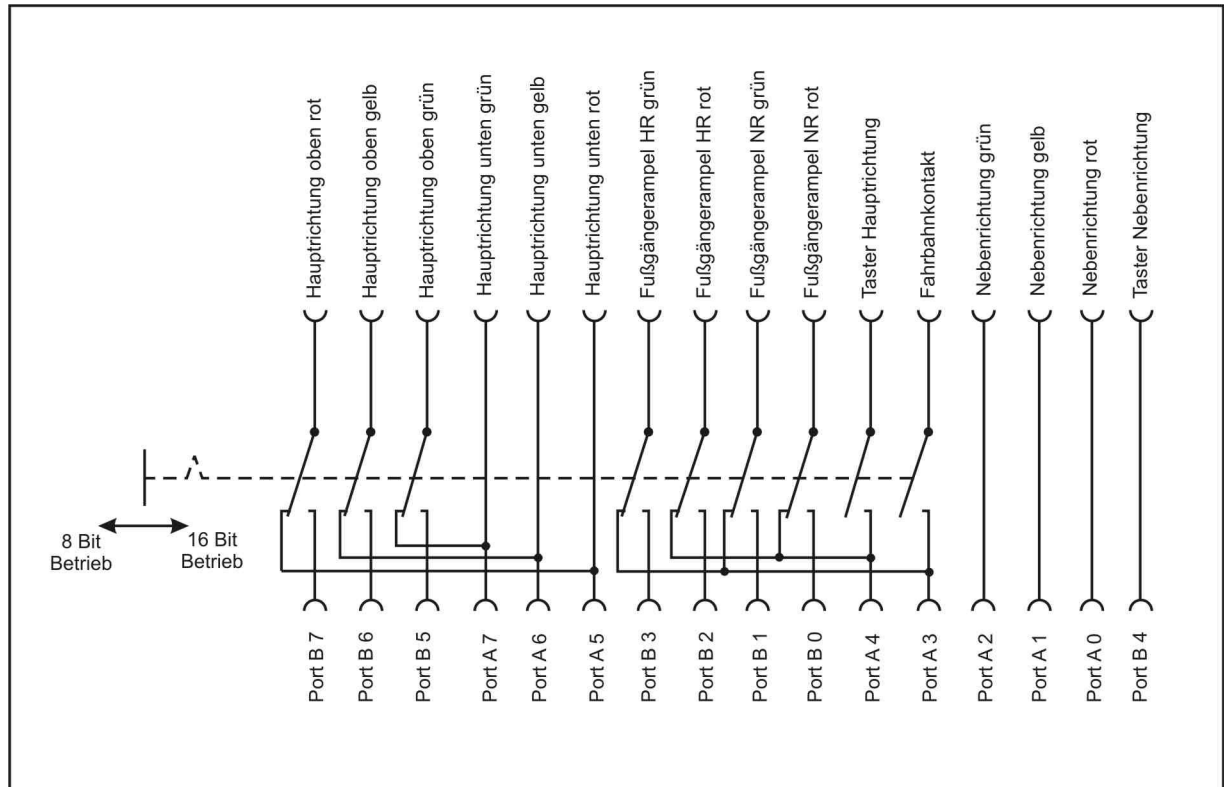


Abb. 4.3

5. Blockschaltbild

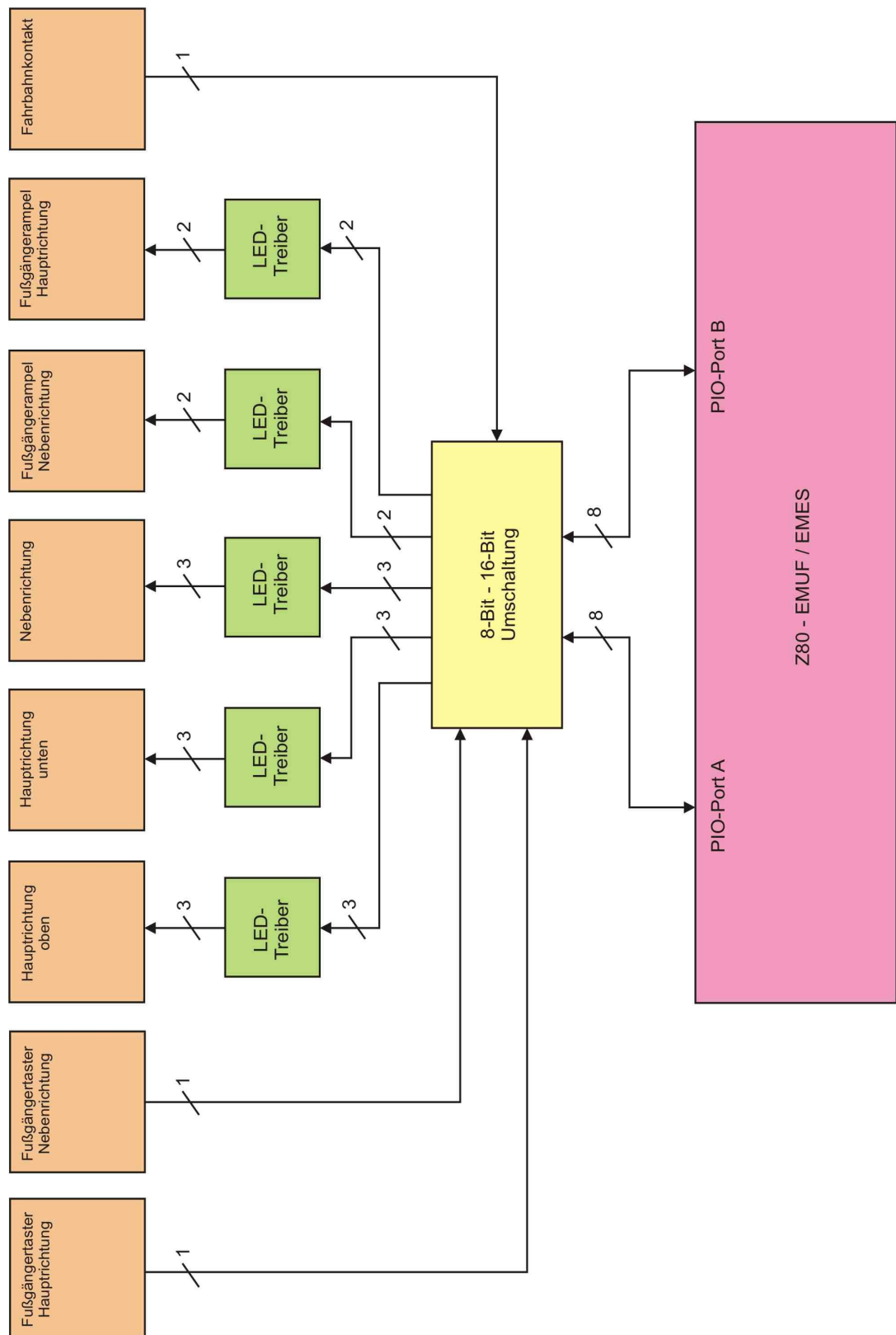


Abb. 5.1

6.1 Stromlaufplan

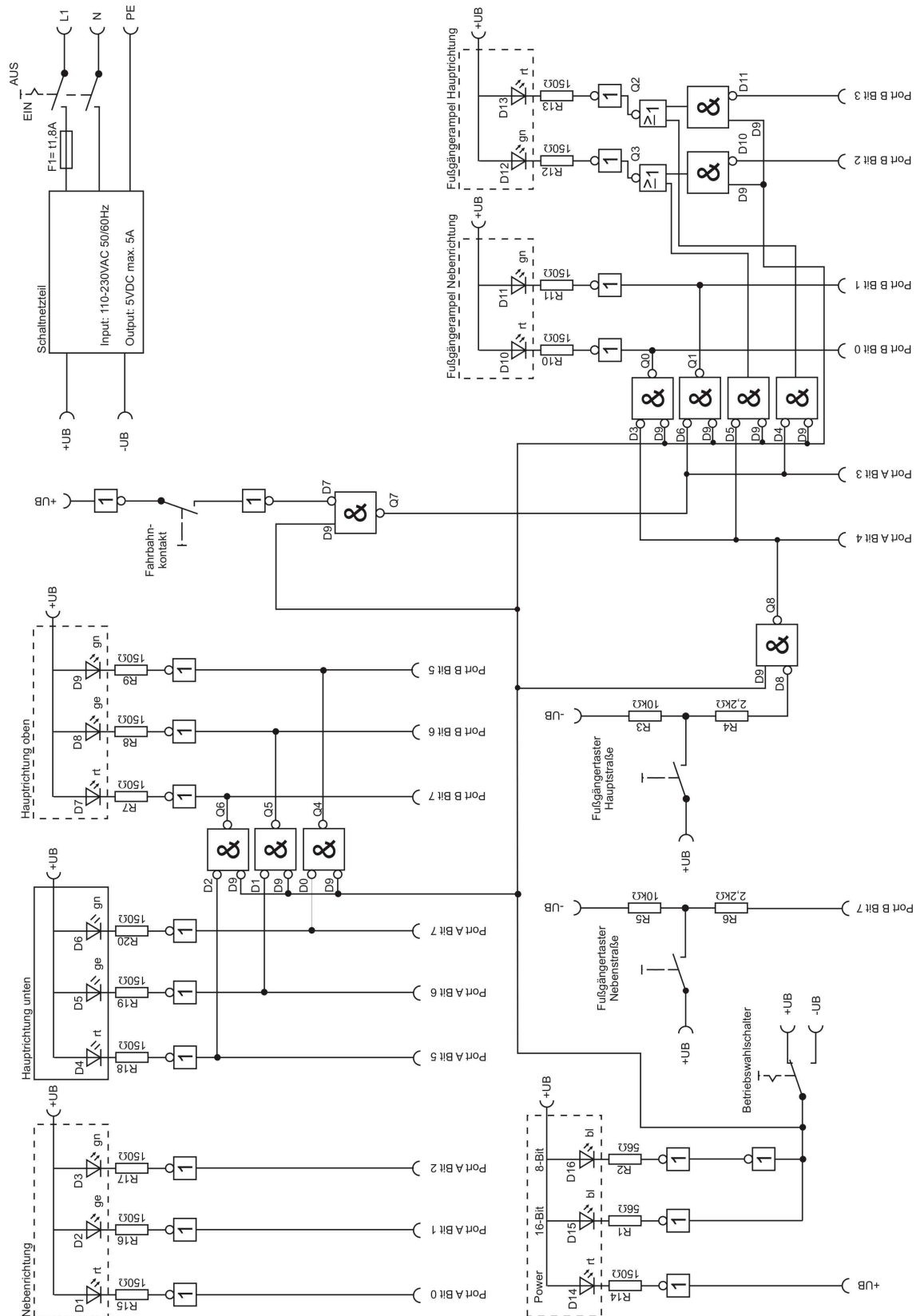


Abb. 6.1

6.3 EAGLE Schaltplan Adapterplatine

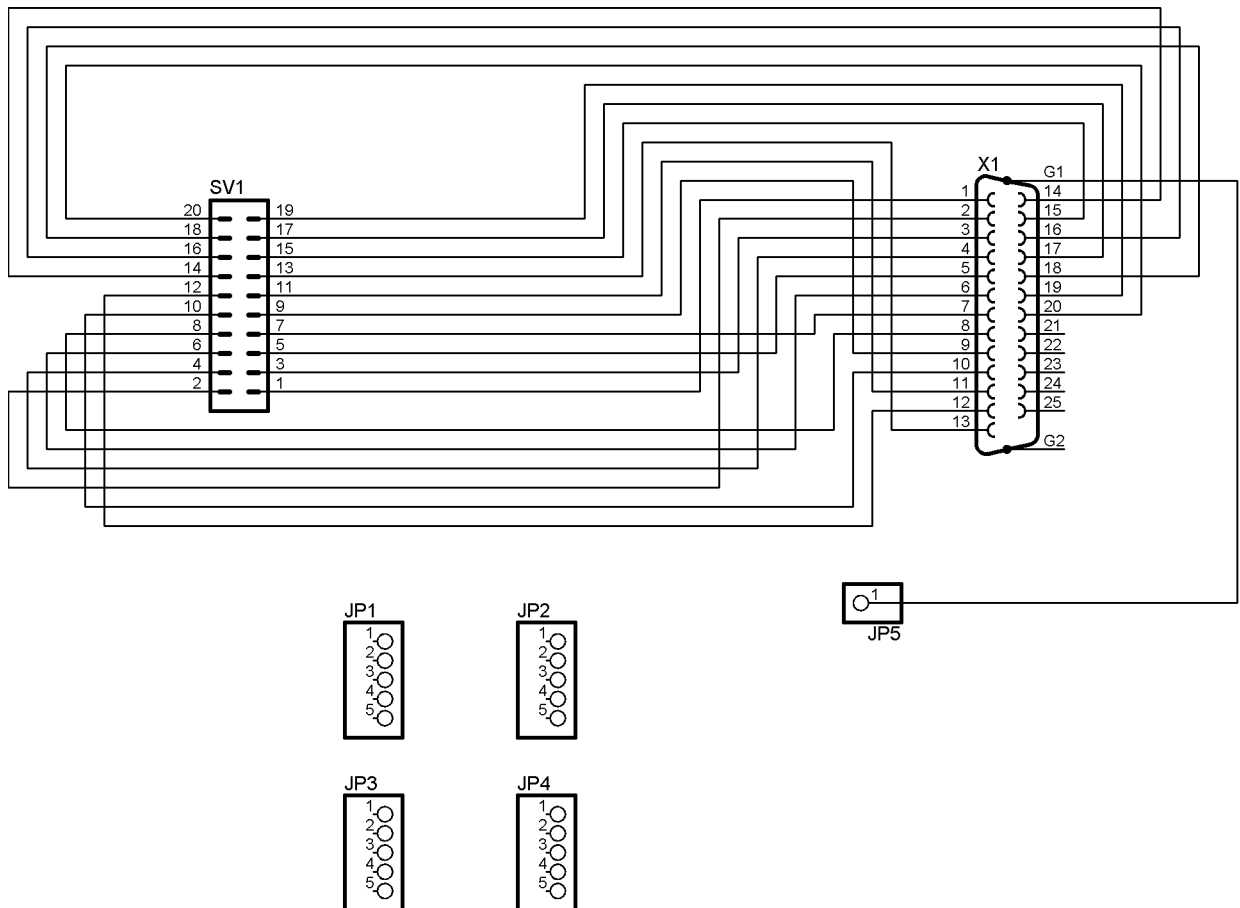


Abb. 6.3

7. Funktionsweise

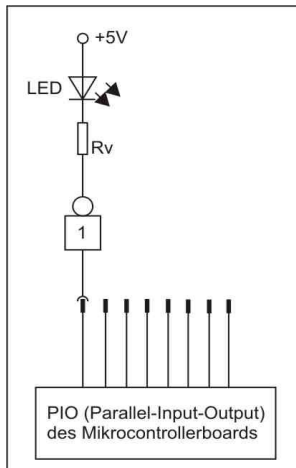


Abb. 7.1

7.1 Grundsätzlicher Aufbau

Die grundsätzliche Funktionsweise zur Ansteuerung einer Leuchtdiode (LED) ist relativ einfach gelöst.

