

HTBLuVA Graz – Gösting
Ibererstraße 15-21
8051 Graz



Abteilung für Elektronik

MIDI - Stepsequencer mit Arpeggiator

**Diplomarbeit im Fach Technische Informatik / Elektronik
und Digitaltechnik**

Eingereicht von:	Stefan Fötschl (TINF) Daniel Rutte (EDT)
im Schuljahr	2011 / 2012
der Klasse	5 BHELI
bei	Prof. DI. Helmut Benesch
am	11. Mai. 2012

Wir bedanken uns bei unserem Betreuer für die freundliche Unterstützung bei der Durchführung unserer Diplomarbeit.

Inhaltverzeichnis

<i>Inhaltverzeichnis</i>	3
1. Diplomandenvorstellung	5
1.1. Diplomand Stefan Fötschl.....	5
1.2. Diplomand Daniel Rutte	6
2. Einführung	7
2.1. Kurzbeschreibung	7
2.2. Abstract.....	8
2.3. Blockschaltbild zur Kurzbeschreibung	9
3. Hardware	10
3.1. Allgemeine Beschreibung	13
3.2. Arduino MEGA 2560.....	15
3.2.1. Allgemeines	15
3.2.2. Microcontroller AVR ATmega2560.....	17
3.3. Spannungsregler LM7805	20
3.3.1. Allgemeines	20
3.3.2. Beschaltung	21
3.4. 6N139 - Optokoppler.....	22
3.4.1. Allgemeines:	22
3.5. AS 1100	23
3.5.1. Allgemeines	23
3.6. LC- Display	26
3.6.1. Allgemein	26
3.6.2. Beschaltung	28
3.7. 74HCT14	29
3.7.1. Allgemeines	29
3.8. MIDI –In/ -Out/ -Thru.....	30
3.9. Platinen	32
3.9.1. Eagle- Library.....	32
3.9.2. Schaltplan.....	39
3.9.3. Layout.....	43
3.9.4. Aufbau	48
4. Software	49

4.1.	mySmartUSB Light	49
4.1.1.	Allgemein	49
4.1.2.	Technische Spezifikationen	50
4.1.3.	Programmierungsumgebung	50
5.	<i>Kostenaufstellung</i>.....	54
6.	<i>Probleme und Verworfenne Ideen</i>.....	55
6.1.	Design	55
6.2.	Schaltplan.....	55
7.	<i>Abbildungsverzeichnis</i>	56
8.	<i>Tabellenverzeichnis</i>.....	58
9.	<i>Quellenangabe</i>	59
10.	<i>Eidesstattliche Erklärung</i>	60

1. Diplomandenvorstellung

1.1. Diplomand Stefan Fötschl



Abb. 1-1: Stefan Fötschl

Diplomand: Stefan Fötschl

Geburtstag: 31.01.1993

Fachrichtung: Elektronik / Hardware Software Co-Design

Schwerpunktfach: Technische Informatik

Komplementärfach: Industrielle Elektronik

Klasse: 5 BHELI

Zuständig für: Schaltplan, Dokumentation, Programmierung

Betreuungslehrer: Prof. DI. Helmut Benesch

1.2. Diplomand Daniel Rutte



Abb. 1-2: Daniel Rutte

Diplomand: Daniel Rutte

Geburtstag: 08.02.1993

Fachrichtung: Elektronik / Telekommunikation und Nachrichtentechnik

Schwerpunktfach: Elektronik- und Digitaltechnik

Komplementärfach: Hochfrequenztechnik

Klasse: 5 BHELI

Zuständig für: Eagle- Library, Layout, Pinbelegung, Dokumentation, Aufbau der Hardware

Betreuungslehrer: Prof. DI. Helmut Benesch

2. Einführung

2.1. Kurzbeschreibung

Die Diplomarbeit behandelt einen MIDI (musical instrument digital interface) – Stepsequencer, ein Gerät, mit dem man 32 Töne aus einem Basis-Ton erschaffen kann, mit einem Arpeggiator, welcher dem Benutzer erlaubt, die Töne in Echtzeit via Keyboard und nicht nur am Stepsequencer selbst zu ändern.

Nachdem der Akkord auf einem Keyboard gespielt worden ist, wird er zum Stepsequencer gesendet, wo der Akkord in seine Einzelteile aufgeteilt werden kann. Um Töne zu konfigurieren, muss man die Drehregler am Gerät drehen und die gewünschten Noten einstellen (als Beispiel die Note C).

Mit einem solchen Stepsequencer müssen Musiker dieselbe Zerlegung nicht immer und immer wieder selbst spielen und können dem Arpeggiator die Aufteilung der Töne überlassen. Dieses Zusammenspiel ist in unserem Gerät weltweit das erste Mal im Einsatz.

Ein Stepsequencer ist ein Gerät, welches Musiker auf viele Arten nutzen können.

2.2. Abstract

The diploma thesis deals with a MIDI (musical instrument digital interface) – step sequencer which is all in all a device with which you can make up to 32 sounds from only one base sound, with an Arpeggiator which allows you to change your sounds in real time via your keyboard instead of only changing the sounds on your step sequencer.

After playing the chord on your keyboard it's sent to the step sequencer where it is separated into parts. To add sounds you have to turn dials on the device and choose the notes you want to hear (for example the note C).

With such a step sequencer musicians need not play the same partition over and over again and can let the Arpeggiator do the work at segmenting the sounds. This combination is used in our device worldwide for the first time.

A step sequencer and especially one with an Arpeggiator is a very useful device which you, if you are a musician, can use in a lot of ways.

2.3. Blockschaltbild zur Kurzbeschreibung

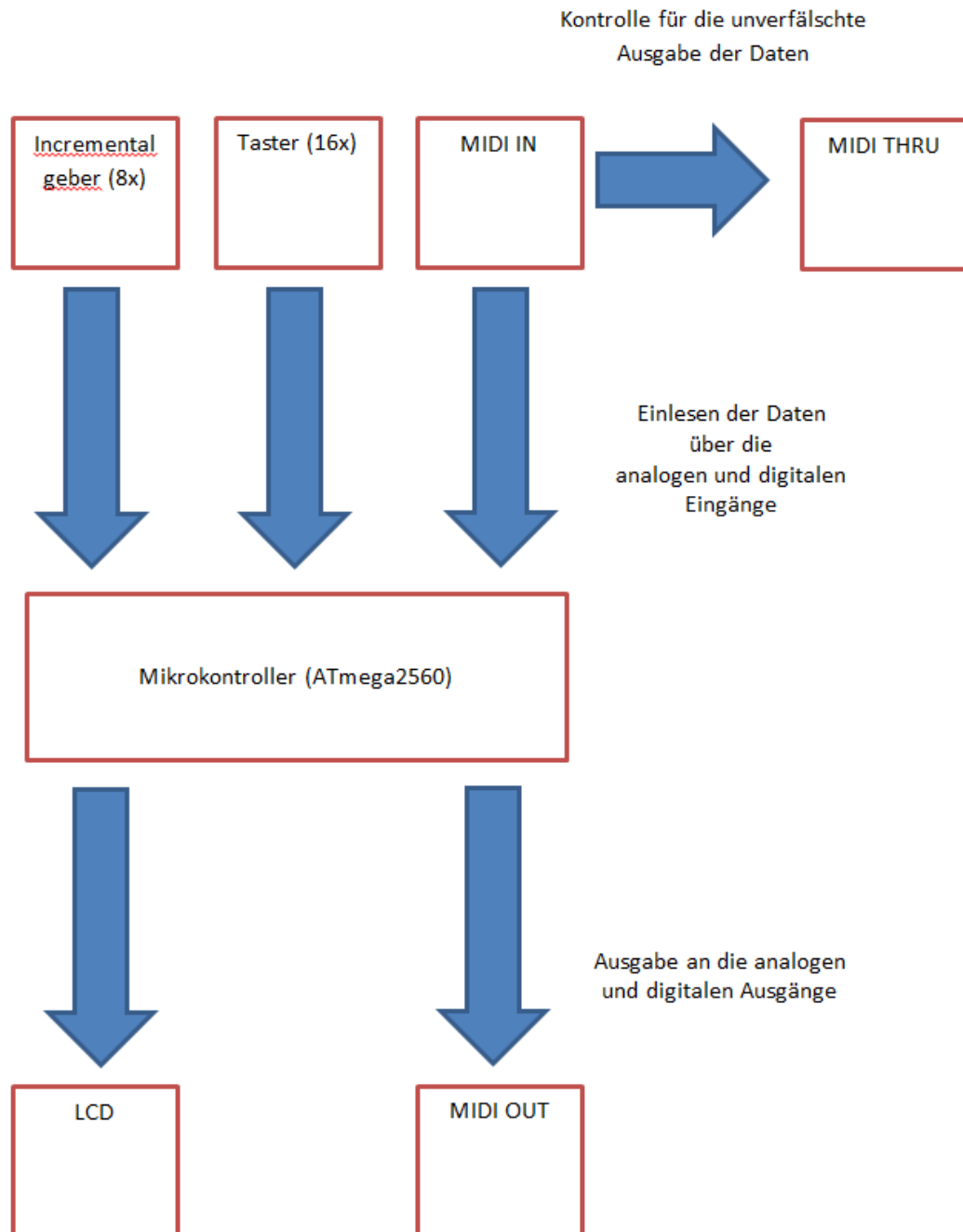


Abb. 2-1: Blockschaltbild zur Kurzbeschreibung

3. Hardware

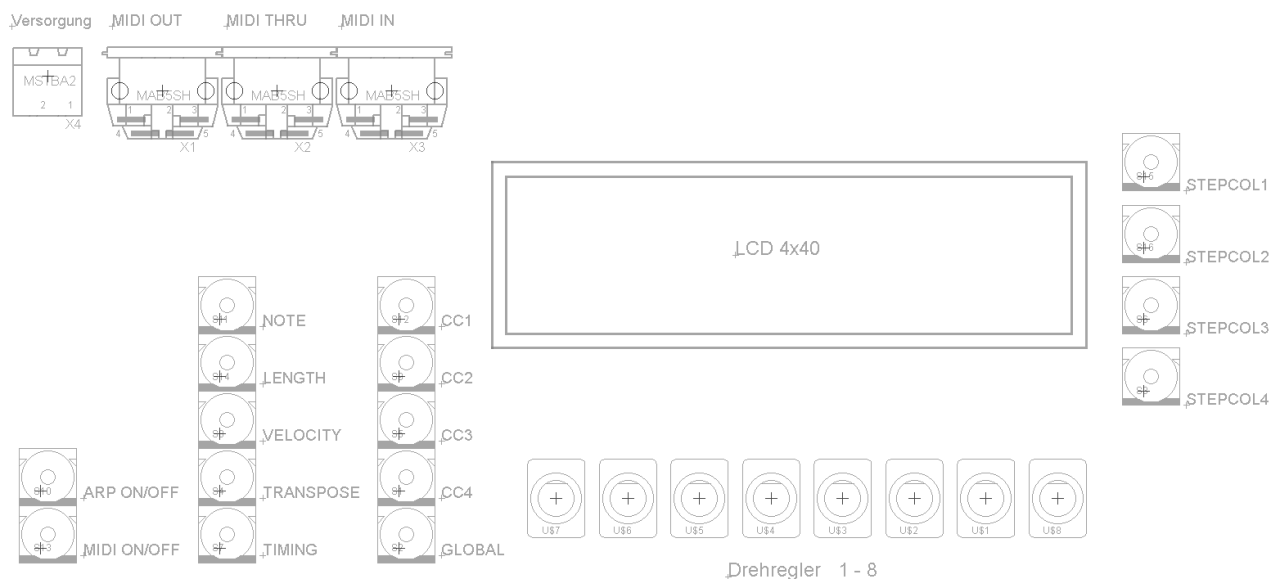


Abb. 3-1: Ansicht der Tasten, Drehreglers etc.

Herkömmliche Stepsequenzer und Arpeggiatoren erlauben elektronischen Musikern nur rudimentäre Gestaltungsmöglichkeiten ihrer Notenfolgen. Das bei dieser Diplomarbeit entwickelte Gerät schließt diese Lücke, indem es präzise, dem Takt angepasste Stepsequenzen erzeugt, die vom gespielten Akkord abhängig gemacht werden können und von der Programmierbarkeit der Abfolge völlig frei sind. Folgende Parameter können für eine maximal 32 Schritte lange Sequenz programmiert werden: Notenlänge (length), Lautstärke (velocity), Tonhöhe (pitch), Transponierung (trans) und vier Controllerdatensequenzen (CC). Weiteres kann das Tempo extern oder intern vorgegeben werden.

Das Board kann außerdem mit AC oder DC versorgt werden.

Es gibt 16 Funktionstasten und 8 Drehregler, welche im folgenden Text grau hinterlegt sind.

Die nachfolgenden Beispiele zeigen die Ausgabe am LCD wenn man einen der Funktionstaster drückt.

Note kurz gedrückt + Drehen eines Drehreglers: #1...#8

Mit Halten von Note + Drehen eines Drehreglers: C0...C7

Note kurz gedrückt + Drücken eines Drehreglers: on...off

Length gedrückt + Drehen eines Drehreglers: 1...256

Length gedrückt + Drücken eines Drehreglers: Glide on...off

Velocity gedrückt + Drehen eines Drehreglers: 1...127

Halten von CC (1-4) + Drehen des 1. Drehreglers: C1...C4

CC (1-4) gedrückt + Drehen eines Drehreglers: 0...127

CC (1-4) gedrückt + Drücken eines Drehreglers: akt. ... inakt.