Wie in Abb. 7.1 zu sehen, wird die Betriebsspannung von +5 Volt direkt auf die Anode der LED gelegt. Über einen Vorwiderstand R_v , der den Strom (I) auf 20mA begrenzt, wird die LED an den Ausgang eines Invertierers gegeben. Der Eingang des Invertierers liegt direkt am Parallelausgang des Mikrocontrollerboards.

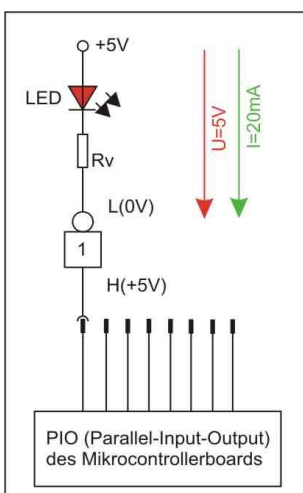


Abb. 7.2

7.2 Das Mikrocontrollerboard liefert High-Pegel

Abbildung 7.2 zeigt den Fall eines High-Pegels (H) am Ausgang des Mikrocontrollerboards. Das Mikrocontrollerboard liefert ein High-Pegel an den Eingang des Invertierers. Der Invertierer wandelt den High-Pegel am Eingang in einen Low-Pegel am Ausgang des Invertierers. Dieser Low-Pegel (L) sorgt dafür, dass ein Potentialunterschied zwischen der Betriebsspannung und dem Low-Pegel von $U = 5$ Volt entsteht. Dadurch kann nun ein Strom fließen. Die Leuchtdiode leuchtet.

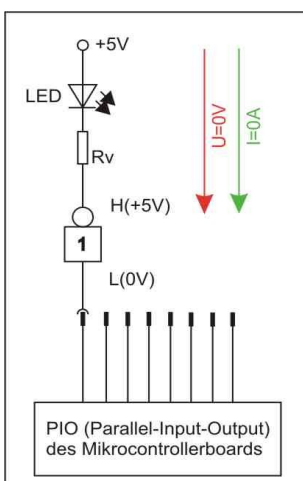


Abb. 7.3

7.3 Das Mikrocontrollerboard liefert Low-Pegel

Abbildung 7.3 zeigt den Fall eines Low-Pegels am Ausgang des Mikrocontrollerboards.

Das Mikrocontrollerboard liefert ein Low-Pegel an den Eingang des Invertierers. Der Invertierer wandelt den Low-Pegel am Eingang in einen High-Pegel am Ausgang des Invertierers. Dieser High-Pegel sorgt dafür, dass kein Potentialunterschied zwischen der Betriebsspannung und dem High-Pegel entsteht. Dadurch leuchtet keine LED.

7.4 Die Funktionsweise der Taster

Abb. 7.4 zeigt die Schaltungen der Fußgängertaster für die Haupttrichtung und die Nebenrichtung sowie den Fahrbahnkontakt. Die Schaltungen sind beide einfach gehalten:

Fußgängertaster Haupt-, und Nebenrichtung:

Wird der Taster geschlossen so fällt eine Spannung von 5 Volt über dem $10\text{k}\Omega$ Widerstand ab. Der $2,2\text{k}\Omega$ Widerstand dient lediglich der Strombegrenzung für die nachfolgende Schaltung im EMES/EMUF.

Fahrbahnkontakt:

Der Eingang des ersten Inverters liegt direkt an der Betriebsspannung und liefert dadurch am Ausgang einen Low-Pegel. Wird der Kontakt durch Druck auf den Taster geschlossen, so liegt am Eingang des zweiten Inverters ein Low-Pegel, der dann am Ausgang einen High-Pegel liefert.

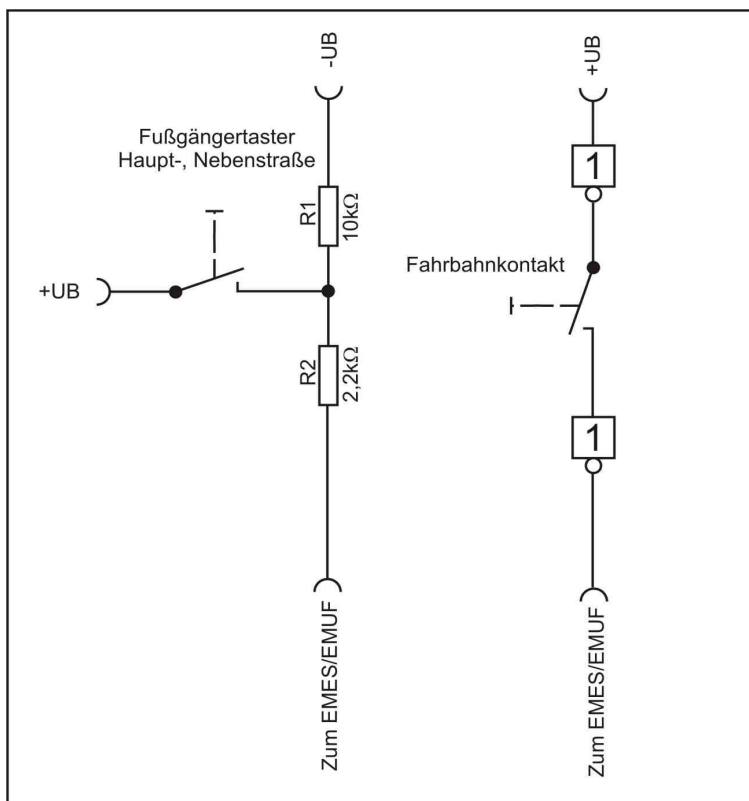


Abb. 7.4

8. Aufbau Ampelgehäuse

8.1 Frontplatte, schematisch

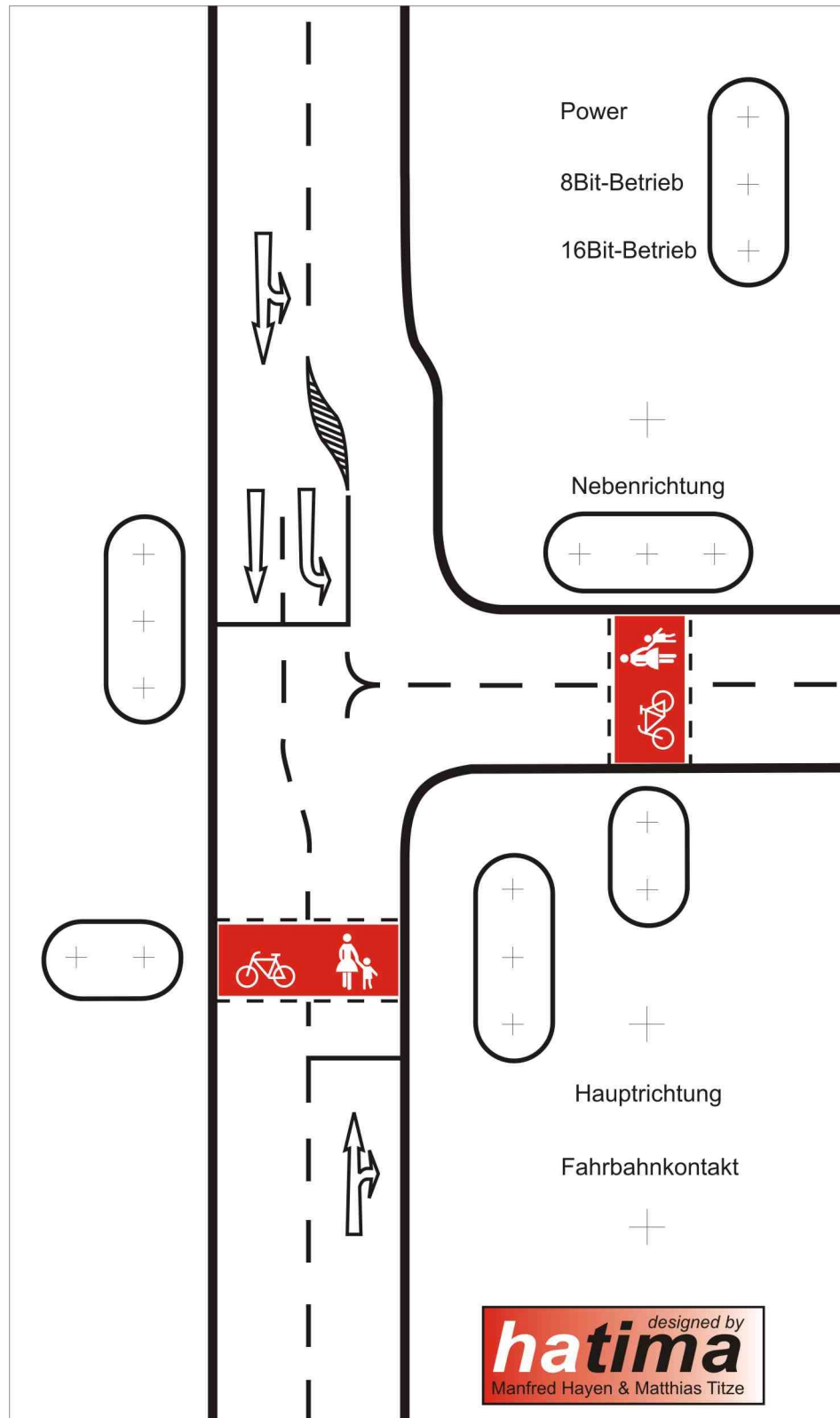


Abb. 8.1

8.2 Bohrplan Frontplatte

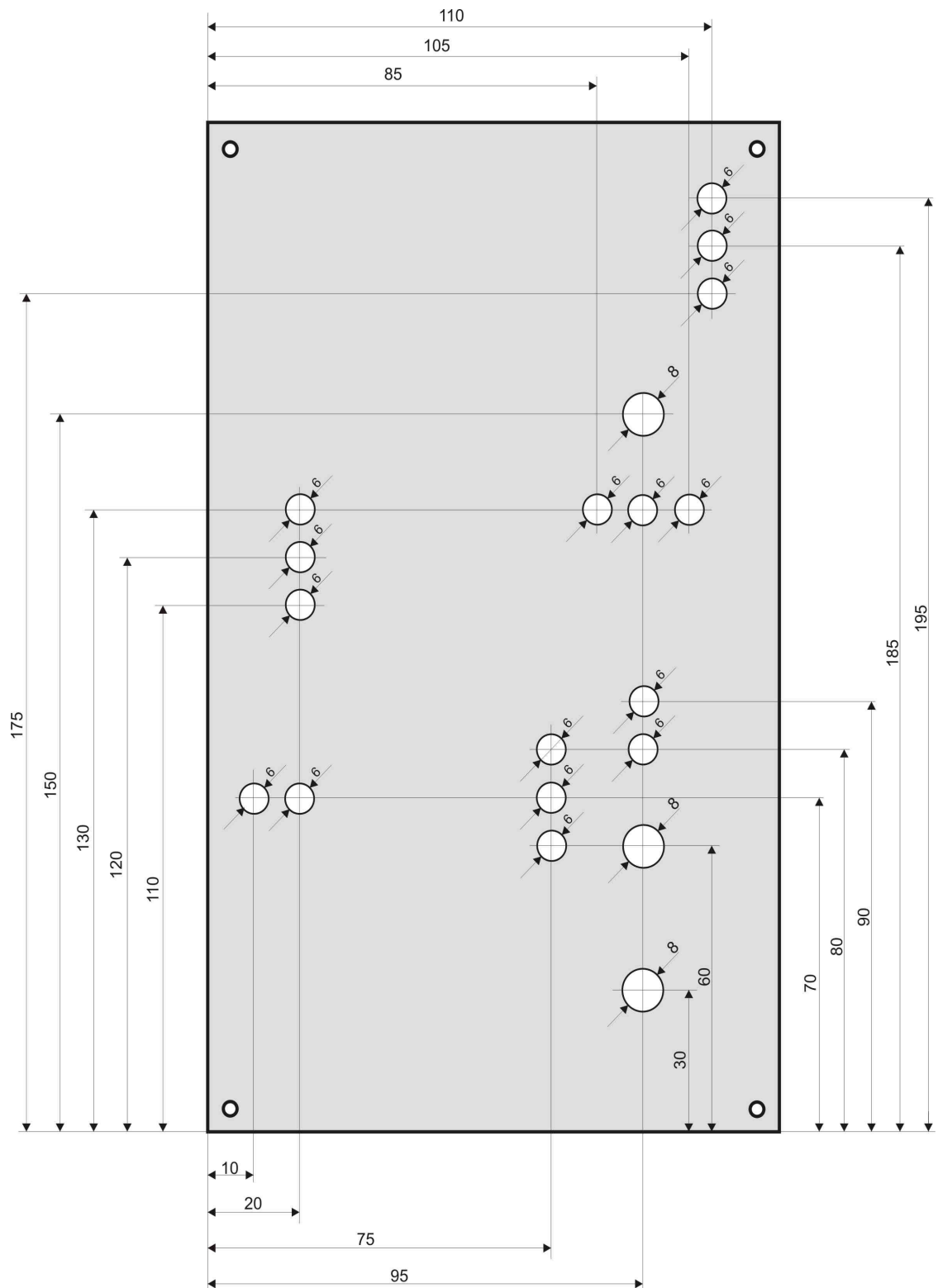


Abb. 8.2

8.3 Verdrahtung Frontplatte

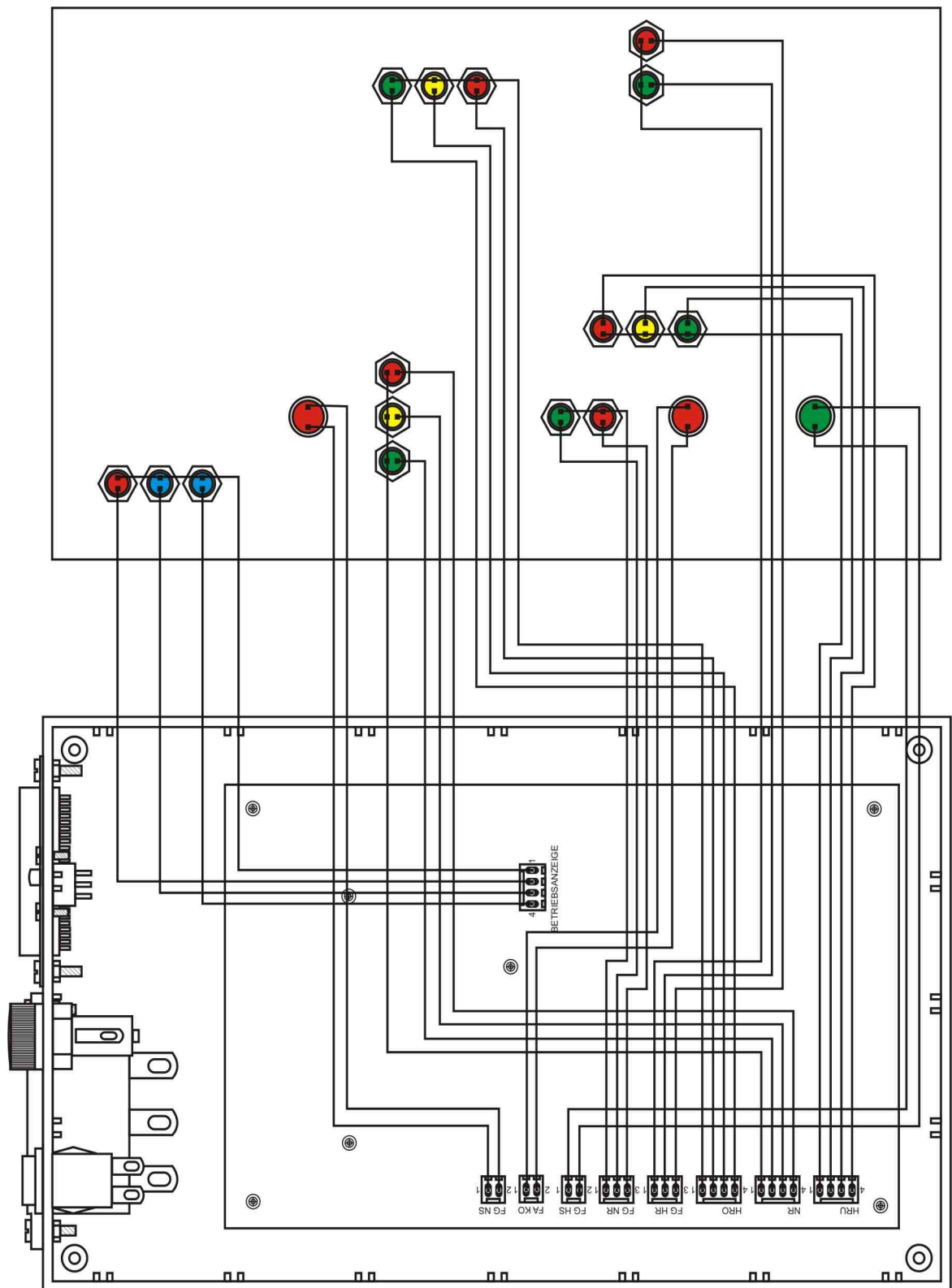


Abb. 8.3

8.4 Der Innenaufbau (schematisch)

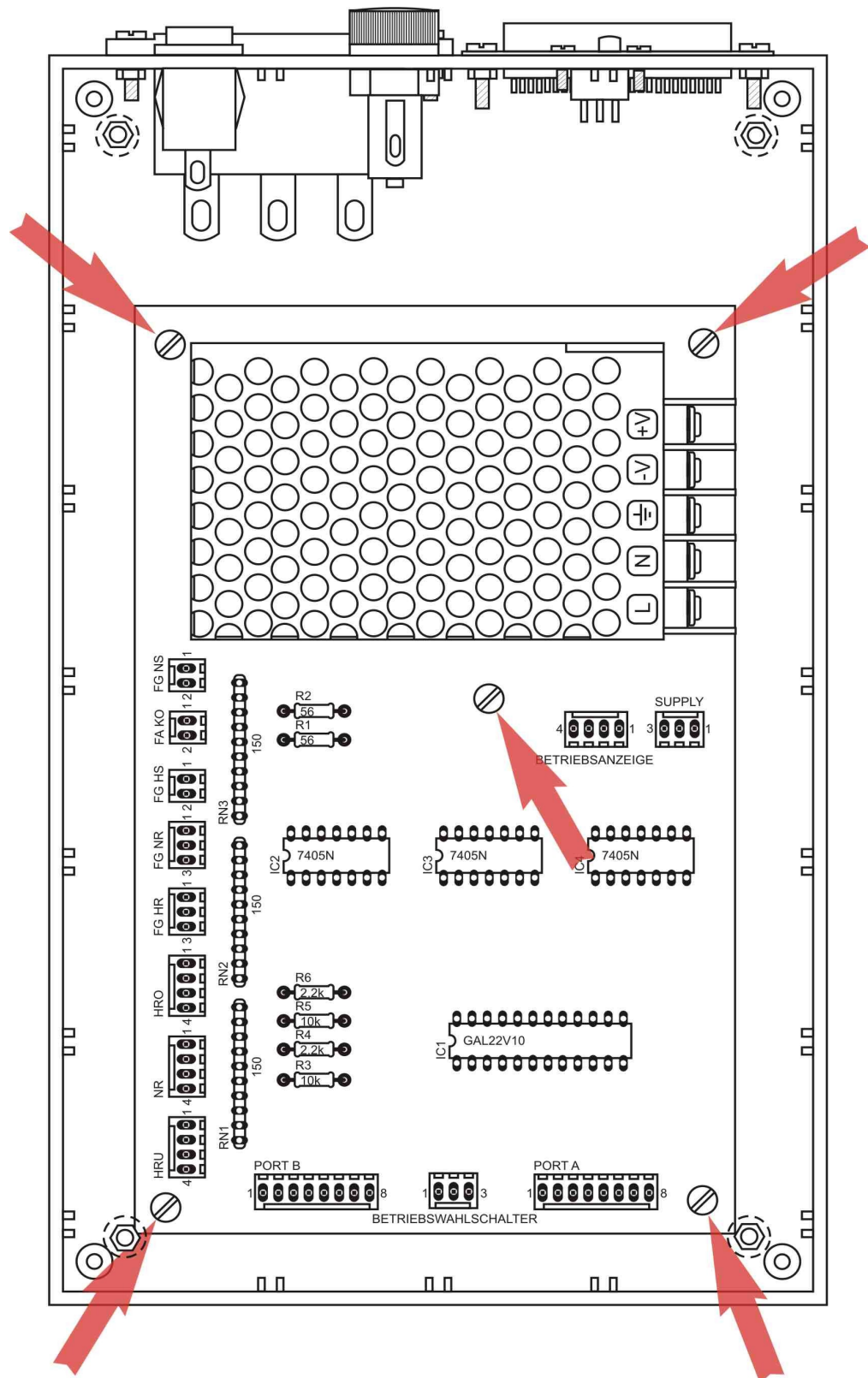


Abb. 8.4

8.5 Innenaufbau, Verdrahtung

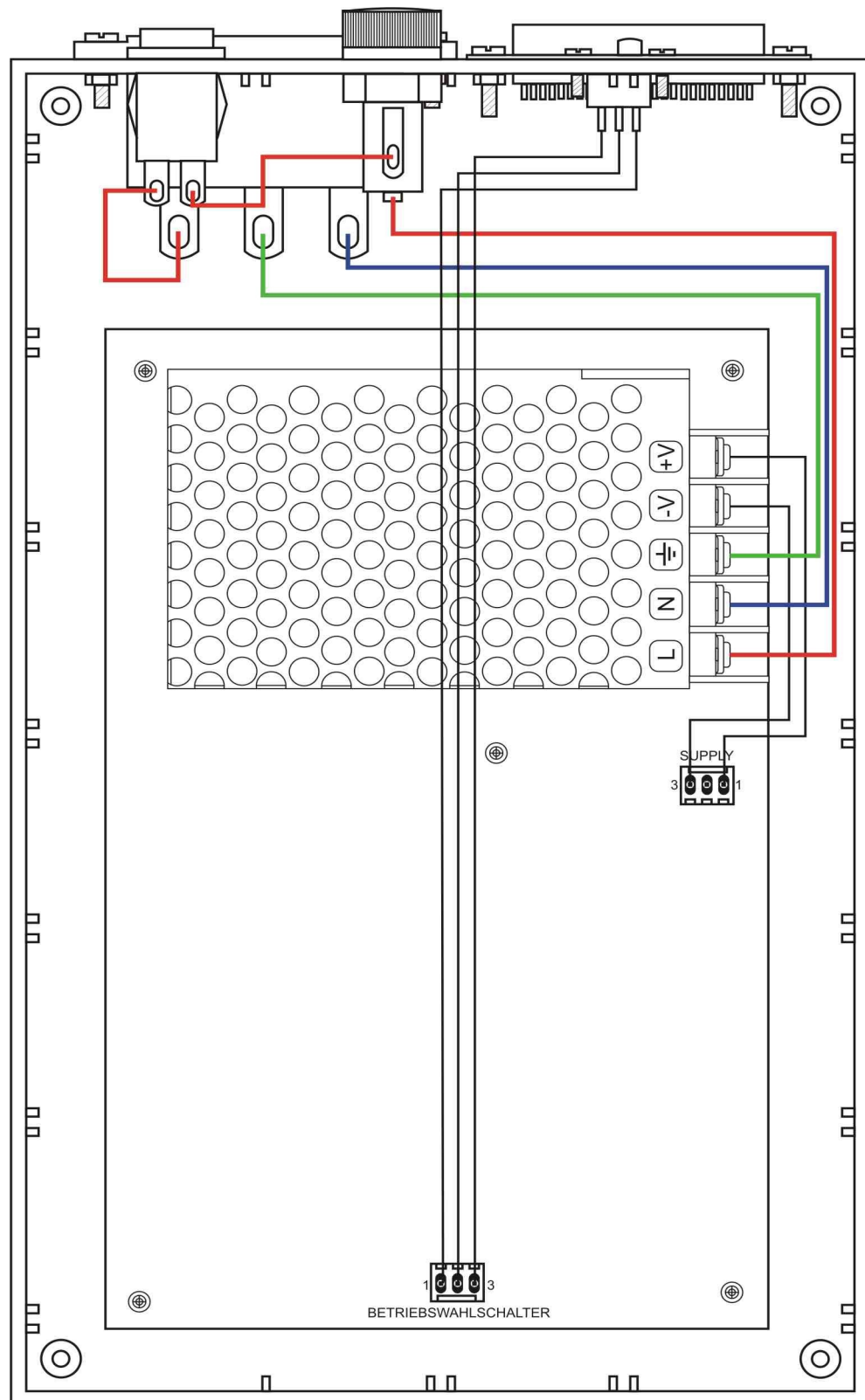


Abb. 8.5

8.6 Verdrahtung der 25poligen Sub-D Buchse im Gehäuse

Da die Projektvorgabe eine universelle, handelsübliche Verbindungsleitung vom Mikrocontrollerboard zur Lichtzeichenanlage vorsah, haben wir uns für eine 25polige Sub-D-Leitung entschieden. Die passende Buchse, die im Gehäuse eingebaut wurde, muss nun mit folgender Belegung mit den 8poligen Platinensteckverbindern (Port A und Port B) verlötet werden.

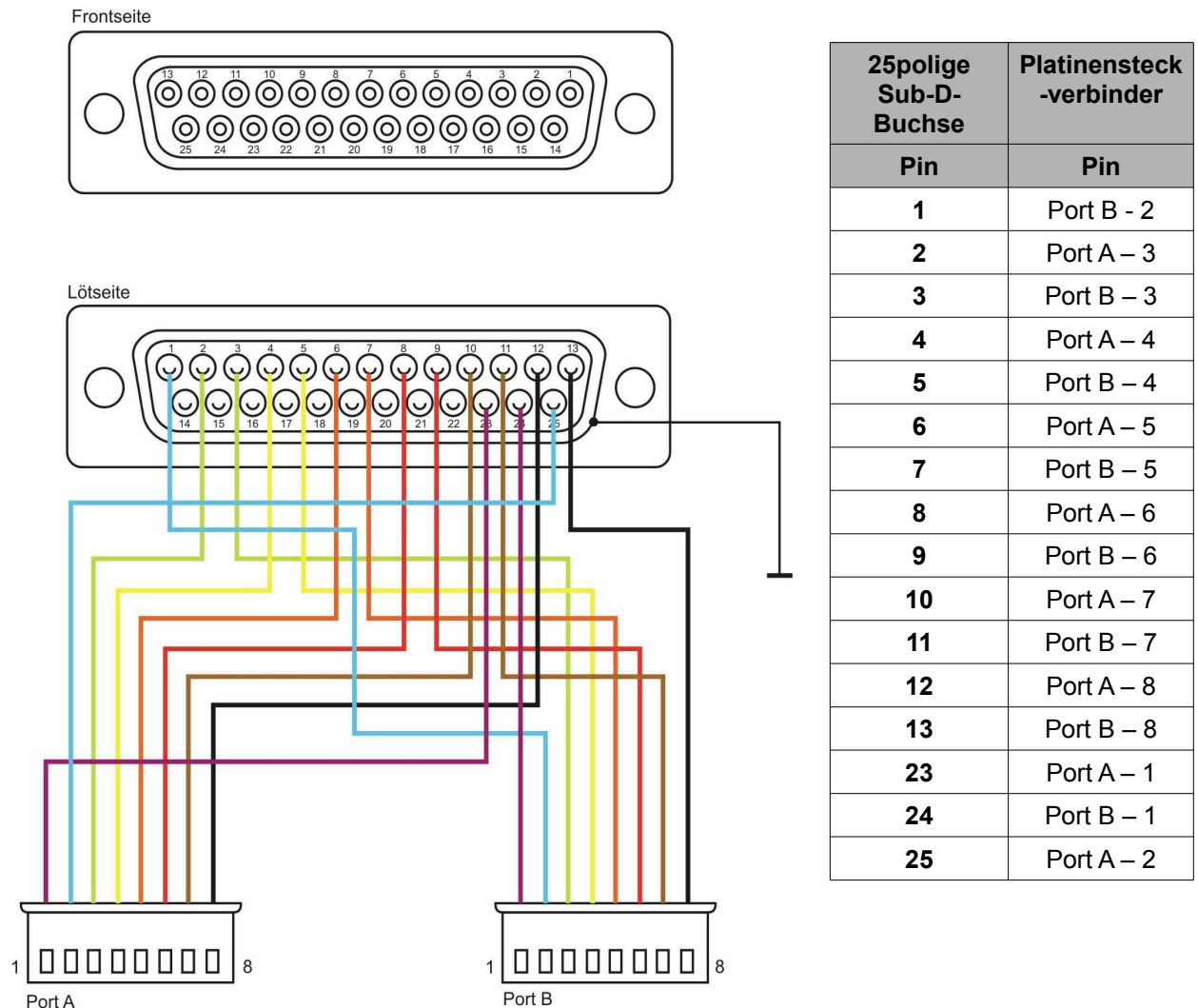


Abb. 8.6

Am Gehäuse der Sub-D-Buchse wird eine zusätzliche Leitung angelötet, die dann mit der Schaltungsmasse verbunden wird. Diese Masse wird über die Kabelschirmung der Sub-D-Leitung mit der Masse des EMES/EMUF verbunden. Dies haben wir so gemacht, damit die Signale beiderseitig den gleichen Bezugspunkt haben.