Transpose gedrückt + Drehen eines Drehreglers: -24...+24

Transpose gedrückt + Drücken eines Drehreglers: on...off

Timing gedrückt + Drehen eines Drehreglers: -3...0...3

Timing gedrückt + Drücken eines Drehreglers: on...off

Global gedrückt + Drehen des 1. Drehreglers: PS: 1...128

Global gedrückt + Drücken des 2. Drehreglers: saved

Global gedrückt + Drücken des 3. Drehreglers: Rename

Global gedrückt + Drücken und Drehen des 3. Drehreglers: Cursor: Name

Global gedrückt + abermaliges Drücken des 3. Drehreglers beim gewünschten Buchstaben: A...Z, a...z

Global gedrückt + abermaliges Drehen des 3. Drehreglers: A...z,a...z

Global gedrückt + abermaliges Drücken des 3. Drehreglers: Cursor: Name

Global gedrückt + Drehen des 4. Drehreglers: C0...c7

Global gedrückt + Drehen des 5. Drehreglers: MIDI IN 1...16

Global gedrückt + Drehen des 6. Drehreglers: MIDI OUT 1...16

Bei den Tastern ARP ON/OFF und MIDI ON/OFF leuchtet bei aktivem Zustand die LED des Tasters, bei inaktivem Zustand leuchtet diese nicht.

Rechts des Displays befinden sich die 4 Taster mit denen man zwischen den Reihen des Displays wechseln kann:

- STEPCOL1
- STEPCOL2
- STEPCOL3
- STEPCOL4

Direkt links von diesen Tastern befinden sich die 4 LEDs welche anzeigen in welcher Reihe man sich gerade befindet.

Unter dem Display befinden sich 8 Drehregler mit denen man in der ausgewählten Spalte und durch Druck auf den gewünschten Taster, die speziellen Funktionen verändern kann.

Direkt über den Drehreglern befinden sich 8 LEDs durch die man sehen kann in welcher Spalte man sich befindet.

Über dem LCD befinden sich ebenfalls 8 LEDs die genau anzeigen wie das Programm durchgeführt wird. Ist es in Spalte 1, leuchtet die 1. LED, ist es in Spalte 2, leuchtet die 2. usw.

3.1. Allgemeine Beschreibung

Es existieren drei Eingangsbuchsen: MIDI – IN, MIDI – OUT, MIDI – Thru.

Das kleine Netzteil besteht aus einem Vollwellengleichrichter zum Schutz vor Verpolung und einem LM7805 zur Spannungsstabilisierung.

Beschreibung der Regler und ihrer Funktionen:

Auf der linken Seite des Displays befindet sich ein Tastenblock, der die speziellen Funktionen der 8 Drehgeber aufruft, dazu zählen:

1. **Note:** Notenummer im Akkord (z.B. #1=Grundton, #2=zweiter Ton, ... #8=achter Ton, Acc=Accord, Low=Lowest Ton, Hi=Highest Ton). Wird während des Drehens die Taste Note gehalten, stellt man einen Fixton (C0 – C7) ein.
Erklärung: Unter einer Note versteht man die absolute Tastennummer des Keyboards z.B. C4.
Durch Drücken wird die Note ein-/ausgeschaltet.
2. **Length:** Hier wird die Länge der Note in „MIDI Ticks“ eingestellt (1 – 256). Überlappt diese Note mit dem nächsten Step wird trotzdem ein NoteOn geschickt, die zu spielende Länge darf jedoch nicht gekürzt werden.
Durch Drücken wird Glide aktiviert, was bedeutet, dass bis zur nächsten Note kein NoteOff gesendet wird.
3. **Velocity:** Hier wird die Anschlagstärke der Note eingestellt (1-127). Die Accent-Funktion wird noch nicht implementiert.
4. **Continious (1-4):** 4 Taster für 4 Spuren. Durch gedrückt halten des Tasters CC1 bis CC4 und Drehen des ersten Reglers, wählt man den jeweiligen Continious Controller an. Durch Drehen des jeweiligen Reglers stellt man den Wert des jeweiligen Continious Controller ein. Durch Drücken des jeweiligen Reglers wird das Senden aktiviert/deaktiviert.
Hinweis: Die CC Daten werden vor den Noten gesendet.
5. **Transpose:** Hier wird eingestellt um wie viel Halbtöne die Note des Steps erhöht oder verkleinert wird (-24 bis +24). Durch Drücken wird die Transpose Funktion der jeweiligen Note ein-/ausgeschaltet
6. **Timing:** Im Timing Mode verschiebt man mit dem Regler den Startpunkt jeweils um einen Tick nach links oder nach rechts. Maximal um +-3 Ticks.
Durch Drücken wird die Timing Funktion der jeweiligen Note ein-/ausgeschaltet.
7. **Global:** Preset (Regler 1): Durch Drehen am Regler sucht man das gewünschte Preset und durch Drücken des Drehreglers lädt man das gewünschte Preset (128 Presets).
Save (Regler 2): Durch Drücken wird eine Sequenz gespeichert um sie später erneut aufrufen zu können.
Rename (Regler 3): Durch Drücken kommt man in den Mode Rename.
 - Durch Drehen kann man via Cursor zwischen den Buchstaben wechseln.
 - Durch abermaliges Drücken wählt man den zu ändernden Buchstaben aus.

- Durch abermaliges Drehen kann man den Buchstaben beliebig verändern.
- Durch abermaliges Drücken kommt man wieder in den Cursor Mode.

Basisnote (Regler 4): Hier wird die Basisnote C0 – c7 gewählt.

MIDI - Empfangskanal (Regler 5): Hier wird der MIDI – Kanal eingestellt an dem MIDI – Daten empfangen werden (1-16, all).

MIDI - Sendekanal (Regler 6): Hier wird der MIDI – Kanal eingestellt an dem MIDI Daten gesendet werden (1-16).

Ebenfalls auf der linken Seite des Displays befinden sich die beiden Taster für:

- 1. ARP ON/OFF (Ausgangsfiler):** Hier schaltet man die Arpeggiator Funktion ein/aus. Bei eingeschaltetem Arpeggiator werden die empfangenen MIDI Noten zerlegt, bei ausgeschaltetem Arpeggiator werden sie weitergeschickt. Grundsätzlich werden alle MIDI Daten die am Input empfangen werden, an MIDI Out weitergeleitet, mit Ausnahme der MIDI Noten ON/OFF Befehle. Ist der ARP ON/OFF Schalter ausgeschaltet, werden alle Befehle weitergeleitet
- 2. MIDI ON/OFF (Eingangsfiler):** Wird im laufenden Betrieb auf off geschaltet, werden keine weiteren MIDI ON Befehle angenommen
Hinweis: ARP ON/OFF und MIDI ON/OFF sind Teile des Presets und werden mit gespeichert/geladen.

Rechts des Displays befinden sich die 4 Taster mit denen man zwischen den Reihen des Displays wechseln kann:

1. STEPCOL_1
2. STEPCOL_2
3. STEPCOL_3
4. STEPCOL_4

Direkt links von diesen Tastern befinden sich die 4 LEDs welche anzeigen in welcher Reihe man sich gerade befindet.

Unter dem Display befinden sich 8 Drehregler mit denen man in der ausgewählten Spalte und durch Druck auf den gewünschten Taster, die speziellen Funktionen verändern kann.

Direkt über den Drehreglern befinden sich 8 LEDs durch die man sehen kann in welcher Spalte man sich befindet.

Über dem LCD befinden sich ebenfalls 8 LEDs die genau anzeigen wie das Programm durchgeführt wird. Ist es in Spalte 1, leuchtet die 1. LED, ist es in Spalte 2, leuchtet die 2. usw.

3.2. Arduino MEGA 2560

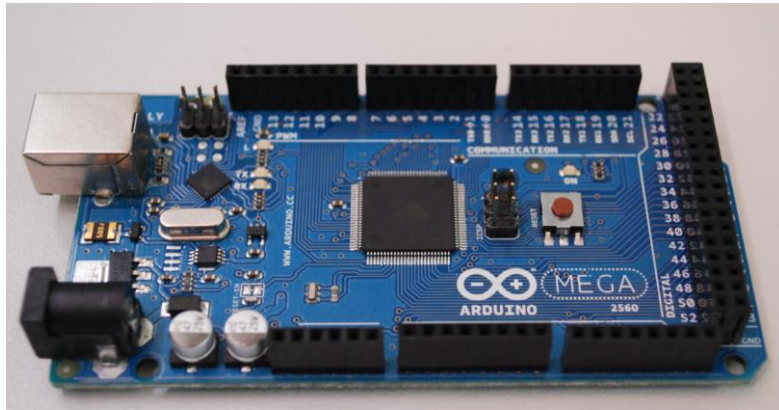
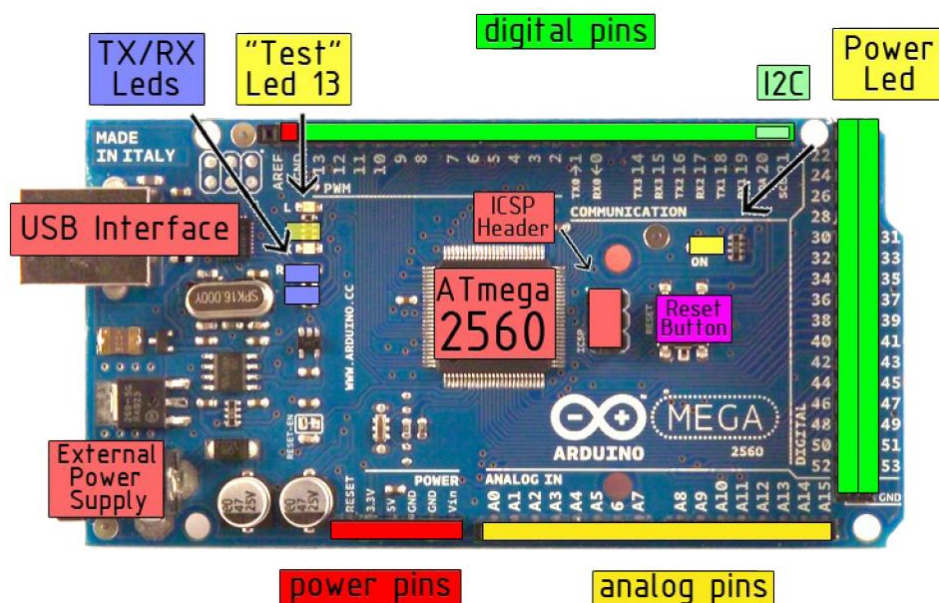


Abb. 3-2: Arduion MEGA 2560

3.2.1. Allgemeines

Das Arduino MEGA 2560 ist ein Mikrocontroller-Board, basierend auf dem ATmega2560.

Es hat 54 digitale Input/Output Pins, 16 analoge Inputs, 4 UARTs, einen 16 MHz Oszillator, eine USB Schnittstelle usw. Es besitzt alles, um einen Mikrocontroller zu supporten. Man kann es einfach am PC anschließen und es sofort nach Belieben programmieren. Das MEGA ist mit den meisten Shields, die für den Arduino Duemilanove oder Diecimila designt wurden, kompatibel.



1

Abb. 3-3: Pinbelegung des Arduino-Boards

¹ QuelleF1

3.2.1.1. Technische Spezifikationen

Mikrocontroller	ATmega2560
Operating Voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 kB (8 kB for Bootloader)
SRAM	8 kB
EEPROM	4 kB
Clock Speed	16 MHz

Tabelle 3-1: Technische Spezifikationen des Arduino Boards

3.2.2. Microcontroller AVR ATmega2560

2



Abb. 3-4: AVR ATmega2560

3.2.2.1. Allgemeines

Der Mikrocontroller ist ein ATMEGA 2560 des Herstellers AVR, welcher bereits auf dem verwendeten Arduino Board vorhanden ist. Es handelt sich um einen 8 Bit- Mikrocontroller. Er bietet außerdem einen Flashspeicher mit 256kByte. Der ATMEGA 2560 besitzt mehrere Interrupts sowie auch Unmengen von verwendbaren Pins zur Beschaltung, was sehr wichtige Kriterien für die Auswahl darstellten, da durch die einzelnen Einstellungen einige Interrupts benötigt sowie die einzelnen Pins zum Zusammenschalten der Bauelemente verwendet werden.

3.2.2.1.1. Eigenschaften

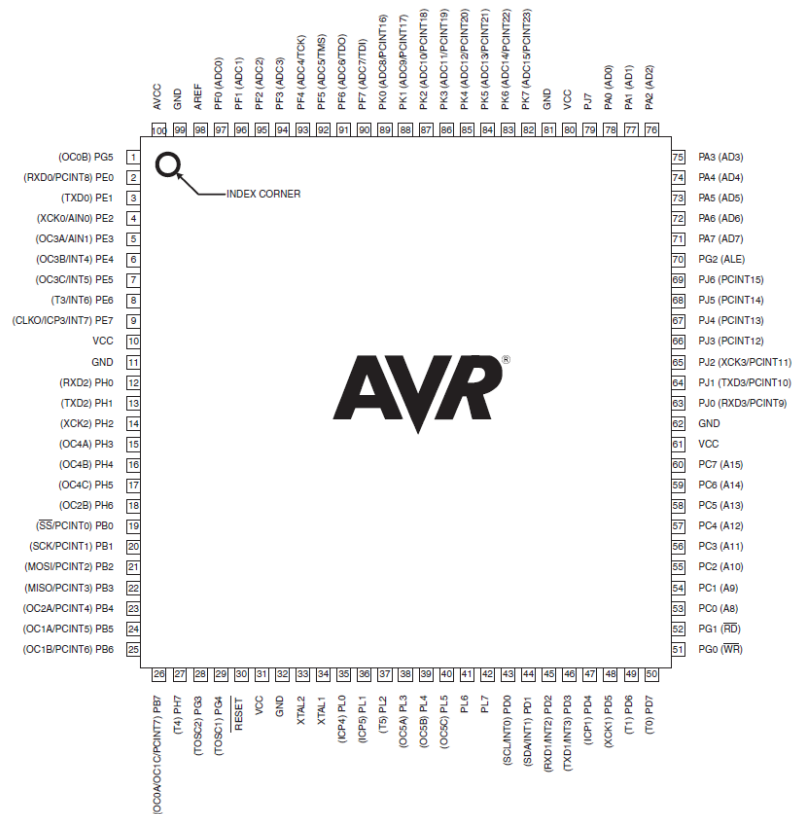
Symbol	Name	Min	Max	Unit
V_{IL}	Input Low Voltage	-0.5	0.2V	V
V_{IH}	Input High Voltage	0.7	$V_{CC}+0.5$	V
V_{OL}	Output Low Voltage	-	0.9	V
V_{OH}	Output High Voltage	4.2	-	V
I_{IL}	Input Low Leakage	-	1	μA
I_{IH}	Input High Leakage	-	1	μA

Tabelle 3-2: Eigenschaften des AVR ATmega2560

Wird mit +5V Versorgt, I_{CC} sollte maximal 14mA betragen.

² QuelleR1

3.2.2.1.2. Pin-Konfiguration



3

Abb. 3-5: Pinbelegung des AVR ATmega2560

3.2.2.1.3. Pinbelegung

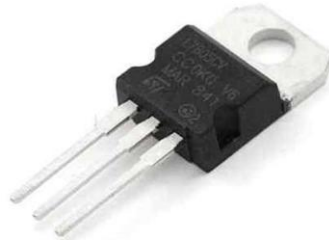
Bauteil	AVR	Arduino	Bauteil	AVR	Arduino
Incr			LCD		
INCR1A	PK0	A8	DB7	PA7	D29
INCR2A	PK1	A9	DB6	PA6	D28
INCR3A	PK2	A10	DB5	PA5	D27
INCR4A	PK3	A11	DB4	PA4	D26
INCR5A	PK4	A12	EN2	PA3	D25
INCR6A	PK5	A13	EN1	PA2	D24
INCR7A	PK6	A14	RW	PA1	D23
INCR8A	PK7	A15	RS	PA0	D22

³ QuelleR2

INCR1B	PC0	D37	Taster		
INCR2B	PC1	D36	Taster mit Led1	PF0	A0
INCR3B	PC2	D35	Taster mit Led2	PF1	A1
INCR4B	PC3	D34	Taster mit Led3	PF2	A2
INCR5B	PC4	D33	Taster mit Led4	PF3	A3
INCR6B	PC5	D32	Taster mit Led5	PF4	A4
INCR7B	PC6	D31	Taster mit Led6	PF5	A5
INCR8B	PC7	D30	Taster mit Led7	PF6	A6
			Taster mit Led8	PF7	A7
INCR1S	PG5	PWM4	Taster mit Led9	PL0	D49
INCR2S	PE3	PWM5	Taster mit Led10	PL1	D48
INCR3S	PH3	PWM6	Taster mit Led11	PL2	D47
INCR4S	PH4	PWM7	Taster mit Led12	PL3	D46
INCR5S	PH5	PWM8	Taster mit Led13	PL4	D45
INCR6S	PH6	PWM9	Taster mit Led14	PL5	D44
INCR7S	PB4	PWM10	Taster mit Led15	PL6	D43
INCR8S	PB5	PWM11	Taster mit Led16	PL7	D42
MIDI			AS1100		
MIDI (IN)	PD2	RX1	AS1100LOAD	PB0	53
MIDI (OUT)	PD3	TX1	AS1100CLK	PB1	52
MIDI ON	PE4	PWM2	AS1100DIN	PB2	51
MIDI OFF	PE5	PWM3			

Tabelle 3-3: Pinbelegung des Arduinos bzw. AVR's

3.3. Spannungsregler LM7805



4

Abb. 3-6: LM7805

3.3.1. Allgemeines

Sehr viele Bauelemente arbeiten in der Elektronik im TTL- Bereich. In diesem Bereich sind die Spannungspegel von 0 Volt bis 5 Volt. Dies ist mit dem Fixspannungsregler LM7805, welcher die Eingangsspannung ausgangsseitig auf 5 Volt herabregelt, sehr gut realisierbar. Die Eingangsspannung sollte zwischen 7 Volt und 35 Volt liegen. Je höher die Input-Spannung gewählt wird, desto mehr überschüssige Energie wird in Wärme umgewandelt. Bei einer zu hohen Wärmebildung muss der LM7805 mit einem geeigneten Kühlkörper gekühlt werden.

3.3.1.1. Pinbelegung

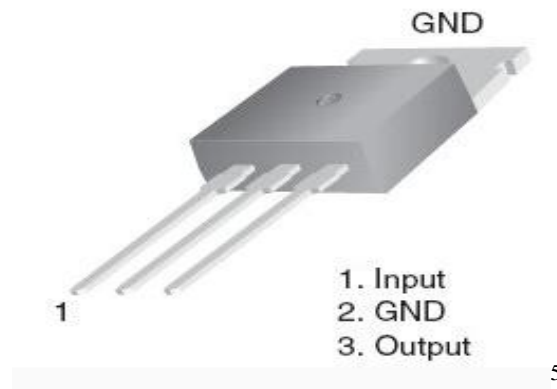


Abb. 3-7: Pinbelegung des LM7805

⁴ QuelleR3

⁵ QuelleR4

3.3.1.2. Berechnung der Wärmebildung

$$P_V = (V_{IN} - V_{OUT}) * I_{OUT} + V_{OUT} * I_R$$

$$P_V = (12 - 5) * 1 + 5 * 10^{-3} * 5 = 7,025W$$

➔ Es bildet sich eine enorme Wärme am Spannungsregler. Dadurch wird meist ein Kühlblech benötigt

3.3.1.3. Berechnung des Kühlblechs

$$R_{TH} = \frac{T_{STG} - T_U}{P_V} - R_{TJC}$$

$$R_{TH} = \frac{150 - 25}{7,025} - 5 = 12,7936 \text{ } ^\circ\text{C/W} \rightarrow \text{Sollte der Kühlkörper abführen können}$$

3.3.2. Beschaltung

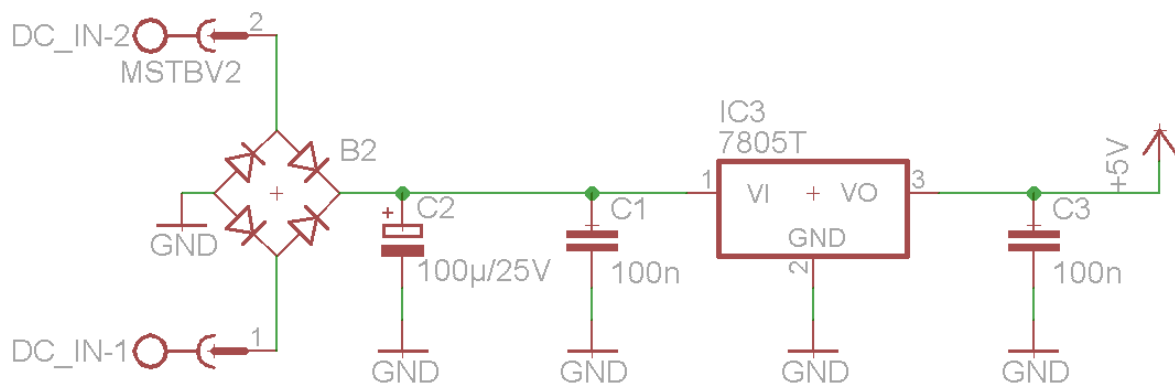


Abb. 3-8: Beschaltung des LM7805

Die Kondensatoren am Ein- und Ausgang des LM7805 können auftretende Spannungsschwankungen kompensieren.

3.4. 6N139 - Optokoppler



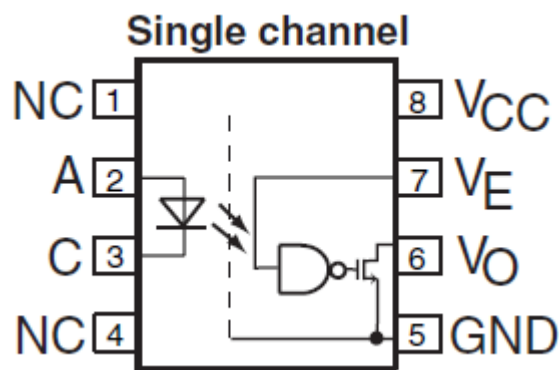
6

Abb. 3-9: 6N139

3.4.1. Allgemeines:

Der 6N139 ist ein Optokoppler, der durch die galvanische Trennung Masseschleifen am MIDI- Eingang vermeidet, ansonsten kann es zum sogenannten Brummen kommen. Außerdem dient der Optokoppler zur Sicherung, dass im schlechtesten Fall - durch einen Kurzschluss - nicht die gesamte Hardware zerstört wird.

3.4.1.1. Pinbelegung



7

Abb. 3-10: Pinbelegung des 6N139

⁶ QuelleR5

⁷ QuelleR6

3.5. AS 1100

3.5.1. Allgemeines



Abb. 3-11: AS1100

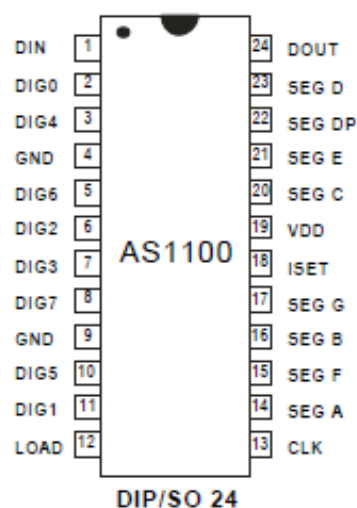
Der AS1100 kann zur Ansteuerung einer 7 Segment Anzeige oder auch für eine 8x8 LED- Matrix verwendet werden. Er kann über eine übliche 4 Draht serielle Schnittstelle programmiert werden.

Der AS1100 verfügt über einen BCD code-B Decoder und einen 64Bit Memory, welcher zum Speichern der LED- Einstellungen dient. Jede LED beziehungsweise jedes Segment kann separat angesprochen werden. Es wird nur ein externer Widerstand benötigt, um den Strom einzustellen. Somit kann man mit einem AS1100 bequem einzelne Einstellungen über eine LED –Matrix anzeigen lassen.

Der AS1100 hat einen sehr geringen Standby-Strom mit nur 20 μ A und im Betrieb weniger als 50 μ A. Er kann über Software und einen externen Clock zurückgesetzt werden.

Um bei einer LED-Matrix die einzelnen LEDs separat anzusteuern, sind die Reihen mit den Segment Divers verbunden und die Spalten mit DIG 0-7

3.5.1.1. Pinbelegung



8

Abb. 3-12: Pinbelegung des AS1100

⁸ QuelleR7

Register	Address					Hex
	D15-D12	D11	D10	D9	D8	Code
No-Op	X	0	0	0	0	0xX0
Digit 0	X	0	0	0	1	0xX1
Digit 1	X	0	0	1	0	0xX2
Digit 2	X	0	0	1	1	0xX3
Digit 3	X	0	1	0	0	0xX4
Digit 4	X	0	1	0	1	0xX5
Digit 5	X	0	1	1	0	0xX6
Digit 6	X	0	1	1	1	0xX7
Digit 7	X	1	0	0	0	0xX8
Decode Mode	X	1	0	0	1	0xX9
Intensity	X	1	0	1	0	0xXA
Scan Limit	X	1	0	1	1	0xXB
Shutdown	X	1	1	0	0	0xXC
Not used	X	1	1	0	1	0xXD
Reset and ext. Clock	X	1	1	1	0	0xXE
Display Test	X	1	1	1	1	0xFF ⁹

Abb. 3-13: Adressen des AS1100

3.5.1.2. Technische Spezifikationen

Beschreibung	Symbol	Min	Typ	Max	Einheit
Operating Temp Range	T_{AMB}	-40	-	+85	°C
Operating Supply Voltage	V_{DD}	4.0	5.0	5.5	V
Shutdown Supply Current	I_{DDSD}	-	20	50	µA
Operating Supply Current	I_{DD}	-	300	-	mA
Display Scan Rate	f_{OSC}	500	800	1300	HZ
Digit Drive Sink Current	I_{DIGIT}	320	-	-	mA
Segment Drive Source Current	I_{SEG}	-30	-40	-45	mA
Segment Drive Current Matching	ΔI_{SEG}	-	3.0	-	%
Digit Drive Source Current	I_{DIGIT}	-2	-	-	mA
Segment Drive Sink Current	I_{SEG}	5	-	-	mA