8.7 Seitenansicht

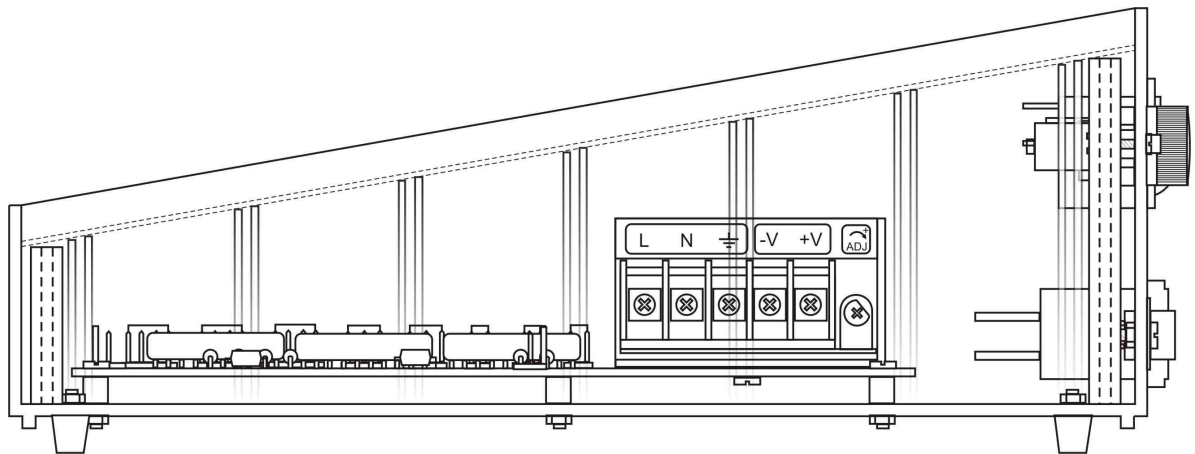


Abb. 8.7

Rückansicht

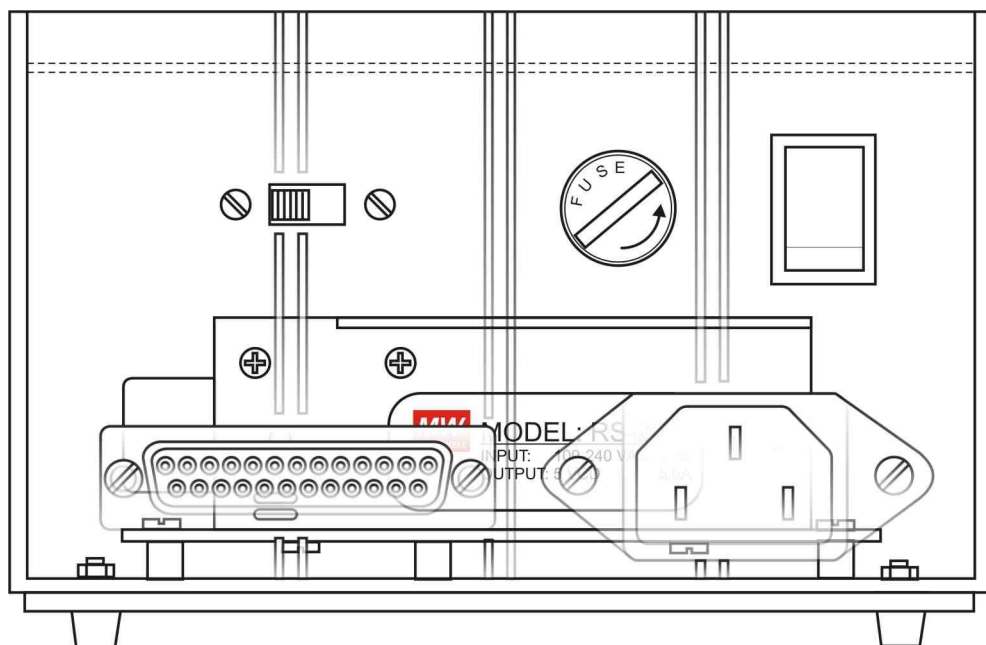


Abb. 8.8

8.8 Bohrplan Gehäuserückwand

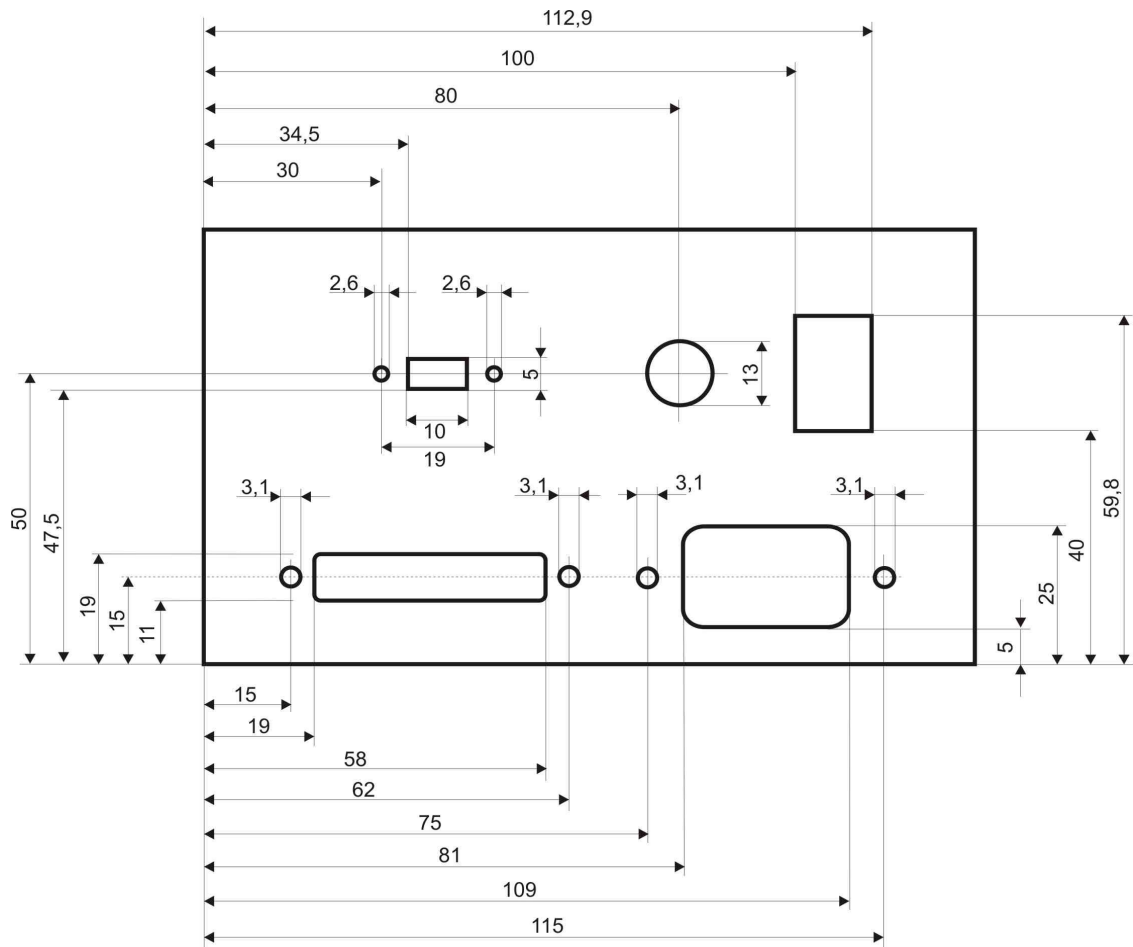


Abb. 8.9

8.9 Layouts

Ampelplatine Oberseite:

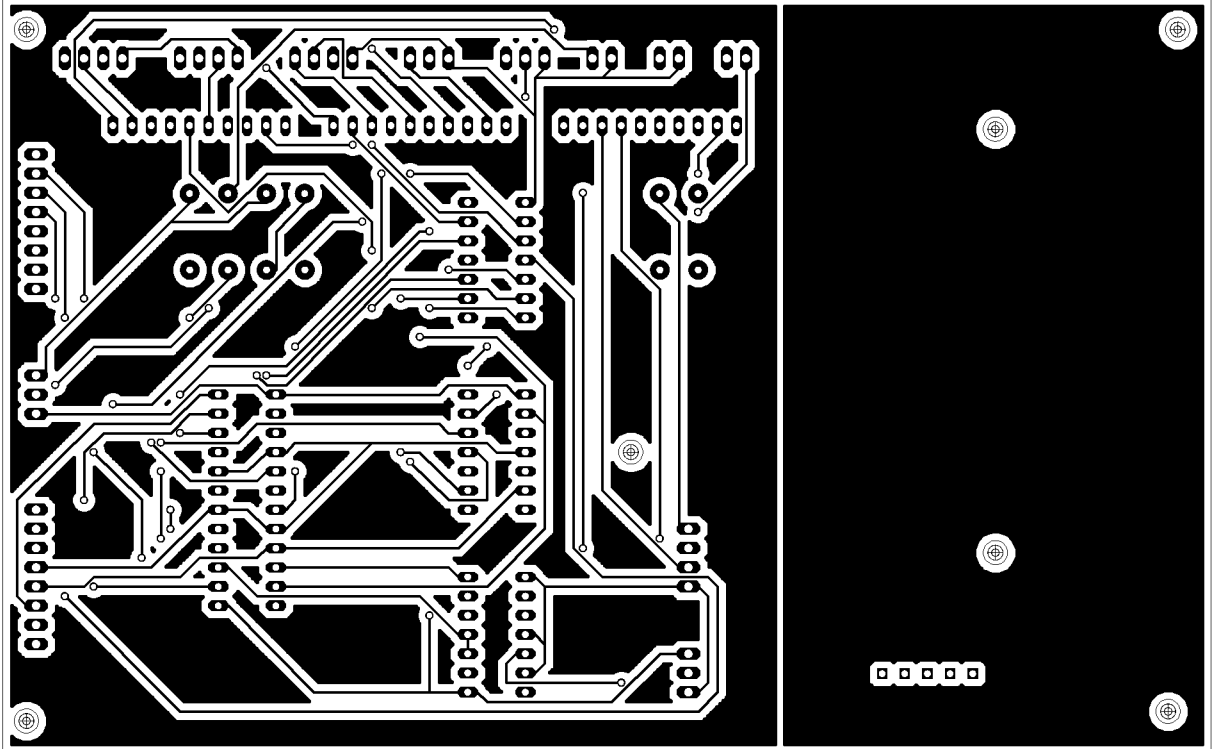


Abb. 8.10

Ampelplatine Unterseite:

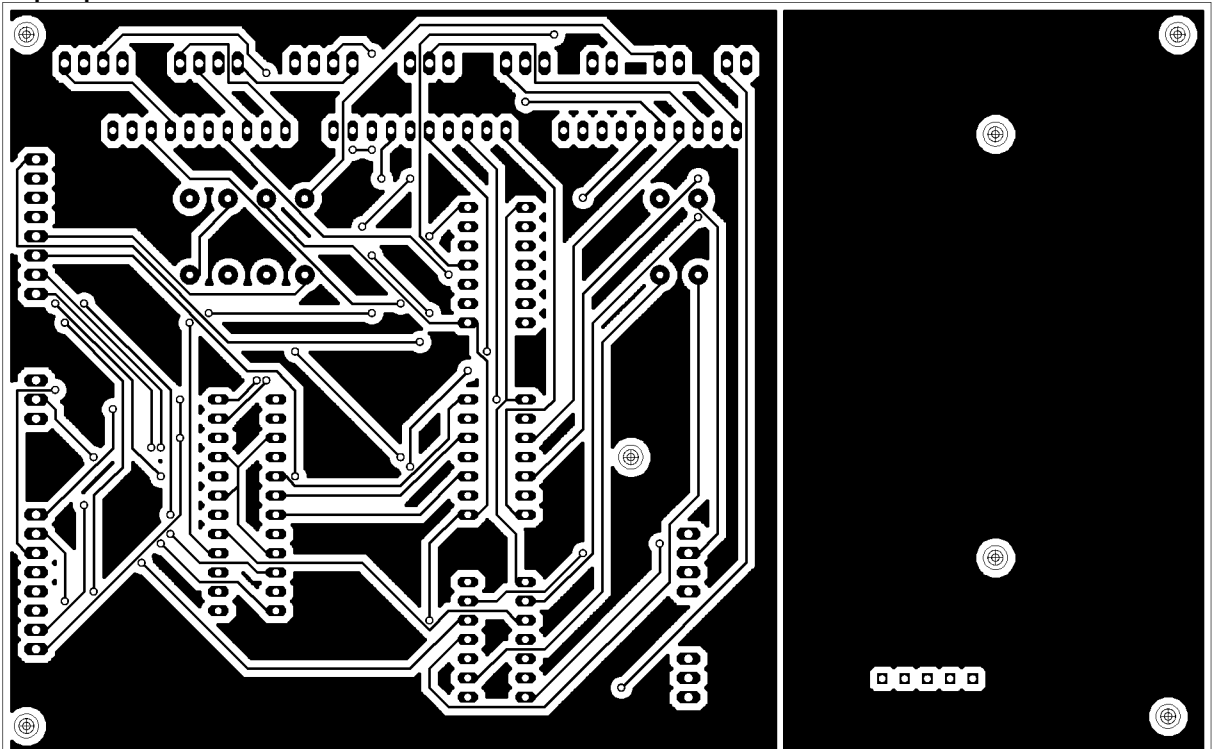


Abb. 8.11

8.10 Bestückungsplan

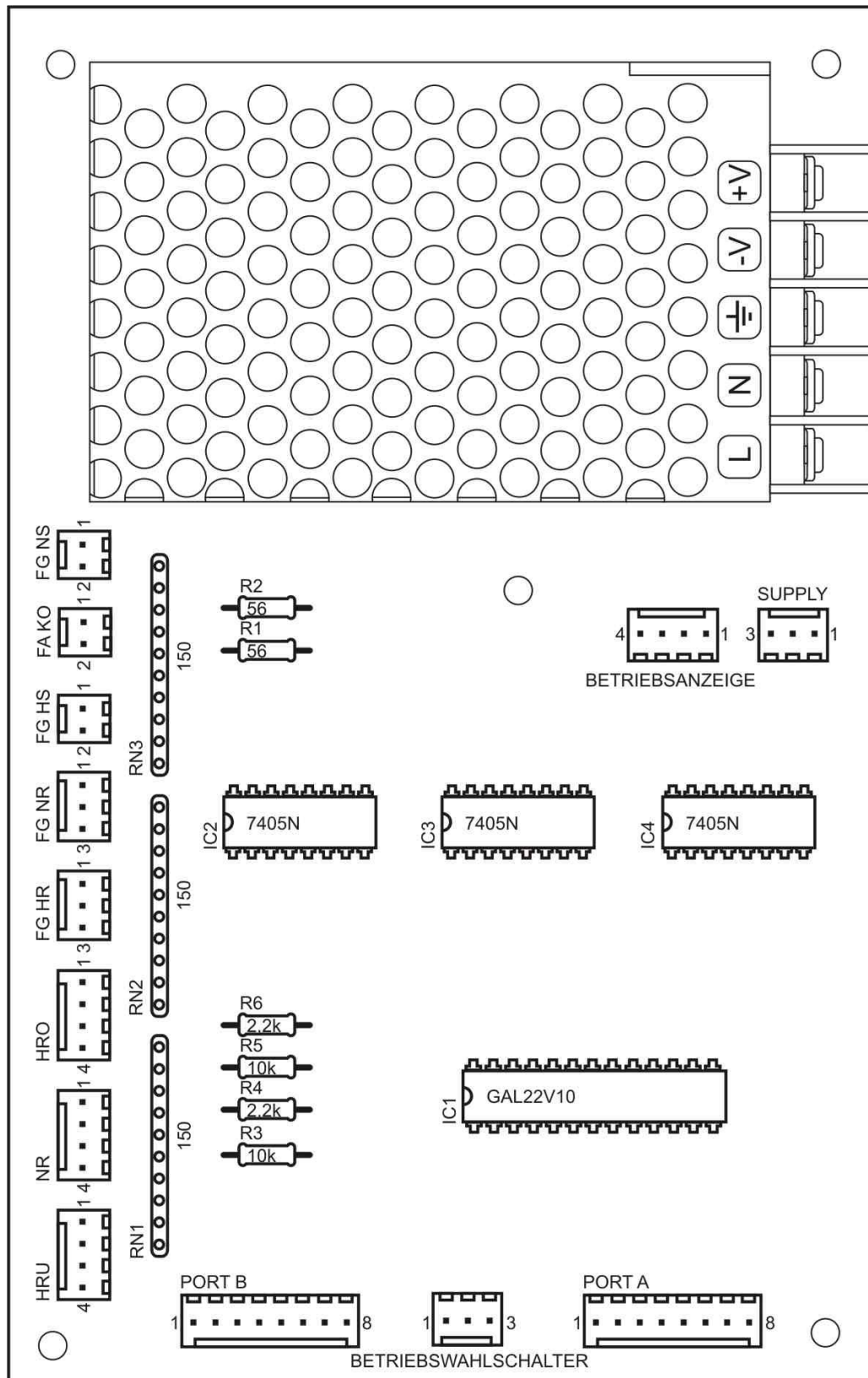


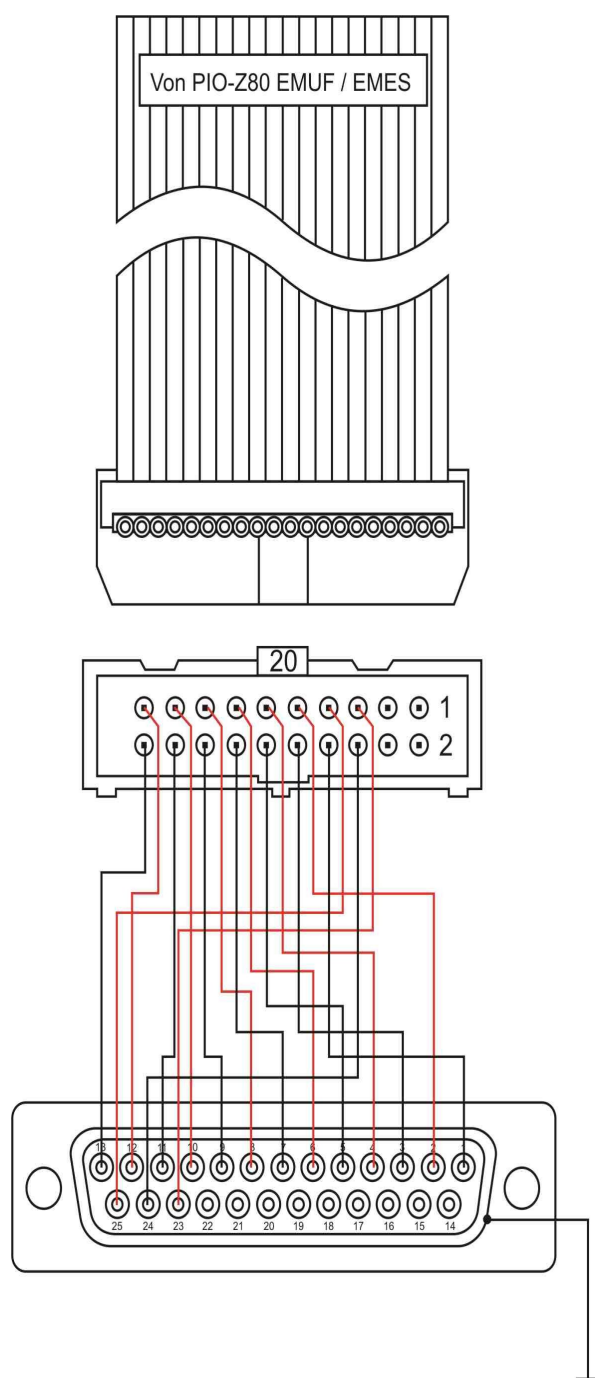
Abb. 8.12

9. Aufbau Adapterplatine

9.1 Funktionsweise und Stromlaufplan

Eine weitere Forderung an das Projekt war eine universelle handelsübliche Verbindungsleitung. Um diese Leitung, in diesem Fall eine 25polige Sub-D-Leitung (ein einfaches Verlängerungskabel für Drucker), mit dem EMUF / EMES zu verbinden ist eine Adapterplatine notwendig.

Diese Adapterplatine haben wir wie folgt realisiert.



Pinbelegung:

Pin-Nr.(PIO)	Signalname	Sub-D Pin
1	ARDY	nicht belegt
2	BRDY	nicht belegt
3	ASTB	nicht belegt
4	BSTB	nicht belegt
5	Port A Bit 0	23
6	Port B Bit 0	24
7	Port A Bit 1	25
8	Port B Bit 1	1
9	Port A Bit 2	2
10	Port B Bit 2	3
11	Port A Bit 3	4
12	Port B Bit 3	5
13	Port A Bit 4	6
14	Port B Bit 4	7
15	Port A Bit 5	8
16	Port B Bit 5	9
17	Port A Bit 6	10
18	Port B Bit 6	11
19	Port A Bit 7	12
20	Port B Bit 7	13

Abb. 9.1

9.2 Aufbau Adapterplatine und Layout

Bestückungsplan

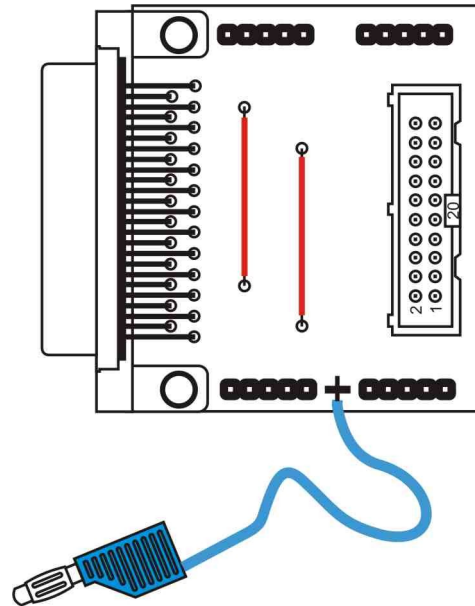


Abb.9.2

Layout

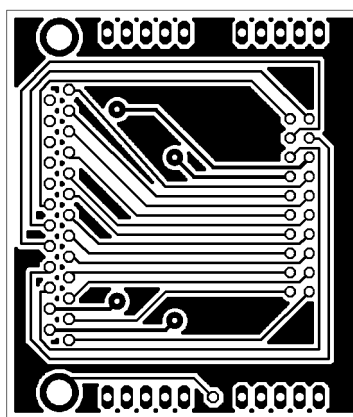


Abb. 9.3

10. GAL und Programmierung

10.1 Funktionsweise GAL Torschaltung

Der GAL wurde als Torschaltung programmiert. Eine Torschaltung ist eine Schaltung, die nur bei gegebener Erlaubnis einen Wert am Ausgang erzeugt, der vom Eingang abhängig ist. Dabei wird eine "UND"-Verknüpfung von dem Eingangssignal und dem Erlaubnis-Signal vorgenommen.

Beispiele für eine Torschaltung:

1. Ausgänge gesperrt

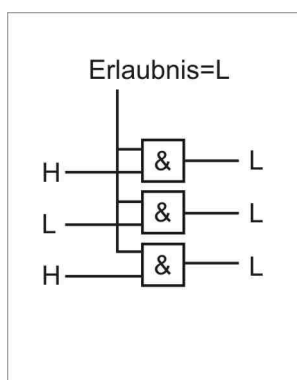


Abb. 10.1

2. Ausgänge geöffnet

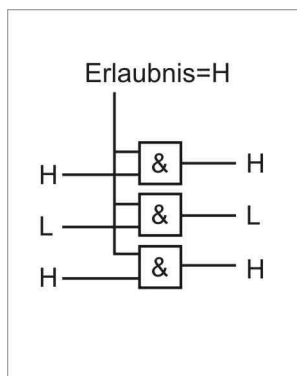


Abb.10.2

Der GAL22V10 besitzt zudem die Möglichkeit der Tristate-Beschaltung. Tristate bedeutet, dass ein Ausgang drei Zustände annehmen kann:

1. High-Pegel
2. Low-Pegel
3. Hochohmig

Dieser dritte Zustand ist für unseren Zweck von außerordentlicher Wichtigkeit, da wir ja quasi einen Schalter programmiert haben. Dieser Schalter unterbricht die Leitung und setzt sie nicht auf einen Low-Pegel (Masse) wie es bei einer normalen "Torschaltung" der Fall gewesen wäre.

10.2 Das Programm FGAL

FGAL ist ein einfacher GAL-Assembler für die GAL Typen 16v8, 20v8 und den hier verwendeten 22v10.

Es erstellt sog. PLD-Dateien (Programmable Logic Device) welche in PALASM2 kompatiblen Zeichensatz und anschließend im JEDEC-Dateiformat gespeichert werden.

Diese JEDEC-Datei (z.B. Ampel.jed) zu erstellen ist zwingend notwendig, da der GAL nicht in der Lage ist Klartextbefehle umzusetzen.

Zum erstellen der Datei sind nur wenige Schritte notwendig:

- Erstellen des Programms in einem beliebigem Texteditor, jedoch mit der Dateinamenserweiterung ".gal" im Ordner des Programms FGAL abspeichern. (z.B. Ampel.gal)
- Öffnen der Eingabeaufforderung (Start-Ausführen-cmd)
- Wechsel in das Programmverzeichnis von FGAL
- Mit dem Befehl: "fgal Ampel.gal" wird diese Datei assembliert und als Ampel.jed im Arbeitsverzeichnis von FGAL abgespeichert.

Das GAL Programm ist erstellt und kann nun mit Hilfe eines GAL - Programmiergeräts übertragen werden.

Beispiel für eine *.GAL-Datei

Title FGAL Testdatei für GAL22V10
Pattern Testdatei
Revision 0.01
Author hatima
Date © 2007

CHIP COUNTER GAL22V10 ; **Bauteildefinition hier: GAL22V10**

; Pin definitions ; **Pinbelegung des GAL's**

[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
CLK	-	-	CLR	LOAD	D0	D1	D2	D3	AR	SP	GND
[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]
-	-	CO	Q3	Q2	Q1	Q0	CI	CEN	-	-	VCC

EQUATIONS ; **Funktionsgleichungen (/=Nicht;*=UND;+=ODER)**

Q0 := D1+D2*D3
 Q1 := /CLR*D3
 Q2 := D2+D3
 Q3 := LOAD*D1

10.3 Programmiergerät Galep4

GALEP-4 ist ein professionelles Universal-Programmiergerät im Pocketformat. Ein eigens entwickeltes Pintreiber ASIC bietet eine Vielseitigkeit, die man sonst nur bei teuren High-End Geräten findet:

GALEP programmiert ohne zusätzliche Adapter oder Zusatzschaltungen EPROMs, EEPROMs, FLASH-EPROMs, serielle EEPROMs, Mikrocontroller, GALs und PALCEs. Low-Voltage Bauteile werden bis 1,3 Volt unterstützt.

GALEP ist nicht viel größer als eine PC-Maus (115x80x33mm). Falls man das Gerät ohne Netzteil betreibt, kann man mit einer Akkuladung bis zu 8 Stunden programmieren - damit ist das Gerät auch bestens für den mobilen Einsatz geeignet. Es wird einfach mit dem Druckerport eines PCs oder Laptops verbunden. Neben Grundfunktionen wie Auslesen, Programmieren, Vergleichen und Löschen bietet die Software komfortable Einstellmöglichkeiten für die Sonderfunktionen des ausgewählten Bauteils. Der Editor erlaubt vielfältige Bearbeitungsmöglichkeiten der Bauteil-Inhalte, die als Datei im Binär-, Intel-Hex-, Motorola-S- oder Jedec-Format geladen oder gespeichert werden können.

Neue Bauteile werden kostenlos per Internet Download in die Software aufgenommen - damit ist GALEP auch nach Jahren noch auf dem neusten Stand.

GALEP's einzigartige Flexibilität wird durch die besondere Hardware ermöglicht. Alle Pins des 40 poligen Programmiersockels sind gleichwertig und bieten drei getrennt einstellbare Spannungen bis zu 25 Volt, Ground, Takt, einstellbare Pull-Up / Pull-Down Widerstände sowie Logik Ein-Ausgänge mit bidirektionalen Level-Shiftern für Low-Voltage Bauteile. Die interne Logik des GALEP ist über ein hochintegriertes FPGA frei programmierbar und wird optimal auf das jeweils ausgewählte Bauteil angepasst. Damit können alle Bauteile mit der bestmöglichen Geschwindigkeit programmiert oder ausgelesen werden.

Systemvoraussetzungen: Windows 95, 98, ME, NT oder Windows 2000

Lieferumfang: GALEP-4-Programmer, PC-Verbindungskabel, Steckernetzteil, Software mit Anleitung in Deutsch und Englisch auf CD. 6 x NiMH AAA Akkus (600mA) sind nicht im Lieferumfang enthalten und nur notwendig, wenn Akku-Betrieb gewünscht wird.