Tabelle 3-4: Technische Spezifikationen des AS1100

⁹ Quelle R8

3.5.1.3. Beschaltung

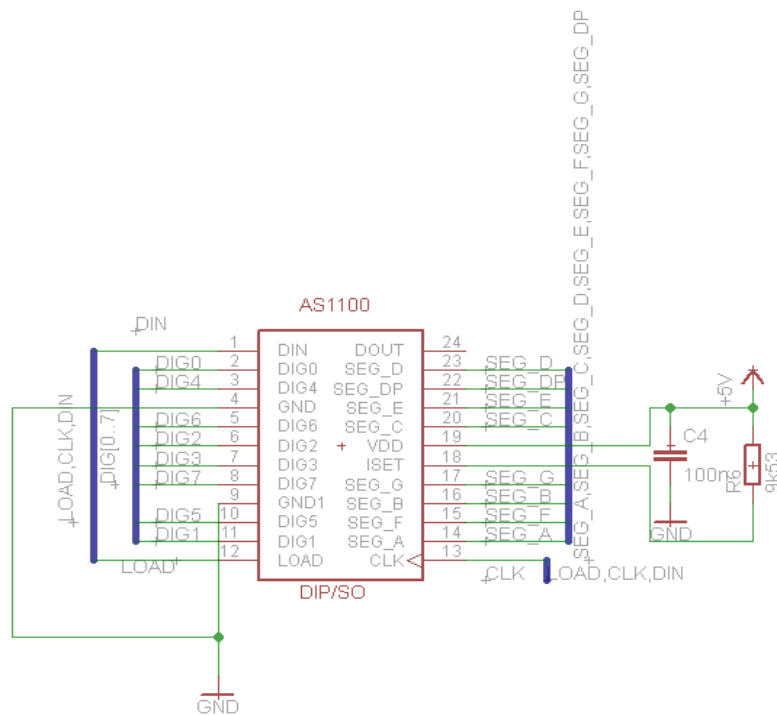


Abb. 3-14: Beschaltung des AS1100

3.6. LC- Display

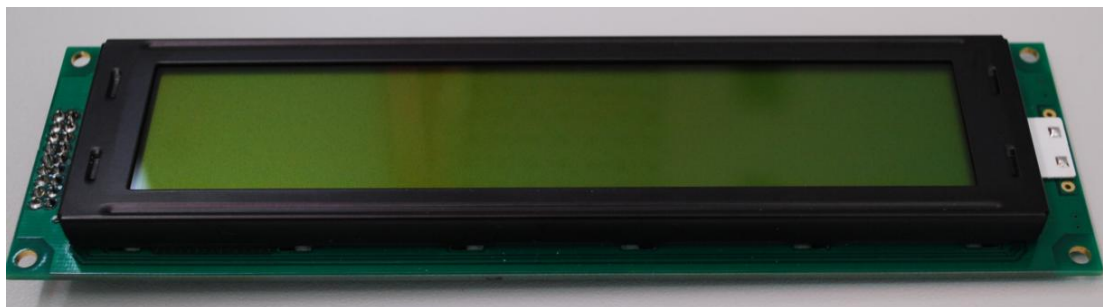


Abb. 3-15: LC-Display 4x40

3.6.1. Allgemein

Es handelt sich um ein 40x4 alphanumerisches LCD

3.6.1.1. Technische Spezifikationen

Beschreibung	Symbol		Min	Typ	Max	Einheit
Input Spannung	Vdd	+5V	4.7	5.0	5.5	V
	Vdd	+3.3V	2.7	3.0	3.3	V
Versorgungstrom	Idd	Vdd = 5V	---	3.5	4.0	mA
Vorgeschlagene LCD Betriebsspannung	Vdd-V0	0 °C	4.7	5.0	5.5	V
		25 °C	4.3	4.5	4.7	V
		50 °C	4.1	4.3	4.5	V
LED forward Spannung	Vf	25 °C	---	4.2	4.6	V
LED forward Strom	If	25 °C	---	300	---	mA

Tabelle 3-5: Technische Spezifikationen des LC- Displays

3.6.1.2. Pinbelegung

LCD	LCD-Pin	AVR-Pin	ARDUINO-Pin	Funktion
DB7	1	PA7	D29	H/L Data Bus Line
DB6	2	PA6	D28	H/L Data Bus Line
DB5	3	PA5	D27	H/L Data Bus Line
DB4	4	PA4	D26	H/L Data Bus Line
DB3	5	NC	NC	H/L Data Bus Line
DB2	6	NC	NC	H/L Data Bus Line
DB1	7	NC	NC	H/L Data Bus Line
DB0	8	NC	NC	H/L Data Bus Line
EN1	9	PA2	D24	Chip Enable Signal
R/W	10	PA1	D23	H/L Read/Write
RS	11	PA0	D22	Register Select
V0	12	GND	GND	Contrast Select
VSS	13	GND	GND	GND
VDD	14	5V	5V	Power Supply
EN2	15	PA3	D25	Chip Enable Signal
NC	16	NC	NC	No Connection
A	17	5V	5V	Backlight
K	18	GND	GND	Backlight

Tabelle 3-6: Pinbelegung des LC- Displays

3.6.2. Beschaltung

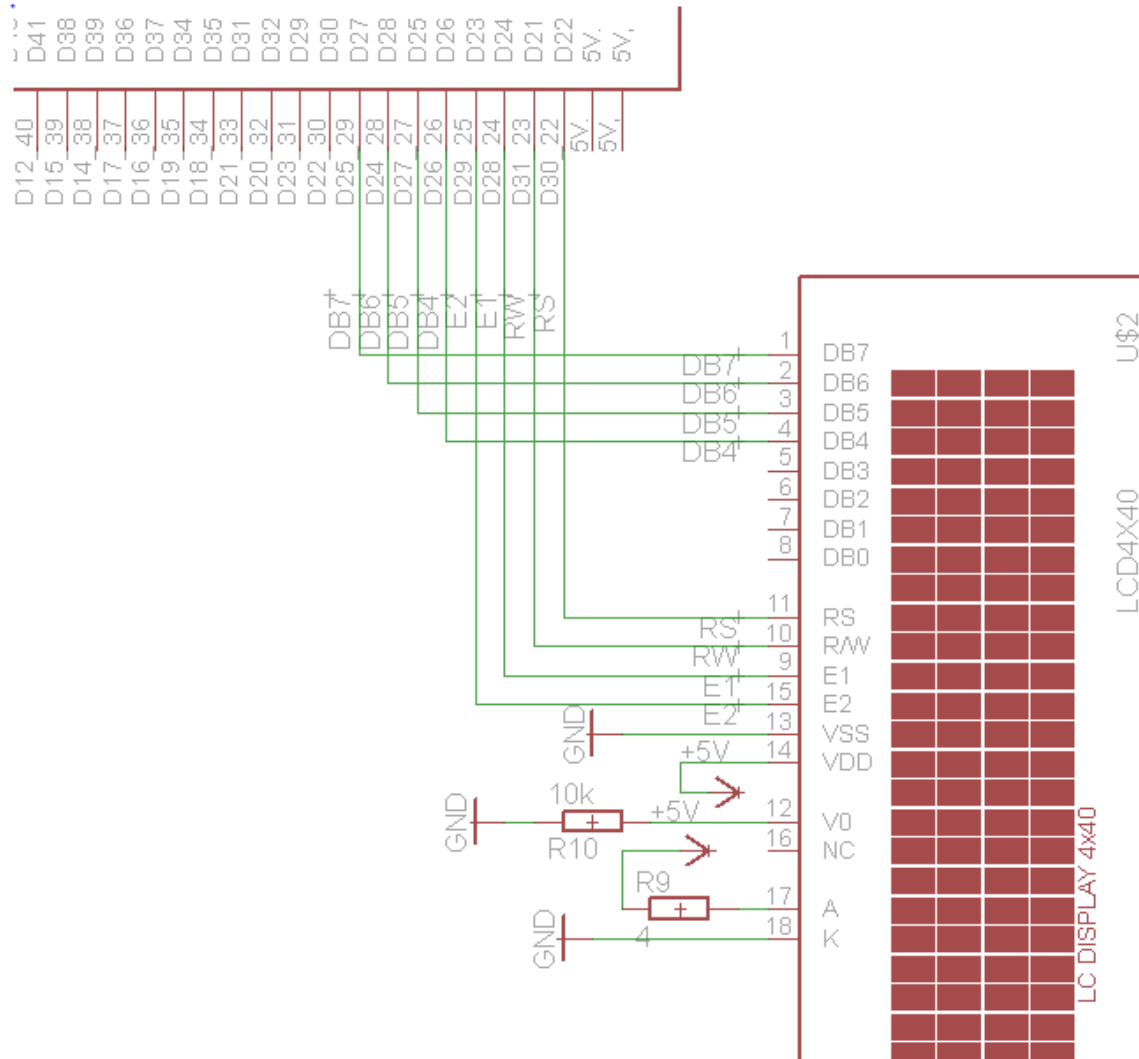


Abb. 3-16: Beschaltung des LCDs

3.7. 74HCT14



Abb. 3-17: 74HCT14

3.7.1. Allgemeines

Der 74HCT14 ist ein High – Speed Si – Bgate CMOS Gerät und ist Pin kompatibel mit Low Power Schottky TTL.

Der 74HCT14 versorgt sechs invertierende Buffer mit Schmitt-Trigger Aktionen. Er ist fähig, langsam wechselnde Input Signale in scharf definierte, flimmerfreie Output Signale zu transformieren.

3.7.1.1. Pinbelegung

11

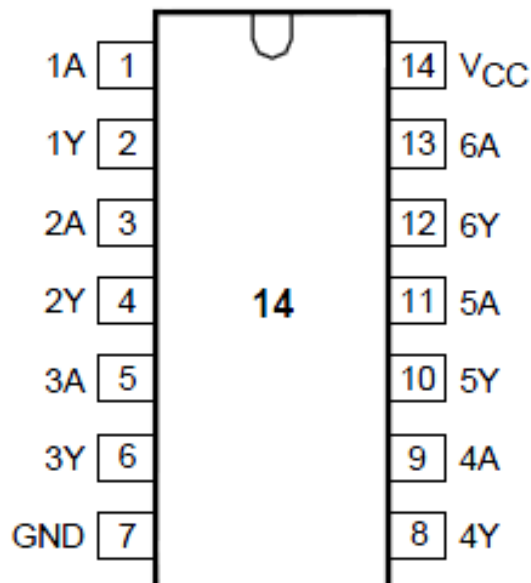


Abb. 3-18: Pinbelegung des 74HCT14

¹⁰ QuelleR9

¹¹ QuelleR10

3.8. MIDI -In/ -Out/ -Thru

3.8.1.1. Allgemeines

(engl.: musical instrument digital interface = „Digitale Schnittstelle für Musikinstrumente“) ist ein Datenübertragungs-Protokoll zum Zwecke der Übermittlung musikalischer Steuerinformationen zwischen elektronischen Instrumenten wie Keyboards, Synthesizern, Drum-Computern oder auch PCs bzw. Laptops. Es gibt drei MIDI-Anschlüsse mit unterschiedlichen Funktionen: MIDI-IN (Eingang), MIDI-OUT (Ausgang) und MIDI-THRU. Letzterer dient dazu, über den MIDI-Eingang empfangene Daten unbeeinflusst an ein weiteres MIDI-Gerät weiterzuleiten.

Drückt man beispielsweise auf einem Musik-Keyboard eine Taste, werden digitale Datenpakete mit den Informationen über die Tonhöhe, die Anschlagstärke sowie die Dauer des Drückens am MIDI-Ausgang (MIDI-OUT) des Keyboards ausgegeben und zum Beispiel an den MIDI-Eingang eines Computers übermittelt, welcher diese Informationen aufzeichnen, abspeichern, wiedergeben und auch in Form von Notenschrift auf dem Bildschirm sichtbar machen kann. Aufgezeichnete MIDI-Daten müssen daher normalerweise zunächst an ein MIDI-Instrument (also zum Beispiel ein Keyboard) zurückgesendet werden. Erst in dessen Klangerzeugermodul werden daraus hörbare Töne und Sounds erzeugt.

Das MIDI-Protokoll besteht aus Befehlen zur Steuerung der Klangerzeuger in elektronischen Instrumenten und Soundkarten. Solche Befehle sind beispielsweise Note-on („Taste für Note x wurde gedrückt“), Velocity („Anschlagsstärke“) und Note-off („Taste für Note x wurde wieder losgelassen“). So kann man über eine Tastatur ein anderes MIDI-Instrument spielen und dabei dessen Klänge hören.