Abb. 10.3

10.4 Das Programmieren des GAL's 22V10

Nachdem man das GAL-Programm mit Hilfe eines Editors geschrieben und danach mit dem Programm FGAL in das JEDEC-Format assembliert hat, kann man nun mit dem Programmiergerät GALEP-4 diese JEDEC-Datei auf den GAL 22V10 "brennen".

Vorbereitung:

Vor dem "Brennen" muss die Software GALEP32 auf den PC installiert werden. Zudem muss der PC eine parallele Schnittstelle besitzen, an der der GALEP-4 angeschlossen wird.

Ausführung:

- Starte die Software GALEP32
- Öffne die *.jed-Datei
- Bauteil auswählen (hier Lattice GAL22V10D)
- Bauform auswählen (hier DIL-24)
- Das Bauteil in den GALEP-4 einstecken
- Mit der Taste F7 den GAL löschen <mit Enter bestätigen>
- Mit der Taste F3 den GAL programmieren <mit Enter bestätigen>
- Bauteil aus dem GALEP-4 entfernen

Fertig

10.5 Das Galprogramm

Title FGAL GAL file for GAL22V10
Pattern Umschaltung von 8-Bit-, auf 16-Bitbetrieb
Revision 1.0

Author M.Hayen und M.Titze
Date Apr 17, 2007

Pinbelegung

Pin1 D0 = Hauptrichtung unten GN
Pin2 D1 = Hauptrichtung unten GE
Pin3 D2 = Hauptrichtung unten RT
Pin4 D3 = Port A Bit 4
Pin5 D4 = Port A Bit 3
Pin6 D5 = Port A Bit 4
Pin7 D6 = Port A Bit 3
Pin8 D7 = Fahrbahnkontakt
Pin9 D8 = Ampeltaster Hauptrichtung
Pin10 D9 = 0=1Portbetrieb 1=2Portbetrieb
Pin11 D10 = Port B Bit 2
Pin13 D11 = Port B Bit 3

Pin14 Q0 = Fussgaengerampel Nebenrichtung RT
Pin15 Q1 = Fussgaengerampel Nebenrichtung GN
Pin16 Q2 = Fussgaengerampel Hauptrichtung RT
Pin17 Q3 = Fussgaengerampel Hauptrichtung GN
Pin18 Q4 = Hauptrichtung oben GN
Pin19 Q5 = Hauptrichtung oben GE
Pin20 Q6 = Hauptrichtung oben RT
Pin21 Q7 = Port A Bit 3
Pin22 Q8 = Port A Bit 4

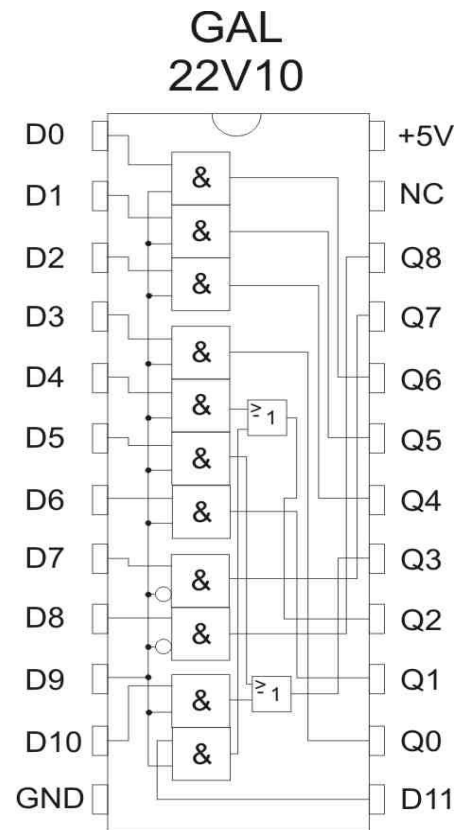


Abb. 10.4

CHIP COUNTER GAL22V10

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 GND

[13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24]
D11 Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 - VCC

EQUATIONS

Q0.TRST = /D9*/D3
Q1.TRST = /D9*/D6
Q2 = /D9*/D4+D9*/D11
Q3 = /D9*/D5+D9*/D10
Q4.TRST = /D9*/D2
Q5.TRST = /D9*/D1
Q6.TRST = /D9*/D0
Q7.TRST = D9*/D7
Q8.TRST = D9*/D8

11. Aufgabenblätter

11.1 Allgemeines

Eine weitere Forderung waren druckfertige Aufgabenblätter zu jeder Betriebsart im Dateiformat .pdf. Diese folgen auf den nächsten Seiten und sind auf der CD als Einzelblattversion im .odt-Format (OpenOffice-Document-Text) sowie im .pdf-Format im Ordner „/Projekt Ampel/Aufgabenblätter“ zu finden.

Aufgabenblatt 1 enthält die Aufgabe „Zeitgeführte Steuerung“
Aufgabenblatt 2 enthält die Aufgabe „Zeit & Prozessgeführte Steuerung“

Die Besonderheiten einer jeden Assemblersprache sowie das Kompilieren, Linken und die Übertragung zu den Mikrocontrollerboards werden in den einzelnen Unterrichten angesprochen.

Die Funktionen der Programme sind der Aufgabenstellungen zu entnehmen.

Des weiteren wurden Programme zur Vorführung jeder Betriebsart (im EPROM, für den EMES/EMUF) gefordert. Diese sind auf der CD im Ordner „/Projekt Ampel/Software“ enthalten und repräsentieren die **Musterlösung** der Ampelsteuerung und werden im folgenden als 8-Bit-Programm (Ampel1.asm) und als 16-Bit-Programm (Ampel2.asm) dargestellt.

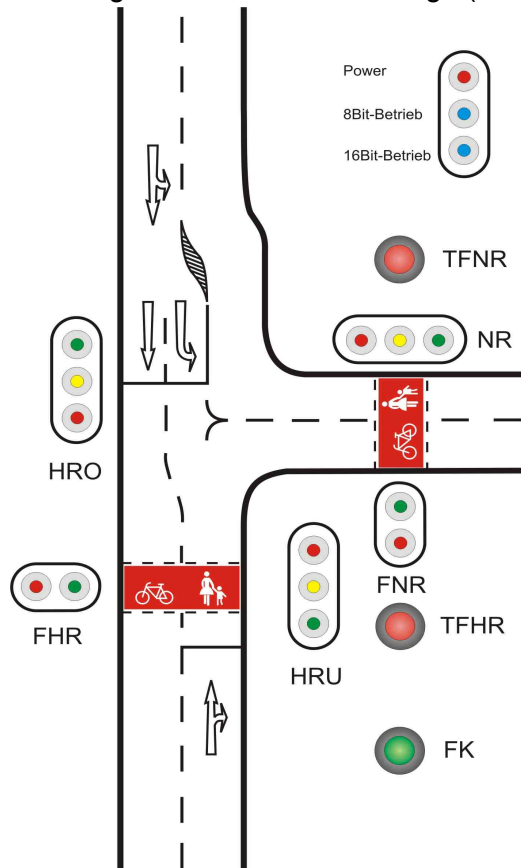
Die Programme wurden in der Z80-Assemblersprache geschrieben. Für andere Mikrocontrollerboards müssen die Programme entsprechend der vorgegebenen Assemblersprache angepasst werden. Auf der mitgelieferten CD findet man zudem fertig kompilierte Programme die mit Hilfe eines Sendeprogramms (TRANSMIT) auf den EMES/EMUF übertragen kann.

Das beigefügte DELAY-Programm dient als Verzögerung und lässt den Mikrocontroller für 100ms in einer Warteschleife, mit Hilfe einer Variable lassen sich die 100ms um das n- fache verlängern. Dies ist wichtig, denn dadurch lassen sich die einzelnen Ampelphasen einfach programmieren.

11.2 Aufgabenblatt 1

Zeitgeführte Steuerungen

Steuerung einer Lichtzeichenanlage (Tabellenverarbeitung)



- Erstellen Sie das Programm 'AMPEL1.asm' unter Verwendung des bereits vorhandenen Programms 'DELAY'!
- Übertragen Sie das Programm zum EMES/EMUF.
- Überprüfen Sie die Funktion am Modell.

Abb. 11.1

Schaltfolge und Portbelegung

Port / Bit	Signal	Anlauf	Phasen							
			1	2	3	4	5	6	7	8
A7	HR grün		■							
A6	gelb	<i>blinkt</i>		■						■
A5	rot				■	■	■	■	■	■
A4	F rot	<i>leuchtet</i>	■	■	■	■	■	■	■	■
A3	grün							■		
A2	NR grün				■					
A1	gelb	<i>blinkt</i>			■		■			
A0	rot		■	■	■			■	■	■
Steuerwort Port A										
Zeit in Sekunden			15	2	1	8	2	6	2	1
Zeitwort für DELAY										

11.3 Aufgabenblatt 2

Zeit- und Prozessgeführte Steuerungen

Steuerung einer Lichtzeichenanlage (Interruptverarbeitung)

Erstellen Sie das Programm 'AMPEL2.asm' ablauffähig für den 'EMES/EMUF'.

Die normale, zeitgeführte Steuerung (Normalablauf) soll durch den Prozess beeinflusst werden (Interruptverarbeitung).

Folgende Abläufe sind zu realisieren:

1. Normalablauf..... Phasen: 1,2,3,12.....>1
2. Fahrbahnkontakt löst aus (FK)..... Phasen: 1,2,3,4,5,6,7,11,12..>1
3. Fußgängertaster löst aus (TFHR; TFNR)..... Phasen: 1,2,3,4,8,9,10,11,12>1
4. Fußgängertaster und Fahrbahnkontakt lösen aus..... Phasen: 1 bis 12.....>1

Schaltfolge und Portbelegung														
Port / Bit	Signal	Anlauf	Phasen											
			1	2	3	4	5	6	7	8	9	10	11	12
B7	HRO rot													
B6	gelb	<i>blinkt</i>												
B5	grün													
B4	< TFHR >		<i>don't care</i>											
B3	FNR grün													
B2	rot													
B1	FHR grün													
B0	rot													
A7	HRU grün													
A6	gelb	<i>blinkt</i>												
A5	rot													
A4	< TFNR >		<i>don't care</i>											
A3	< FK >		<i>don't care</i>											
A2	NR grün													
A1	gelb	<i>blinkt</i>												
A0	rot													
Steuerwort Port B														
Steuerwort Port A														
Zeit in Sekunden		10 x 1Hz	15	2	5	2	1	10	2	1	5	1	1	1
Zeitwort für DELAY														

12. EMES Software

12.1 Das 8-Bit Programm

AMPEL1.ASM

;Ampel rein zeitgeführte Steuerung

external delay,hireg

acon equ 12h

adat equ 10h

```

ampel:      cseg                ;Codesegment
pioinit:   ld a,0cfh           ;Mode Controlword PIO
              out(acon),a        ;zur PIO Port A
              ld a,00000000b      ;alles auf Ausgabe
              out(acon),a        ;
              ld ix,hireg         ;Zeiger auf Hilfsregister

start:      call anlauf
              ld hl,stwtab        ;Zeiger auf Steuer+Zeitworte
              ld b,8              ;Zähler auf 8

phasen:    ld a,(hl)           ;Steuerwort laden
              out(adat),a        ;Ausgabe an PIO
              inc hl              ;Zeiger auf Zeitwort
              ld a,(hl)           ;Zeitwort laden
              ld (ix+3),a         ;Zeitwort ins Zeitregister
              call delay          ;...warten...
              inc hl              ;Zeiger auf nächstes Steuerwort
              dec b               ;Zähler -1
              jp nz,phasen        ;weiter wenn Zähler nicht Null
              jp start            ;Zurück zu Start

anlauf:     ld b,10
schleife:  ld hl,stwtab2
              ld a,(hl)
              out(adat),a
              ld (ix+3),5
              call delay
              inc hl
              ld a,(hl)
              out(adat),a
              ld (ix+3),5
              call delay
              dec b
              jp nz,schleife
              jp start

```

stwtab: **db 91h,150** *;Phase 1 Steuerwort und Zeitwort*
 db 51h,20 *;Phase 2*
 db 33h,10 *;Phase 3*
 db 34h,80 *;Phase 4*
 db 32h,20 *;Phase 5*
 db 29h,60 *;Phase 6*
 db 31h,20 *;Phase 7*
 db 71h,10 *;Phase 8*

stwtab2: **db 10h**
 db 52h

 end

12.2 Das 16-Bit Programm

AMPEL2.ASM

;Ampel interruptgeführte Steuerung

external delay,hireg

acon equ 12h
adat equ 10h
bcon equ 13h
bdat equ 11h

ampel1: cseg

pioinit: ld a,0cfh ;*Mode Controlword PIO*
 out (acon),a ;*zur PIO Port A*
 ld a,00011000b ;*alles auf Ausgabe*
 out (acon),a ;*zur PIO Port A*
 ld a,0cfh ;*siehe A nur für B*
 out (bcon),a
 ld a,00010000b
 out (bcon),a
 ld a,00010111b ;*Interruptsteuerwort disable,or,high,msc follows*
 out (acon),a ;*zum Port A Befehlsregister*
 ld a,11100111b ;*Monitore Bit 3 und 4*
 out (acon),a ;*Maske für Interrupt*
 ld a,low(vektab) ;*Low -Byte der Interruptadrestabelle zur*
 out (acon),a ;*PIO*
 ld a,00010111b ;*Interruptsteuerwort disable,or,high,msc follows*
 out (bcon),a ;*zum Port A Befehlsregister*
 ld a,11101111b ;*Monitore Bit 4*
 out (bcon),a ;*Maske für Interrupt*

 ld a,low(vektab+2) ;*Low -Byte der Interruptadrestabelle 2 zur*
 out (bcon),a ;*PIO*

 im 2 ;*Interrupt Betriebsart Nr.2*
 ld a,high(vektab) ;*High Byte in*
 ld i,a ;*CPU*

 ei ;*enable Interrupt*
 ld a,83h ;*Freigabe Steuerwort PIO*
 out (acon),a ;*für Interrupt an Port A*
 out (bcon),a ;*und an Port B*