3.8.1.2. Beschaltung

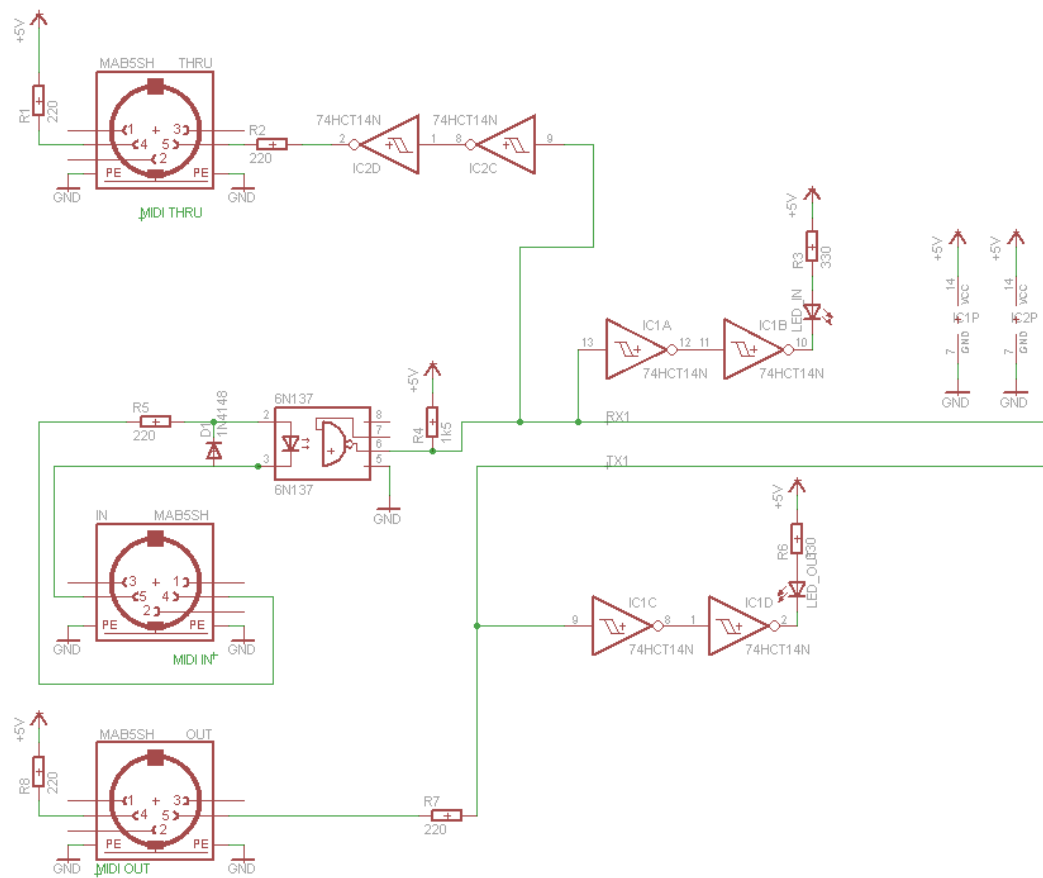


Abb. 3-19: Beschaltung des MIDI -In/ -Out/ -Thru

3.9. Platinen

3.9.1. Eagle- Library

3.9.1.1. *Arduino MEGA 2560*

3.9.1.1.1. Symbol

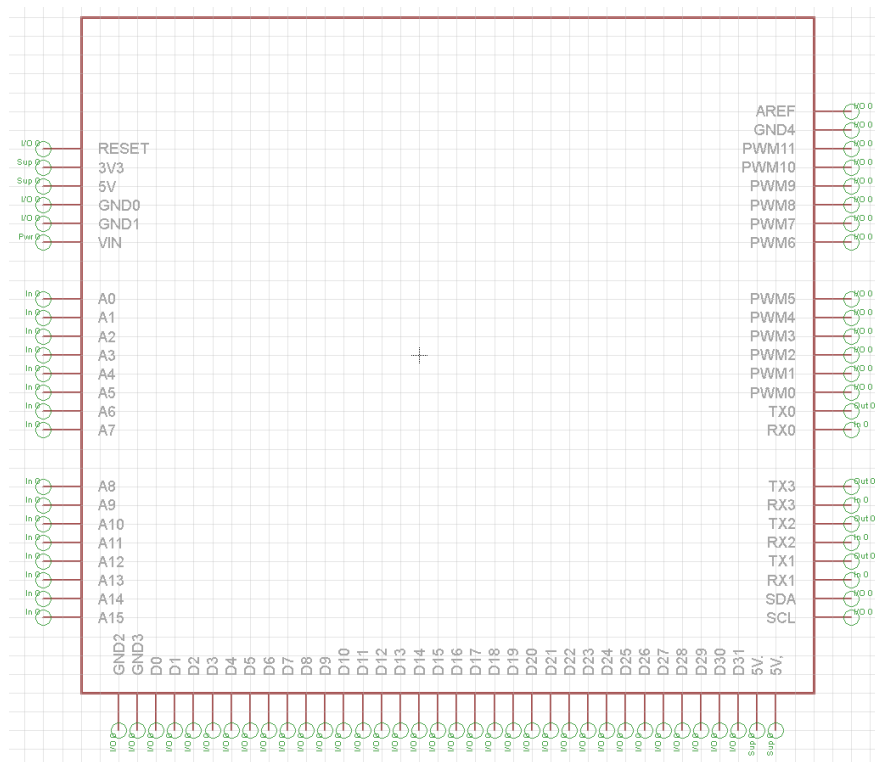


Abb. 3-20: Symbol des Arduino MEGA 2560

3.9.1.1.2. Package

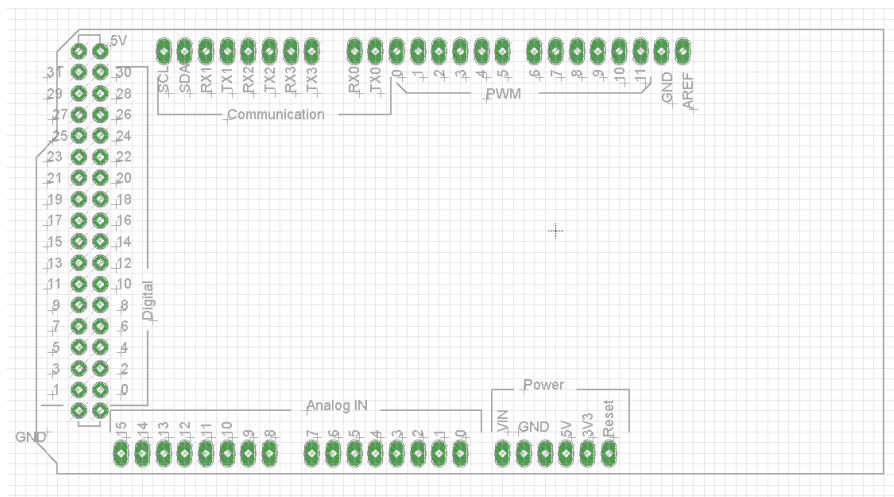


Abb. 3-21: Package des Arduino MEGA 2560

3.9.1.1.3. Device

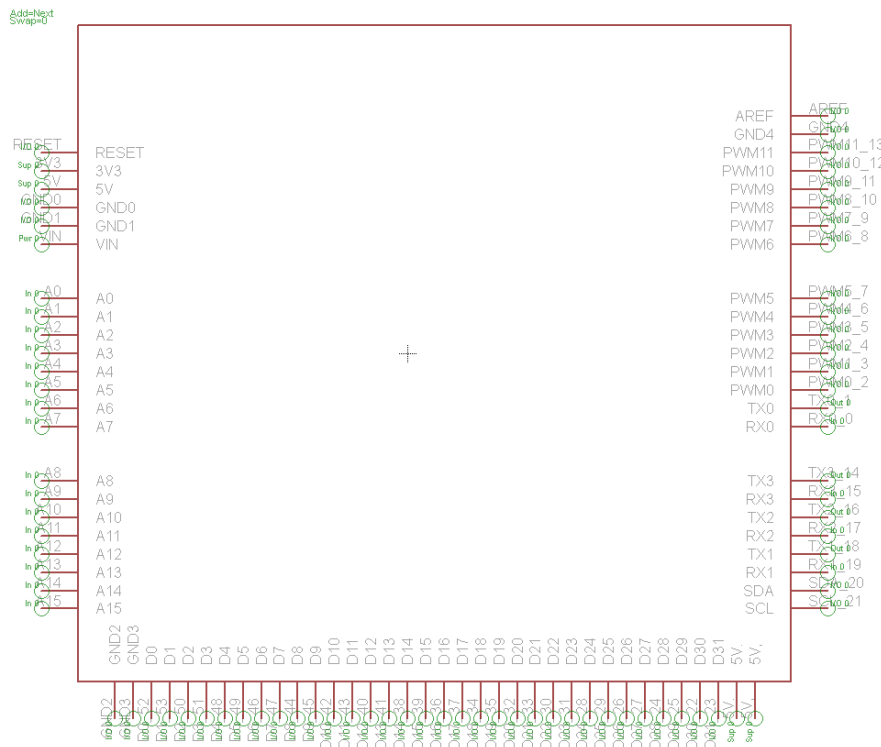


Abb. 3-22: des Arduino MEGA 2560

3.9.1.2. AS 1100

3.9.1.2.1. Symbol

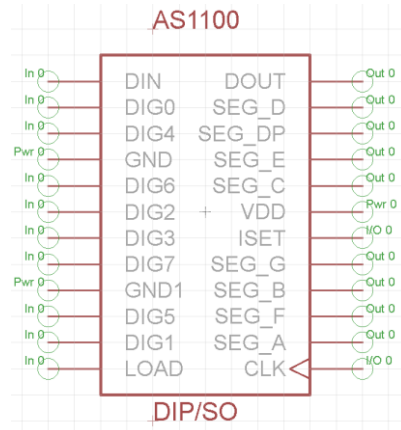


Abb. 3-23: Symbol des AS1100

3.9.1.2.2. Package

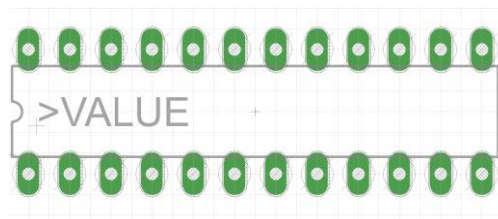


Abb. 3-24: Package des AS1100

3.9.1.2.3. Device

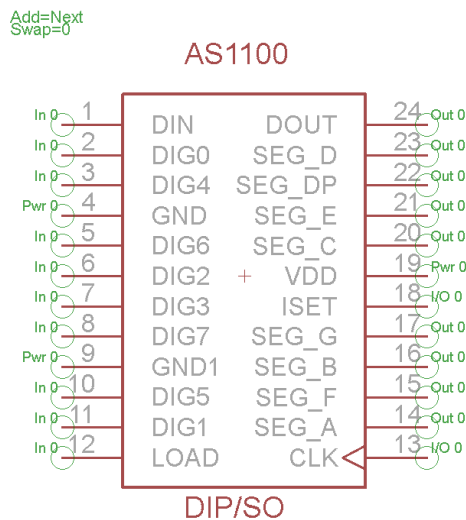


Abb. 3-25: Device des AS1100

3.9.1.3. LC- Display

3.9.1.3.1. Symbol

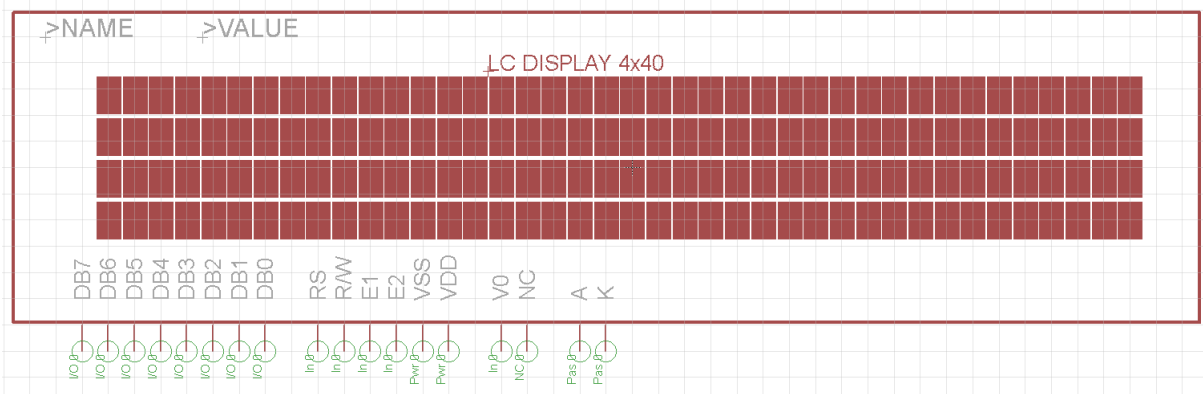


Abb. 3-26: Symbol des LC- Displays

3.9.1.3.2. Package

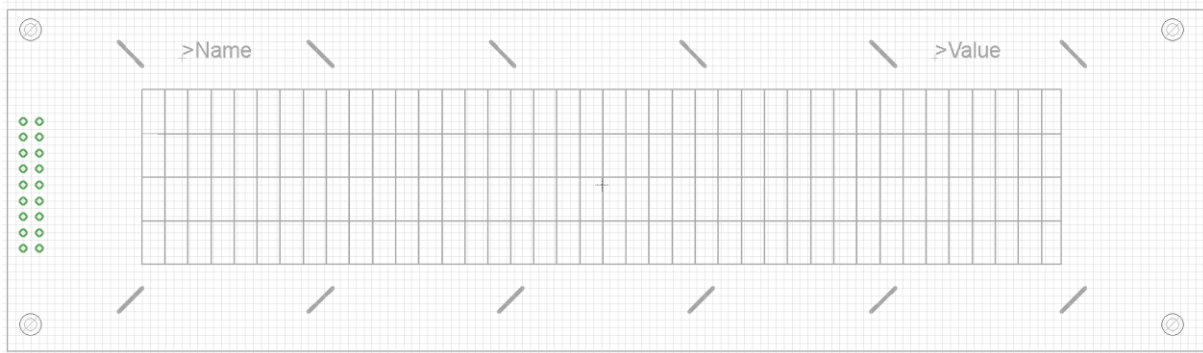


Abb. 3-27: Package des LC- Displays