 ld ix,hireg ;*Zeiger auf Hilfsregister*
 ld (ix+0),00000000b
 call anlauf

anlauf:	ld b,10	<i>;Zähler auf 10</i>
blink:	ld a,42h	<i>;Steuerwort Fußgänger ro Auto ge laden</i>
	out (adat),a	<i>;an PIO Port A senden</i>
	ld a,45h	<i>;für B</i>
	out (bdat),a	
	ld (ix+3),5	<i>;lade Zeitkonstante 0,5 s</i>
	call delay	<i>;warte</i>
	ld a,00h	<i>;Steuerwort Fußgänger ro Auto aus laden</i>
	out (adat),a	<i>;an PIO Port A senden</i>
	ld a,05h	<i>;für B</i>
	out (bdat),a	
	ld (ix+3),5	<i>;lade Zeitkonstante 0,5 s</i>
	call delay	<i>;warte</i>
	dec b	<i>;Zähler -1</i>
	jp nz,blink	<i>;weiter, wenn Zähler nicht Null</i>
	ret	<i>;sonst Start</i>
 start:	 ld hl,stwtab	 <i>;Zeiger auf Steuer- und Zeitworte</i>
	ld b,12	<i>;Zähler auf 12</i>
 phasen:	 call phasentest	
	ld a,(hl)	<i>;Steuerwort laden</i>
	out (adat),a	<i>;und zur PIO</i>
	inc hl	<i>;Zeiger auf Zeitwort</i>
	ld a,(hl)	<i>;wie zuvor für B</i>
	out (bdat),a	
	inc hl	
	ld a,(hl)	<i>;Zeitwort laden</i>
	ld (ix+3),a	<i>;Zeitwort ins Zeitregister</i>
	call delay	<i>;rufe Delay</i>
	inc hl	<i>;Zeiger auf nächstes Steuerwort</i>
	dec b	<i>;Zähler -1</i>
	jp nz,phasen	<i>;weiter, wenn Zähler nicht Null</i>
	jp start	<i>;sonst von vorne</i>
 phasentest:	 ld a,09h	 <i>;lade Akku mit 9 für Zähler Phase 3</i>
	cp b	<i>;prüfe Zähler</i>
	jp z,phase3	<i>;wenn gleich dann nach3</i>
	ld a,08h	<i>;lade Akku mit 8 für Zähler Phase 4</i>
	cp b	<i>;prüfe Zähler</i>
	jp z,phase4	<i>;wenn gleich dann nach4</i>
	ld a,05h	<i>;lade Akku mit 5 für Zähler Phase 7</i>
	cp b	<i>;prüfe Zähler</i>
	jp z,phase7	<i>;wenn gleich dann nach7</i>
	ret	<i>;sonst mache weiter</i>
 phase3:	 ld a,00h	 <i>;setze Akku mit Null für kein Merker</i>
	cp (ix+0)	<i>;prüfe Merker</i>
	ret nz	<i>;wenn Merker dann weiter mit Phase 4</i>
	ld bc,18h	<i>;lade BC mit Anzahl Sprünge bis Phase 12</i>
	add hl,bc	<i>;verschiebe Zeiger Hilfsregister um Anzahl Sprünge</i>
	ld b,01h	<i>;Setze Zähler auf 1</i>
	ret	<i>;mache weiter</i>

```

phase4:      bit 1,(ix+0)      ;prüfe ob Fahrbahnmerker gesetzt
               ret nz           ;wenn ja dann weiter mit Phase 5
               ld bc,0ch        ;lade BC mit Anzahl Sprünge bis Phase 8
               add hl,bc        ;verschiebe Zeiger Hilfsregister um Anzahl Sprünge
               ld b,05h         ;setze Zähler auf 5
               res 1,(ix+0)      ;Fahrbahnmerker zurücksetzen
               ret              ;mache weiter
phase7:      bit 0,(ix+0)      ;prüfe ob Fußgängermerker gesetzt
               ret nz           ;wenn ja dann weiter mit Phase 8
               ld bc,09h        ;lade BC mit Anzahl Sprünge bis Phase 11
               add hl,bc        ;verschiebe Zeiger Hilfsregister um Anzahl Sprünge
               ld b,02h         ;Setze Zähler auf 2
               res 0,(ix+0)      ;Fußgängermerker zurücksetzen
               ret              ;mache weiter

irout_a:      push af          ;Akku und Flagregister auf Stack / sichern der Register
               in a,(adat)      ;lese Port A aus
               bit 4,a          ;prüfe Bit 4 für Fußgänger
               jp nz,fussmerker  ;wenn nicht Null dann zuFußgängermerker

fahrmerker:   set 1,(ix+0)      ;ansonsten Fahrbahnmerker setzen
               pop af          ;Akku und Flags restaurieren
               ei              ;Freigabe CPU
               reti            ;Freigabe PIO

irout_b:      push af          ;Akku und Flagregister auf Stack / sichern der Register

fussmerker:   set 0,(ix+0)      ;Fußgängermerker setzen
               pop af          ;Akku und Flags restaurieren
               ei              ;Freigabe CPU
               reti            ;Freigabe PIO

stwtab:      db 81h,25h,150
               db 41h,25h,20
               db 21h,25h,50
               db 21h,45h,20
               db 23h,85h,10
               db 24h,85h,100
               db 22h,85h,20
               db 21h,85h,10
               db 21h,8Ah,50
               db 21h,85h,10
               db 61h,0C5h,10
               db 61h,25h,10

hugo segment code privat para

vektab:      dw irout_a
               dw irout_b

               end

```

12.3 Das Delay-Programm

```
;Warteschleife
;das Register IX+3 muss vorher geladen werden
;Delay = Verzögerungskonstante in ix+3 x 100ms
;feste Verzögerung ms10 von 100ms
;feste Verzögerung ms1 von 1ms
;variable Verzögerung ms1var ix+2 x 1ms
;konstante 256 bedeutet eine 0 in Zeitregister laden
```

```
delay:      cseg                ;Codesegment
             call ms100         ;warte 100ms
             dec (ix+3)         ;K3 vermindern
             jp nz, delay       ;solange K3 nicht Null
             ret                ;weiter warten

ms100:      ld (ix+2), 98       ;Konstante für 100ms laden
ms1var:     call ms1           ;1ms warten
             dec (ix+2)         ;K2 vermindern
             jp nz, ms1var      ;solange K2 nicht Null
             ret                ;weiter warten

ms1:        ld (ix+1), 74       ;Konstante für 1ms laden
loop:      dec (ix+1)         ;K1 vermindern
             jp nz, loop        ;solange K1 nicht Null
             ret                ;weiter warten

hireg:      dseg                ;Datensegment
             ds 4               ;4 Speicherplätze reservieren
                               ;für Zeitkonstanten

             global delay, ms100, ms1var, ms1, hireg

             end
```

13.1 Stückliste Ampel (Fa. Reichelt Elektronik)

Nr.	Bezeichnung	Bestellnr.:	S.	Anzahl	á	Gesamt
1	Distanzrollen 5mm	DK 5mm	296	5	0,05 €	0,25 €
2	Folie, Selbstklebend	CANSON 987-241	652	1	11,55 €	11,55 €
3	GAL 2210 Dil-24	GAL 22V10-15LP	342	1	2,05 €	2,05 €
4	Gehäuse	Teko 104	282	1	10,40 €	10,40 €
5	IC 74LS	LS05	323	3	0,17 €	0,51 €
6	IC-Fassung	GS-KO 24P-S	447	1	0,71 €	0,71 €
7	IC-Fassung	GS-KO 14P	447	3	0,54 €	1,62 €
8	Kabel, Daten vom EMES	AK 4010	626	1	1,40 €	1,40 €
9	Kaltgeräteanschlusskabel	NKSK 200 SW	93	1	1,20 €	1,20 €
10	Kaltgerätestecker	KES 1	90	1	0,51 €	0,51 €
11	LED 3mm, hell	SLK 3mm bl	364	2	0,50 €	1,00 €
12	LED 3mm, hell	SLK 3mm rt	364	1	0,07 €	0,07 €
13	LED 3mm, superhell	SLH 36 gn	364	5	0,09 €	0,45 €
14	LED 3mm, superhell	SLH 36 rt	364	5	0,09 €	0,45 €
15	LED 3mm, superhell	SLH 36 ge	364	3	0,09 €	0,27 €
16	LED Einbaufassung	EBF I-3	369	16	0,39 €	6,24 €
17	Metall Schichtwiderstand	Metall 56,0	400	2	0,08 €	0,16 €
18	Metall Schichtwiderstand	Metall 2,20k	400	2	0,08 €	0,16 €
19	Metall Schichtwiderstand	Metall 10,0k	400	2	0,08 €	0,16 €
20	Muttern M2,5	SK M2,5-100	921	1	1,45 €	1,45 €
21	Muttern M3	SK M3-100	921	1	0,90 €	0,90 €
22	Netzschalter Baureihe 1855	Wippe 1855.1102	496	1	2,85 €	2,85 €
23	Netzteil 5V	SNT MW25-05M	73	1	17,05 €	17,05 €
24	Platine, Beidseitig	BEL 160x100-2	273	1	2,15 €	2,15 €
25	Platine, Einseitig	BEL 75x100-1	273	1	0,94 €	0,94 €
26	Platinen Steckverbinder	PS 25/5G ws	450	4	0,63 €	2,52 €
27	Platinen Steckverbinder	PS 25/2G ws	450	3	0,25 €	0,75 €
28	Platinen Steckverbinder	PS 25/8G ws	450	2	0,92 €	1,84 €
29	Platinen Steckverbinder	PS 25/3G ws	450	4	0,43 €	1,72 €
30	Schrauben M2,5 8mm	SZK M2,5x8	918	1	1,60 €	1,60 €
31	Schrauben M3 12mm	SZK M3x12-200	918	1	1,65 €	1,65 €
32	Schrumpfschlauch 4:1	SDH 25-4	264	1	1,85 €	1,85 €
33	Schrumpfschlauch 4:1	SDK 4,0 sw	264	1	1,40 €	1,40 €
34	Sicherungen	Träge 1,8A	267	10	0,07 €	0,72 €
35	Sicherungshalter	PL 125000	270	1	0,34 €	0,34 €
36	Stiftleiste 1x36pol	SL 1X36G 2,54	448	1	0,17 €	0,17 €
37	Sub-D Buchse, gewinkelt	D-SUB BU 25EU	442	1	0,43 €	0,43 €
38	Sub-D Buchse, Lötkelch	D-SUB BU 25	442	1	0,10 €	0,10 €
39	Taster, Fahrbahn	Taster 9146 gn	491	1	2,00 €	2,00 €
40	Taster, Fußgänger	Taster 9146 rt	491	2	2,00 €	4,00 €
41	Umschalter Betrieb	T217	494	1	0,12 €	0,12 €
42	Wannenstecker	WSL 20G	453	1	0,07 €	0,07 €
42	Widerstandnetzwerk	SIL 10-5 150	401	3	0,18 €	0,54 €
						<u>86,32 €</u>

13.2 Weitere Bauteile

- „Bastelglas“ oder Plexiglas, 1,5mm dick, Abmessungen wie Frontplatte
- 0,75mm² Litze, schwarz, blau, grün/gelb; zur Verdrahtung von Schalter, Sicherungshalter und Netzanschluss
- Sekundenkleber oder Heißklebepistole zum fixieren der Distanzhülsen für Platinenmontage auf Gehäuseboden

14. Zusammenbau

14.1 Verwendete Werkzeuge, Materialien und Software

Zusammenbau der Ampel-Hardware

- Abisolierzange
- Aceton
- Anreißnadel
- Anschlagwinkel
- Biegelehre
- Elektronikseitenschneider
- Entlötpumpe
- Flachrundzange-gebogen
- Geodreieck
- Hammer, 300g
- Hand(kegel)senker
- Heissluftfön
- Körner
- Lineal
- Lötlack, KontaktChemie SK10
- Lötstation, temperaturgeregelt, 50W, Spitze 0,8mm rund
- Lötzinn Sn60 Pb38 Cu2; F-SW 26
- Meßschieber
- Präzisionshandbohrmaschine (Proxxon) & Bohrer: 3,2mm; 2,5mm; 1,2mm; 1mm; 0,8mm; 0,5mm
- Schlitzschraubendreher
- Schlüsselfeile, vierkant und flachstumpf, Hieb2
- Sekundenkleber
- Skalpell/Messerset
- Schraubzwingen
- Standbohrmaschine & Bohrer: 13mm; 8mm; 6mm,

Programmierung

- Handelsüblicher PC mit Parallelport zum Anschluss des EMES
- Programmiergerät GALEP4 mit Software GALEP32
- Software FGAL, CZ80 und TRANSMIT
- einfacher Texteditor

Verwendete Software zur Planung und Dokumentation

- Corel Draw Version 12 zur Erstellung der Zeichnungen, exportiert ins JPEG-Format
- EAGLE Version 4.16 zur Erstellung des Platinenlayouts
- OpenOffice 2.1 zur Erstellung der Dokumentation

14.2 Zusammenbau der Platinen

- Belichten, Entwickeln und Ätzen der Platinen, anschl. Bohren der Bauteillöcher,
 - 0,5mm Durchkontaktierungen
 - 0,8mm Widerstände
 - 1,0mm Widerstands-Netzwerke und IC-Fassungen
 - 1,2mm Platinensteckverbinder, Sub-D Winkelstecker
 - 3,2mm Schraubenlöcher, Sub-D Winkelstecker
- Entfernen der Fotobeschichtung mit Aceton, anschl. Einsprühen mit Lötlack
- Bestücken der Platine, beginnend mit kleinstem Bauteil.
 - Durchkontaktierungen -> Widerstände -> IC-Fassungen -> R-Netzwerke...
- Bei einigen Platinensteckverbindern wurde ein Anschlußpin herausgezogen!
Grund: 4-polige Platinensteckverbinder nicht verfügbar, 5-pol Variante verwendet.
- Verschrauben von Netzteil und Platine, Verbinden der Taster & LED's mit offenen Enden der Steckverbinder (siehe Zusammenbau Frontplatte)
- Einbau in Gehäuse
- Einstecken der Steckverbinder und einsetzen der IC's in Fassungen

14.3 Zusammenbau der Frontplatte

- Bohrschablone ausdrucken, auf Aluminiumfrontplatte befestigen, Bohrungen vorsichtig körnen, vorbereitete Alu-Frontplatte auf anschl. gemäß Bohrplan bohren, ggf. Bohrlöcher für Taster mit Schlüsselfeilen nachbearbeiten.
- Klebefolie bedrucken und ausschneiden, anschließend aufkleben oder wahlweise normale Transparentfolie verwenden, Öffnungen der LED's und Taster mit Skalpell herausschneiden
- Einsetzen der LED's in Fassungen, anschl. mit Frontplatte verschrauben,
- Einsetzen der Taster, die im Prototyp verwendeten Taster sind mit einer Leuchte ausgestattet, diese wird jedoch nicht angeschlossen. Taster ohne Leuchtfunktion war z.Zt. der Prototypherstellung nicht lieferbar.
- Verdrahten gemäß Verdrahtungsplan, die Anoden der „Ampeln“ wurden etwas gekürzt und mit einem Stück Silberdraht „zusammengeschaltet“.

14.4 Zusammenbau des Gehäuses

- Positionen der Bauteile der Gehäuserückwand mit Hilfe von Geodreieck, Lineal und Anreißnadel übertragen.
- Bohren der Löcher gemäß Bohrschablone, Rückwand & Gehäuseboden
 - Befestigungslöcher der Platine mit Hilfe der Anreißnadel festlegen
- ggf. Öffnungen für Schalter, Netzanschluss mit Schlüsselfeilen nacharbeiten
- Befestigen der Gehäusefüße am Gehäuseboden, Position nach eigenem Ermessen (Empfehlung: so weit außen wie möglich)
- Anlöten der Litze an Netzanschlusstecker, Schalter und Sicherungshalter, anschl. Einbau in Gehäuserückwand
 - **Schrumpfschlauch ist über alle Netzspannung führenden Teile zu ziehen. Reihenfolge beim Anlöten beachten !!!**
- Anlöten der Platinensteckverbinder an 24pol. Sub-D Buchse, gemäß Belegungsplan, anschl. Einbau in Gehäuserückwand
- Anlöten der Platinensteckverbinder an Betriebswahlschalter, gemäß Belegungsplan, anschl. Einbau in Gehäuserückwand
- Distanzhülsen mit einem Tropfen Sekundenkleber auf Gehäuseboden fixieren, ca. 10min antrocknen lassen.
- Platine einsetzen und verschrauben, Schraubenköpfe auf Platine, Muttern am Gehäuseboden
- Verbinden der Platinensteckverbinder mit Platine.
- Einsetzen der Schmelzsicherung in Sicherungshalter
- Anschluss von Netz- und Signalleitung
- Übertragen eines Ampelprogramms von PC auf EMES zu abschließenden Testzwecken
- Funktionstest

15. Entscheidungen und ihre Gründe

- Lieferanten und Hersteller
 - Hier gab es keine Vorgaben seitens der Schule. Die Wahl fiel schnell auf die Fa. Reichelt Elektronik aus Sande, da dort fast alle benötigten Teile sofort verfügbar waren und „im Falle eines Falles“ schnell ein Ersatzteil beschafft werden konnte.
- Gehäuse und Netzteil
 - gefordert wurde „Gehäuseeinbau“ und „internes Netzteil, Anschluss an 230V AC“
Das Gehäuse der Fa. Teko, Typ „Teko104“ war groß genug die Schaltung aufzunehmen, optisch ansprechend (schwarz, Frontplatte Aluminium) , kostengünstig und am Markt verfügbar.
 - Schaltnetzteil der Fa. MeanWell, Typ „RS-25-5“ entsprach den Anforderungen von Bauform und Abmessungen, Leistungsmäßig für Ampelschaltung allerdings immer noch überdimensioniert (5V/5A Dauerleistung; benötigt werden 170mA).
- Platine und doppelseitiges Layout
 - gefordert: „Fertigung eines Prototypen“
Belichtungsgerät, Entwickler und Ätzgerät waren verfügbar, d.h. einen Prototyp auf Lochrasterplatine mit Wire-Technik zu verdrahten viel aus Zeitgründen und fehlendem Verdrahtungswerkzeug aus.
 - Ursprünglich geplantes, einseitiges, geätztes Platinenlayout aus Platzgründen nicht realisierbar, Entscheidung fiel auf Platine mit beidseitiger Leiterbahnführung, da das Schaltnetzteil mit auf der Platine befestigt werden sollte.
- Netzanschluss mit Kaltgerätestecker, separatem Sicherungshalter und Netzschalter
 - Vorgabe war ein internes Netzteil, d.h. Netzspannung im Gehäuse, durch die Aluminium-Frontplatte ist das ausgewählte Gehäuse jedoch nicht mehr schutzisoliert, d.h. ein Anschluss mit Eurostecker ist wegen des fehlenden Schutzleiters nicht zulässig, des weiteren musste auch die Frontplatte mit dem Schutzleiter verbunden werden.
 - Netzseitig ist die Phase bzw. nachfolgende Schaltung durch eine träge 1,8A Sicherung geschützt.
 - Netzschalter schaltet Phase und Null und leuchtet im Betrieb.
- Sub-D Druckerkabelverlängerung als Datenkabel
 - gefordert war eine „universelle handelsübliche Verbindungsleitung“. Die gewählte Sub-D-Verlängerung ist kostengünstig, vielseitig, robust und dank 24poliger Belegung zukunftssicher für evtl. Veränderungen am EMES oder Austausch gegen neueres Mikroprozessorsystem.

15. Entscheidungen und ihre Gründe -Fortsetzung-

- Reparaturfreundlicher Aufbau
 - Ausgewählt wurden 2, 3 und 5 polige Molex-Stift und Buchsenleisten mit ca. 25cm langen, fest verdrahteten, Anschlussleitungen um eine Fehlersuche auch bei geöffnetem Gehäuse zu ermöglichen.
 - Alle IC's wurden in Sockel mit Präzisionskontakten und Abblockkondensator eingesetzt.
 - Allgemein einfacher Aufbau durch digitale Grundsaltung (nur H und L Pegel).
- Steuerung durch GAL
 - gefordert wurde ein Betriebswahlschalter zur Umschaltung von 8Bit und 16Bit Betrieb, umgesetzt durch einen Miniaturschiebeschalter in der Gehäuserückwand.
 - Umsetzung durch einen GAL vom Typ 22v10, dieser übernimmt die Steuerung der 8Bit Betriebsart indem er einige Ports zusammenschaltet (z.B. Steuerung beider Fußgängerampeln nur noch durch zwei Bits; rot/grün; anstatt 4Bits im 16Bit Betrieb)

16. Meilensteine

(oder wie man ein eckiges Schwein durch ein rundes Loch schiebt)

07.02.2007	Schaltungsanalyse der alten Ampelschaltung, Auslesen des Layouts und übertragen in einen vorläufigen Schaltplan
28.02.2007	Bestückung eines Testboards mit Kapazitätssensor zu Testzwecken -> Erfolglos -> Neuverdrahtung des Testboards mit Kapazitätssensor -> Keine Funktion -> Austausch aller Halbleiter -> ohne Erfolg, Kapazitätssensor letztendlich verworfen
07.03.2007	Entfernen von Fehlern im Schaltplan, Verfeinern des Layouts, Entscheidung für Doublelayer getroffen, da einseitiges Layout zu Zeit- und Brückenintensiv.
08.03.2007	Belichten, Entwickeln und Ätzen der ersten Prototyp Platine an der BBS Wittmund. Bohren der Bauteillöcher, Auftragen von Lötack. Bestücken des Prototyps in Heimarbeit, dann erkannt das Lötstellen der IC-Fassungen teils auf Ober- wie Unterseite vorhanden.
14.03.2007	Ändern aller Lötstellen auf Unterseite im Layout, Entwicklung des Adaptersteckers Emes<->Ampel. Einkauf neuer Bauteile, <u>noch 8 Wochen bis Projektabgabe</u>
15.03.2007	Belichten, Entwickeln und Ätzen eines 2. Prototypen an der BBS Wittmund. Bohren und bestücken der Platine, Setzen der Durchkontaktierungen, Widerstände, IC-Fassungen..., Entwurf und Design des Frontplattenlayouts und des Logos
22.03.2007	Beginn der Verdrahtung des Gehäuses, Netzanschluss, Sicherung und Schalter und verdrahten der Frontplatte
04.04.2007	Programmieren des GAL's, anschl. erster Funktionstest -> keine Funktion, Fehlersuche, 2 Lötstellen und eine Durchkontaktierung erneuert, Anschlußsteckerbelegung des Datenkabels im Ampelgehäuse korrigiert. Fehlerhafte Leiterbahn unterbrochen. <u>Noch 4 Wochen bis Projektabgabe</u>
17.04.2007	Neuprogrammierung des GAL's, kleine Veränderungen des Layouts. Funktionsfähig!!!

17.1 Vorgehen bei der Problemlösung

Vor der Fehlersuche muss sichergestellt sein, dass das Gerät auch einen Fehler hat.

Warum?

Bei diesem Gerät handelt es sich um ein Peripheriegerät, das heißt ein Fehler der sich in der Peripherie zeigt, kann bereits im Mikrocontrollerboard oder in der Software vorhanden sein. Man sollte also anhand eines Zweitgeräts überprüfen ob es sich nicht um einen der oben genannten Fehler handelt.

Ist das Mikrocontrollerboard und die Software fehlerfrei, kann mit der Fehlersuche in der Ampelschaltung begonnen werden.

Fehlerindikation	Mögliche Fehlerursache
Die Ampelschaltung zeigt gar nichts an.	1. Überprüfen Sie Netzanschluss und Kabel. 2. Sicherung an Gehäuserückwand prüfen. 3. Im Schaltnetzteil befindet sich eine zusätzliche Sicherung (eingebaute LED leuchtet bei Netzspannung)
Power-LED leuchtet nicht.	1. Leuchtdiode und/oder Vorwiderstand defekt 2. IC 4 defekt
Eine oder mehrere LED's der Nebenrichtung leuchten nicht.	1. Leuchtdiode und/oder Vorwiderstand defekt 2. IC 2 defekt
Eine oder mehrere LED's der Hauptrichtung unten leuchten nicht.	1. Leuchtdiode und/oder Vorwiderstand defekt 2. IC 2 defekt 3. Eingänge des GAL's defekt
Eine oder mehrere LED's der Hauptrichtung oben leuchten nicht.	1. Leuchtdiode und/oder Vorwiderstand defekt 2. IC 3 defekt 3. GAL defekt
Eine oder mehrere LED's der Fußgängerampel Hauptrichtung leuchten nicht.	1. Leuchtdiode und/oder Vorwiderstand defekt 2. IC 3 defekt 3. GAL defekt
Die rote LED der Fußgängerampel Nebenrichtung leuchtet nicht.	1. Leuchtdiode und/oder Vorwiderstand defekt 2. IC 3 defekt 3. GAL defekt
Die grüne LED der Fußgängerampel Nebenrichtung leuchtet nicht.	1. Leuchtdiode und/oder Vorwiderstand defekt 2. IC 4 defekt 3. GAL defekt
Der Fahrbahnkontakt löst nicht aus.	1. Taster defekt 2. IC 4 defekt 3. GAL defekt
Der Fußgängertaster Nebenrichtung löst nicht aus.	1. Taster Fußgänger Nebenrichtung defekt 2. Widerstände R5 und/oder R6 defekt 3. GAL defekt
Der Fußgängertaster Hauptrichtung löst nicht aus.	1. Taster Fußgänger Hauptrichtung defekt 2. Widerstände R3 und/oder R4 defekt 3. GAL defekt

17.2 Ein Fehler, der gar keiner ist

Ein Fehler der uns in der Testphase fasst zum Verzweifeln gebracht hat, ist gar keiner. In der Testphase haben wir zwischen den beiden Testprogrammen (8-Bit und 16-Bit Programm) hin und her gewechselt. Hierbei passierte es, dass die Ampel nicht mehr funktionierte.

Was war passiert?

Wir sendeten das 8-Bit-Programm an den EMES, die Ampel funktionierte wie gewünscht. Dann sendeten wir das 16-Bit-Programm an den EMES, auch jetzt funktionierte die Ampel korrekt. Daraufhin sendeten wir wieder das 8-Bit-Programm an den EMES und die Ampel funktionierte plötzlich nicht mehr.

Der Grund hierfür liegt nicht in der Ampelschaltung, sondern am EMES/EMUF. Dieser setzt seine PIO-Ausgänge bei einem Rücksetzen durch einen RESET **NICHT** zurück.

Daher:

Wenn auf ein 16-Bit-Programm ein 8-Bit-Programm folgen soll, so muss der EMES/EMUF komplett neu gestartet werden.

18. Technische Daten:

Elektrische Daten

Eingangsspannung:	88V ~ 264 V AC
Betriebsspannung:	5V
Max. Leistung des Netzteils:	5A (25W)
Sicherung:	1,8A träge
Wirkungsgrad:	77%
Stromaufnahme der Ampelschaltung (je nach Ampelphase):	130 – 160 mA
Betriebstemperatur:	-20 ~ +70°C

Mechanische Daten

Abmessungen Ampel (LxBxH) (mm)	225x130x85
Gewicht (ohne Netz/Datenkabel):	760g

19. Ansprechpartner

Berufsbildende Schulen Friedenstraße Wilhelmshaven

Herr Appenzeller
Herr Tröck

Berufsbildende Schule Wittmund

Herr Hayen

20. Abbildungsverzeichnis

Abb. 4.3	Das Hauptproblem	S. 6
Abb. 5.1	Blockschaltbild	S. 7
Abb. 6.1	Stromlaufplan	S. 8
Abb. 6.2	EAGLE-Schaltplan Ampelplatine	S. 9
Abb. 6.3	EAGLE-Schaltplan Adapterplatine	S.10
Abb. 7.1	Grundsätzlicher Aufbau	S. 11
Abb. 7.2	Ampel bei „high“-Pegel	S. 11
Abb. 7.3	Ampel bei „low“-Pegel	S. 11
Abb. 7.4	Funktionsweise der Taster	S. 12
Abb. 8.1	Frontplatte, schematisch	S. 13
Abb. 8.2	Bohrschablone, Frontplatte	S. 14
Abb. 8.3	Verdrahtung, Frontplatte	S. 15
Abb. 8.4	Innenaufbau, schematisch	S. 16
Abb. 8.5	Verdrahtung, Innenaufbau	S. 17
Abb. 8.6	Verdrahtung der 25poligen Sub-D Buchse im Gehäuse	S.18
Abb. 8.7	Seitenansicht, schematisch	S. 19
Abb. 8.8	Rückansicht, schematisch	S. 19
Abb. 8.9	Bohrschablone Gehäuserückwand	S. 20
Abb. 8.10	Ampelplatine, Layout, Oberseite	S. 21
Abb. 8.11	Ampelplatine, Layout, Unterseite	S. 21
Abb. 8.12	Bestückungsplan Ampelplatine	S. 22
Abb. 9.1	Stromlaufplan Adapterplatine	S. 23
Abb. 9.2	Adapterplatine, schematisch	S. 24
Abb. 9.3	Adapterplatine, Layout	S. 25
Abb. 10.1	Funktionsweise GAL, Ausgang gesperrt	S. 25
Abb. 10.2	Funktionsweise GAL, Ausgang geöffnet	S. 25
Abb. 10.3	Programmiergerät Galep4	S. 27
Abb. 10.4	GAL 22v10, schematisch	S. 29

21. Datenblattverzeichnis

Hersteller	Typ	Seitenzahl
Lattice Semiconductor	GAL 22v10	21
Teko	Gehäuse Teko104	1
Marquardt	Schalter 1855	6
Monacor	Betriebswahlschalter T-217	1
Kingbright	LED L-934 MBC Blau	3
Kingbright	LED L-934 IT Rot, Gelb, Grün	7
VS Optoelectronic	LED Innenreflektor WU-I-3-x	2
MeanWell	Netzteil G3 RS-25	2
EFB Elektronik	D-Sub Buchse, Lötkelch	1
Assmann	D-Sub Snap-In Buchse, 90°	1
Schurter	Taster LDT 9146 rot/grün	5
diverse	Molex Platinensteckverbinder 5 polig	1
diverse	Kaltgeräteeinbaustecker	1