3.9.1.3.3. Device

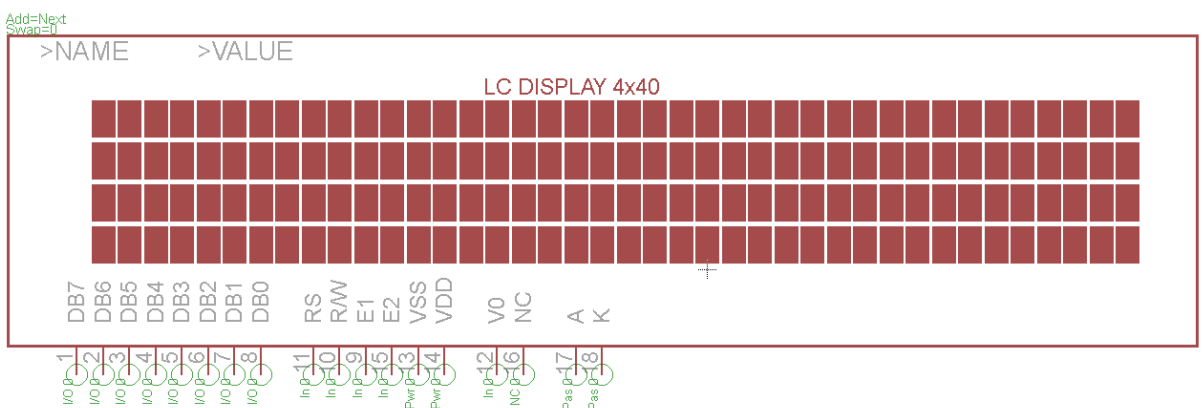


Abb. 3-28: Device des LC- Displays

3.9.1.4. LED- Taster

3.9.1.4.1. Symbol

3.9.1.4.1.1. LED

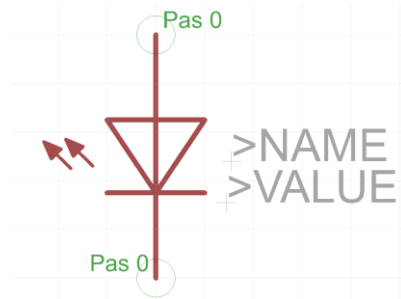


Abb. 3-29: Symbol der LED

3.9.1.4.1.2. Taster

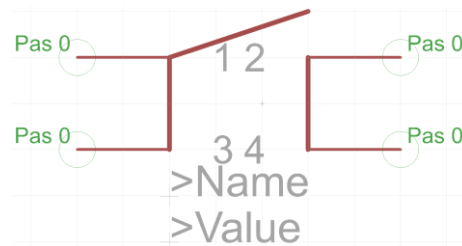


Abb. 3-30: Symbol des Tasters

3.9.1.4.2. Package

3.9.1.4.2.1. Taster ohne LED

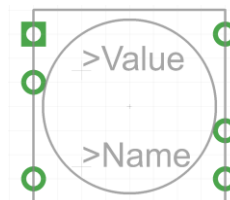


Abb. 3-31: Package des Tasters ohne LED

3.9.1.4.2.2. Taster mit LED

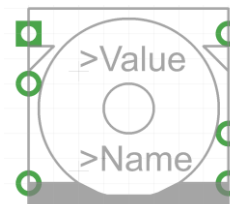


Abb. 3-32: Package des Tasters mit LED

3.9.1.4.3. Device

3.9.1.4.3.1. Taster ohne LED

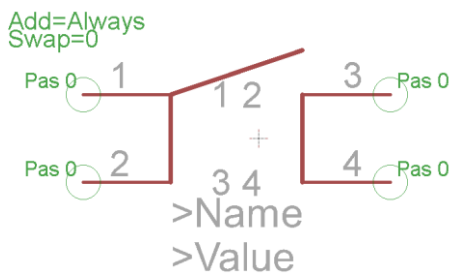


Abb. 3-33: Device des Tasters ohne LED

3.9.1.4.3.2. Taster mit LED

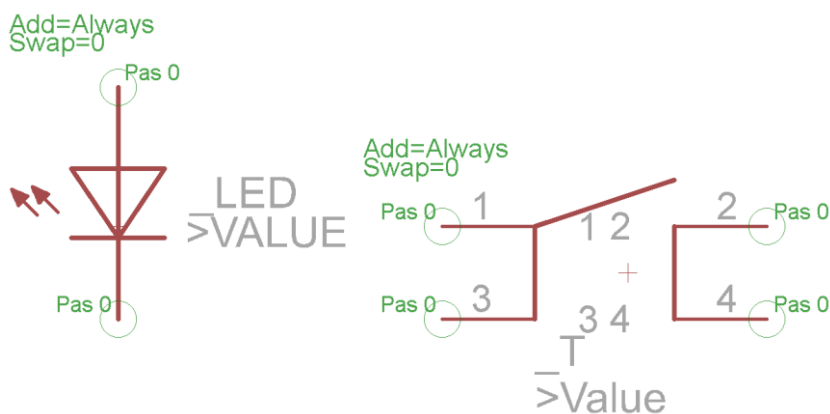


Abb. 3-34: Device des Tasters mit LED

3.9.1.5. Inkrementalgeber

3.9.1.5.1. Symbol

3.9.1.5.1.1. Inkrementalgeber

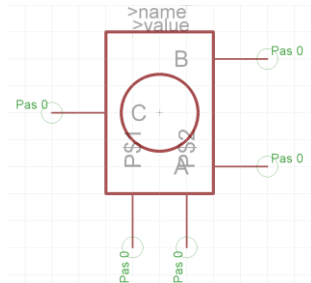


Abb. 3-35: Symbol des Inkrementalgebers

3.9.1.5.1.2. Switch

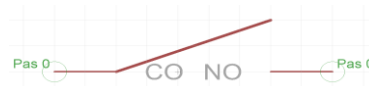


Abb. 3-36: Symbol des Switches

3.9.1.5.2. Package

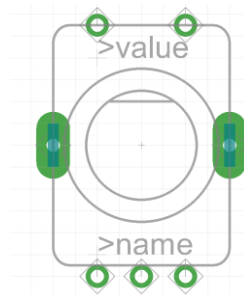


Abb. 3-37: Package des Inkrementalgebers

3.9.1.5.3. Device

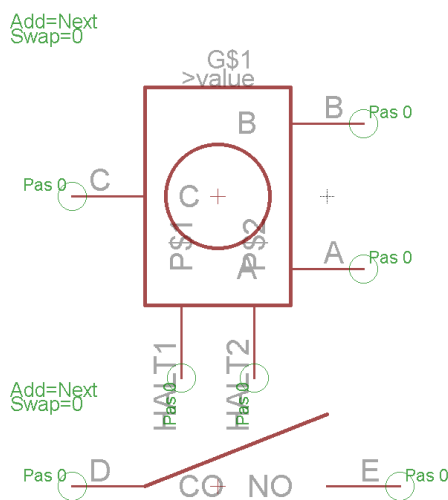


Abb. 3-38: Device des Inkrementalgebers

3.9.2. Schaltplan

3.9.2.1. Gesamtschaltung

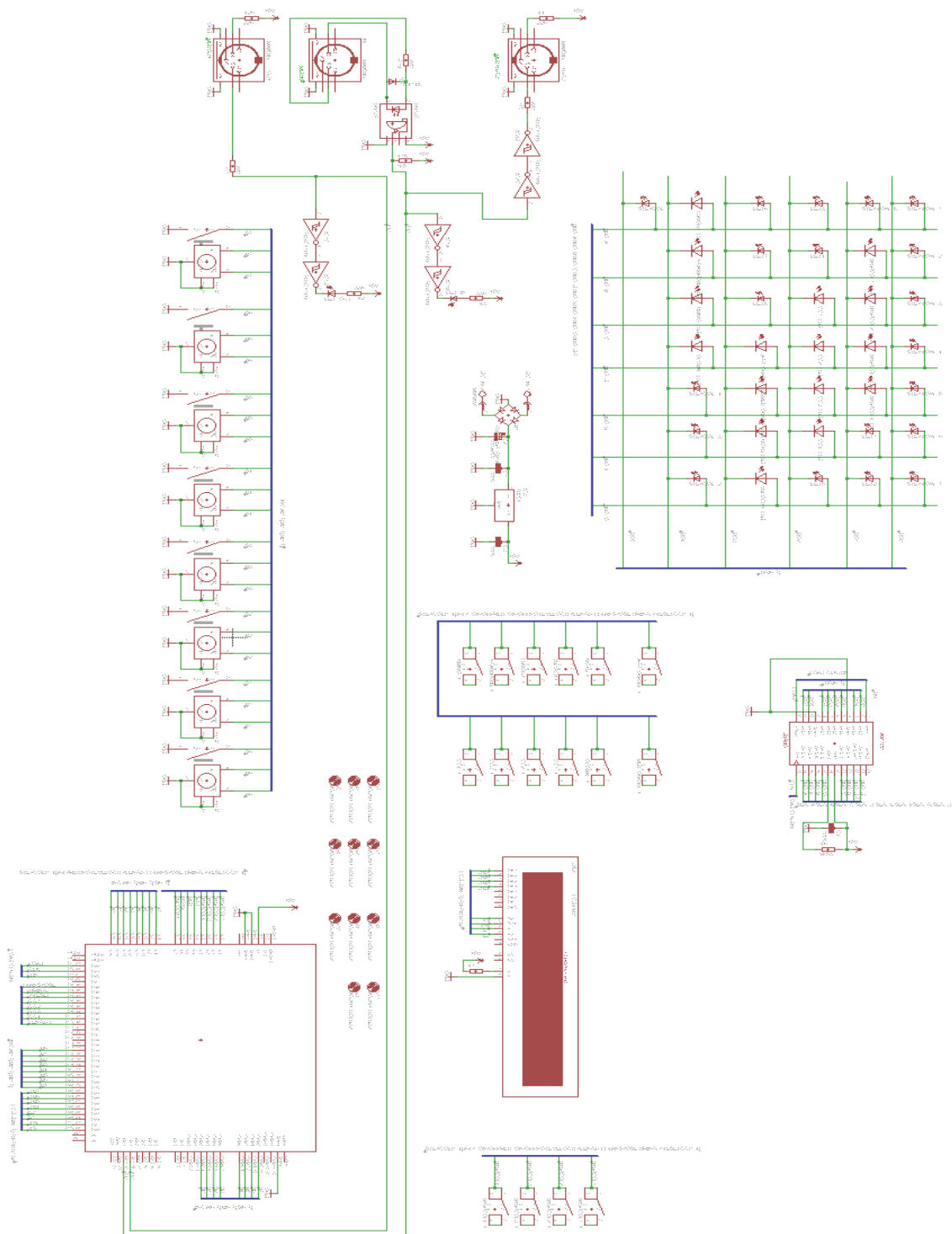


Abb. 3-39: Gesamtschaltung

3.9.2.2. Spannungsregler

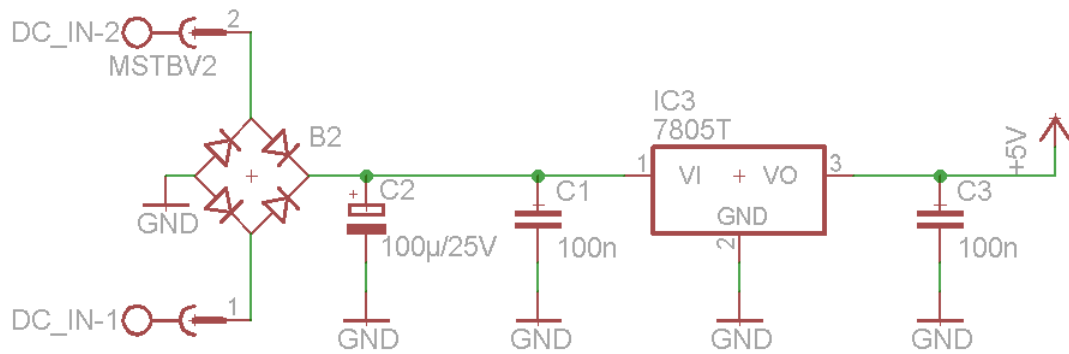


Abb. 3-40: Schaltung des Spannungsreglers

3.9.2.3. LC-Display

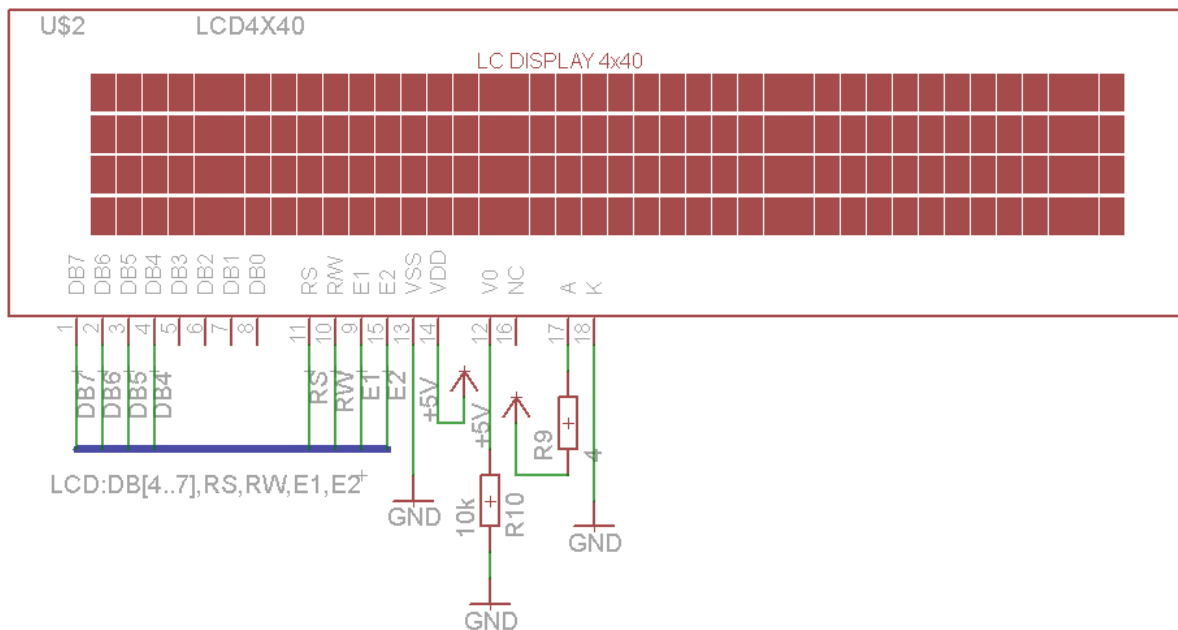


Abb. 3-41: Schaltung des LC- Displays

3.9.2.4. LED- Matrix

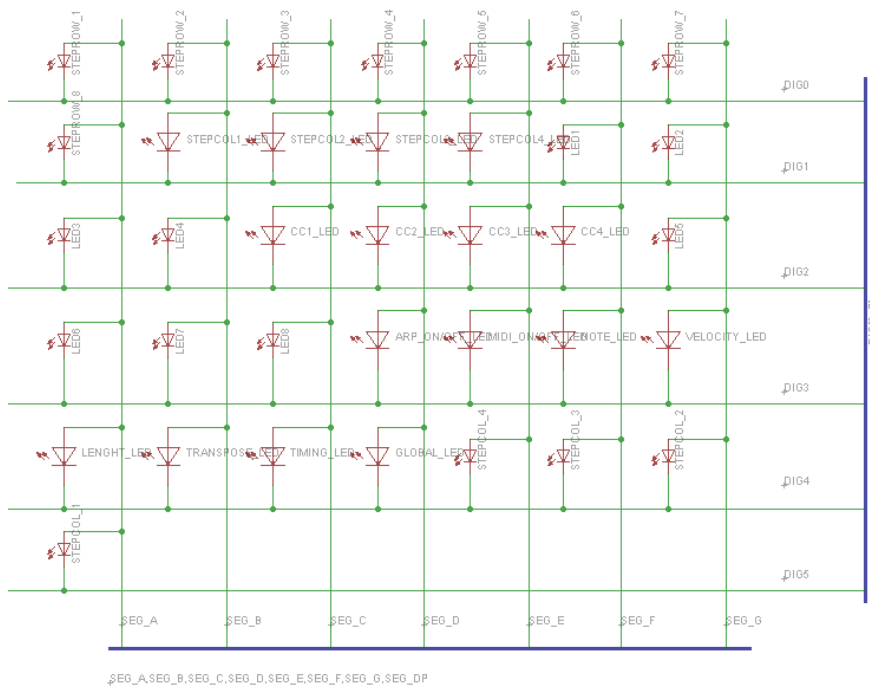


Abb. 3-42: Schaltung der LED- Matrix

3.9.2.5. AS1100

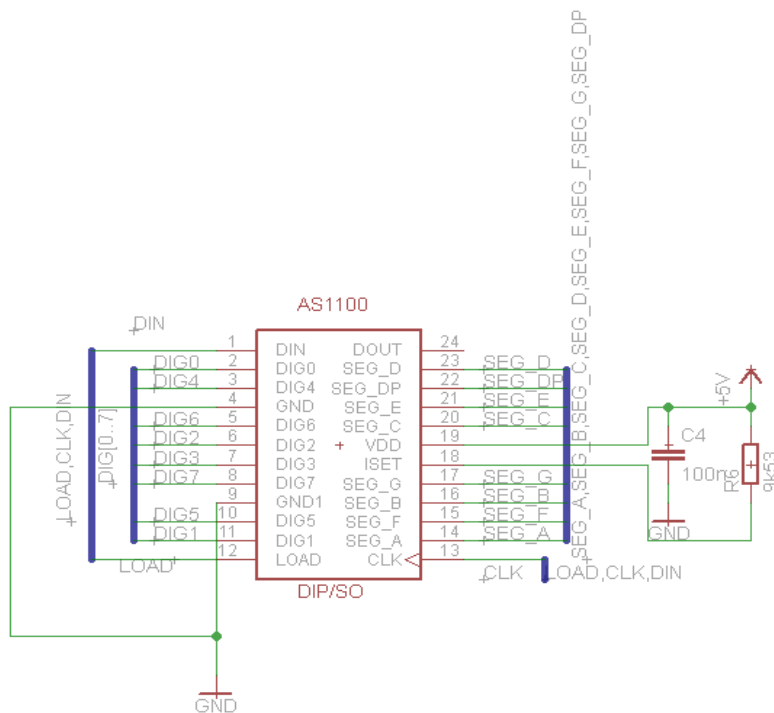


Abb. 3-43: Schaltung des AS1100

3.9.2.6. MIDI- IN/ -OUT/ -THRU

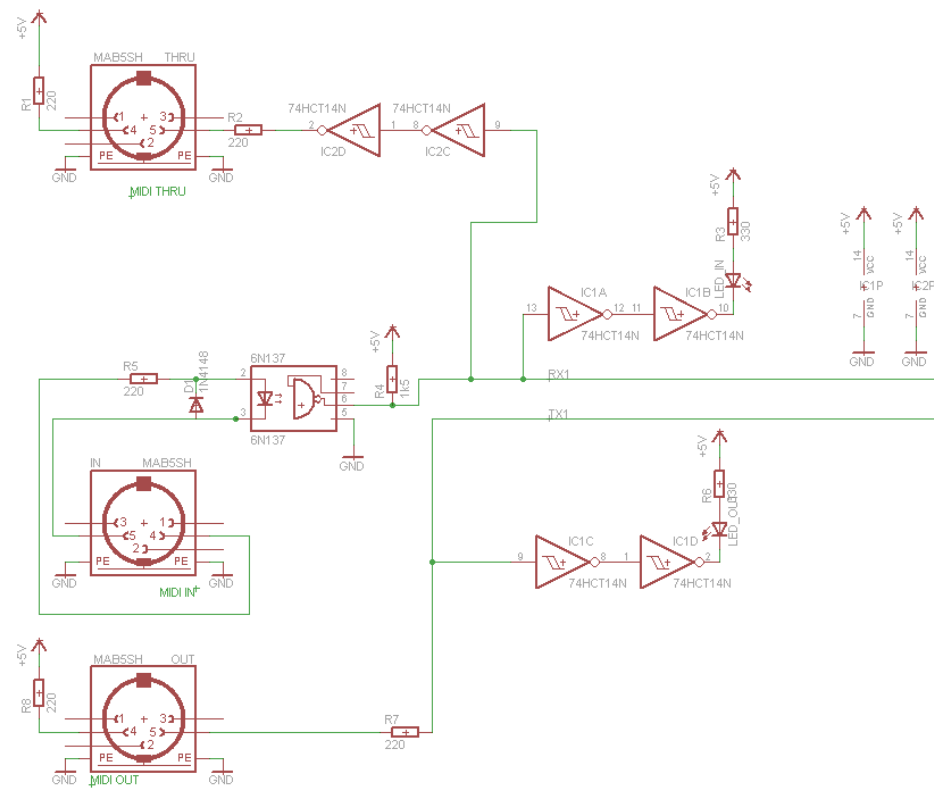


Abb. 3-44: Schaltung der MIDI -In/ -Out/ -Thru

3.9.3. Layout

Es wurde eine doppelseitige Platine verwendet.

Die Maße sind 303 x 104mm

3.9.3.1. Top - und Bottom - Layer

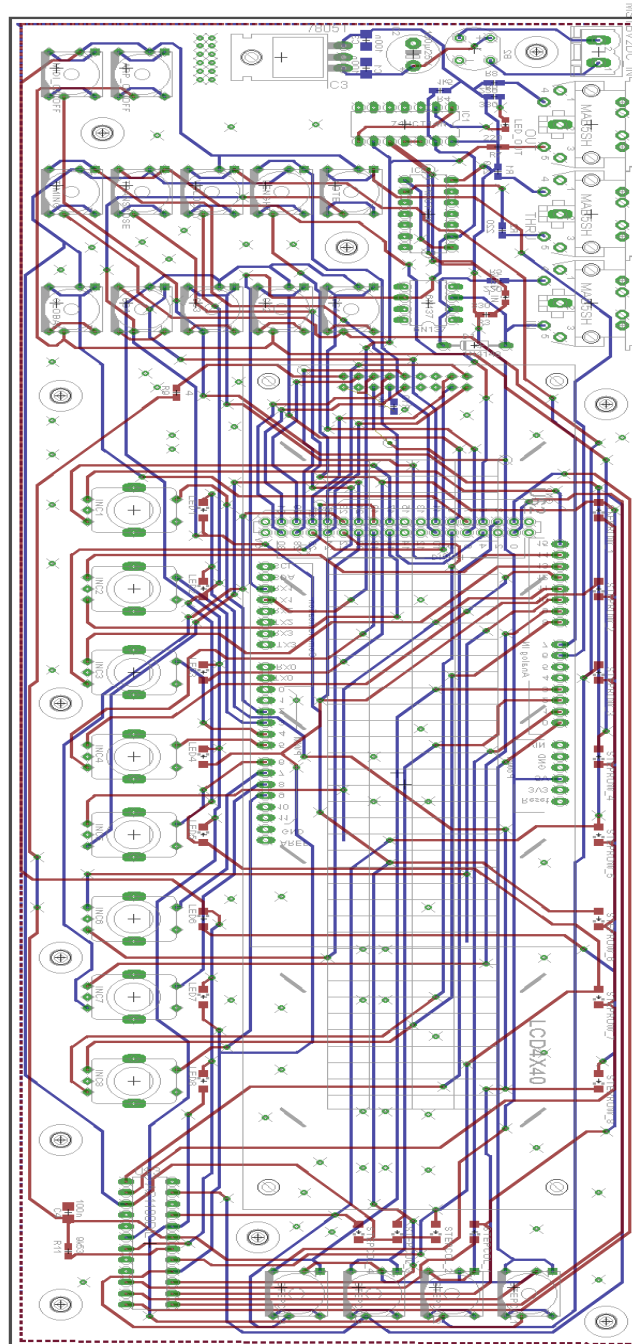


Abb. 3-45: Top- und Bottom- Layer

3.9.3.2. Top - Layer

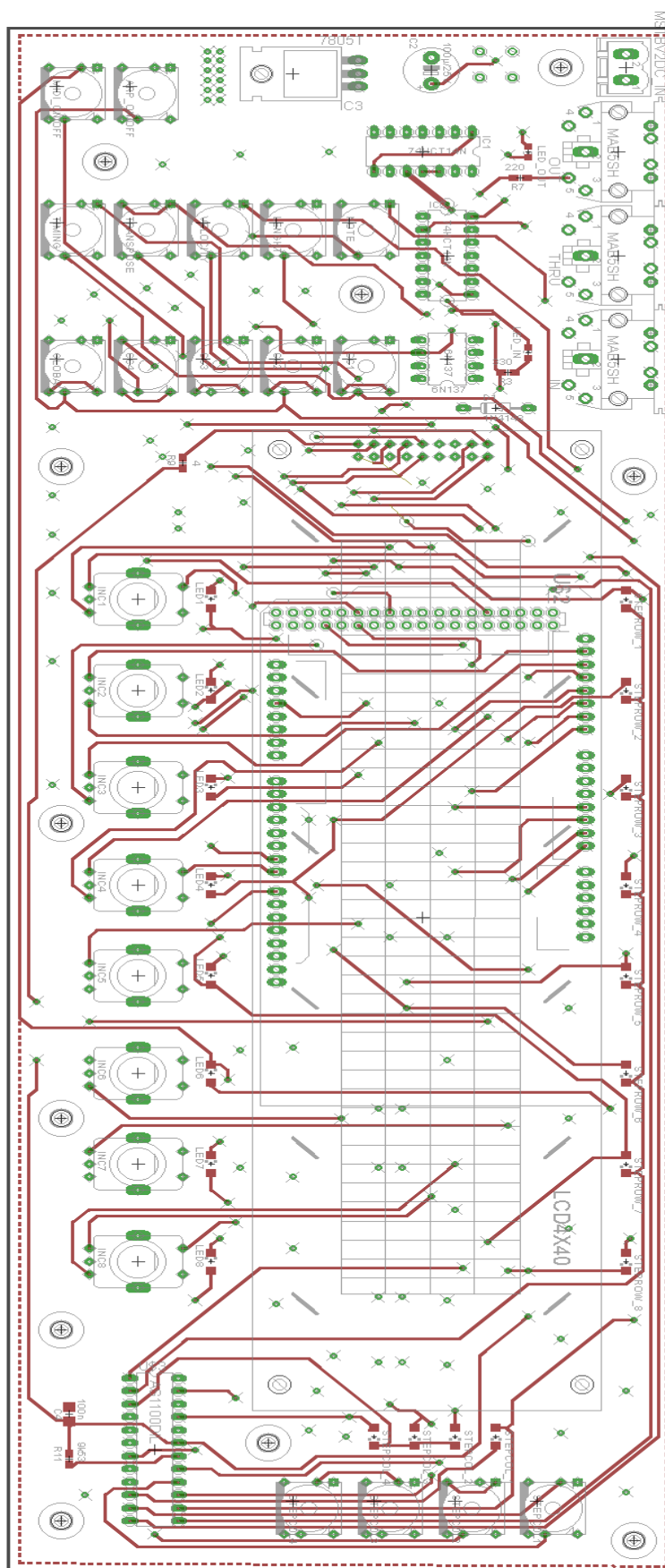


Abb. 3-46: Top-Layer

3.9.3.3. Bottom - Layer

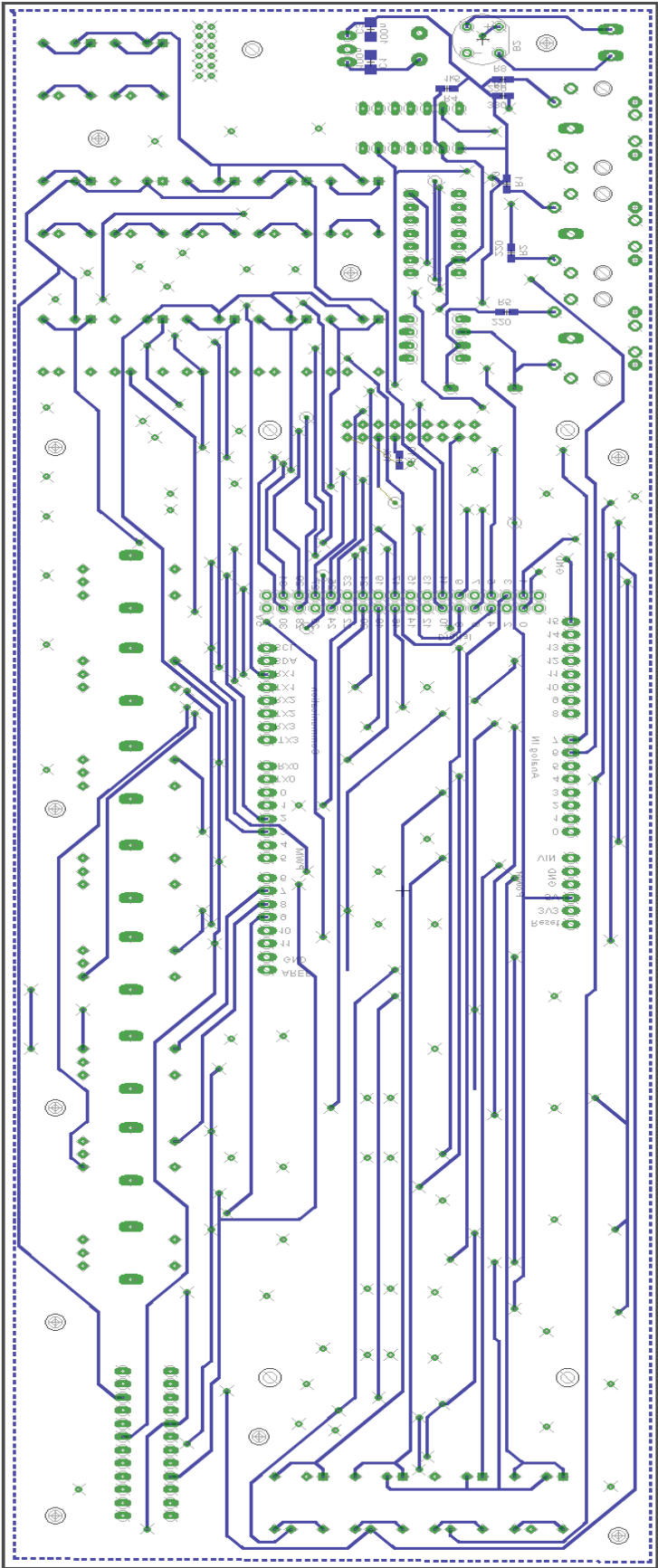


Abb. 3-47: Bottom- Layer

3.9.3.4. Bestückungsplan – Top

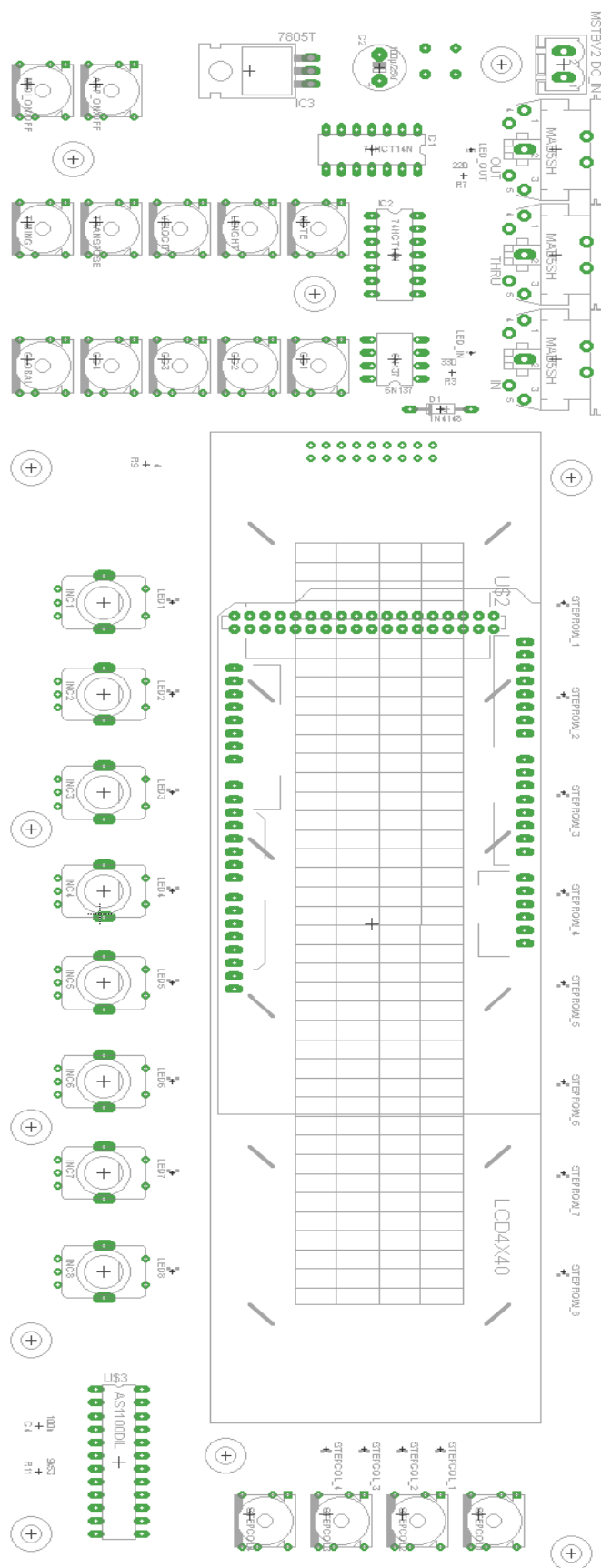


Abb. 3-48: Bestückungsplan Top

3.9.3.5. Bestückungsplan – Bottom

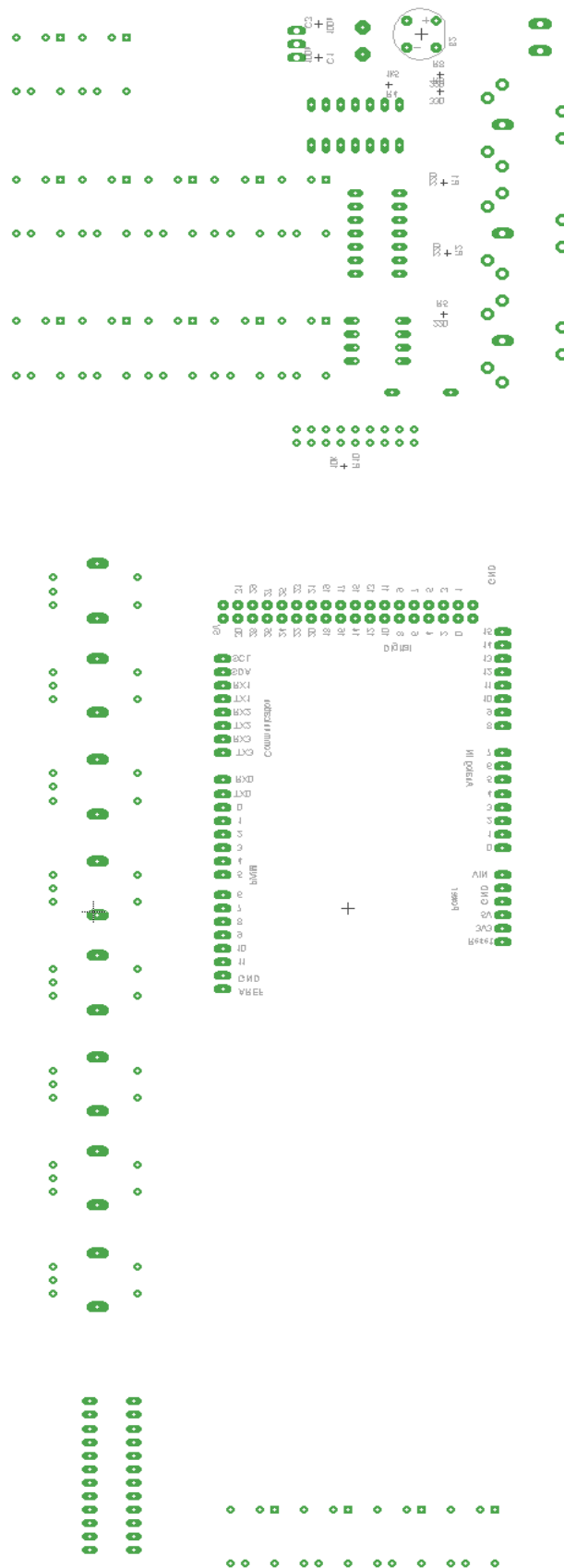


Abb. 3-49: Bestückungsplan Bottom

3.9.4. Aufbau

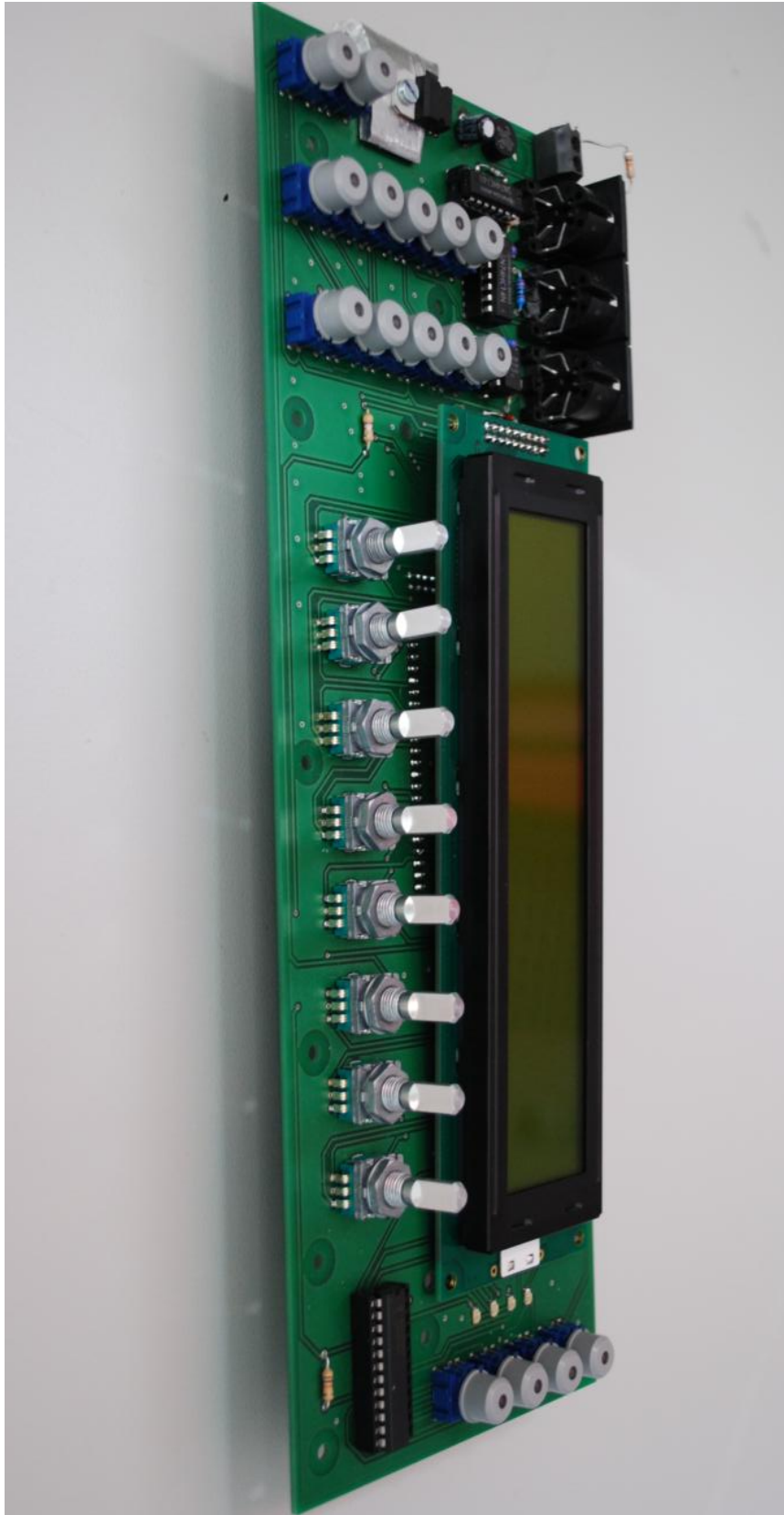


Abb. 3-50: Aufbau

4. Software

4.1. mySmartUSB Light



12

Abb. 4-1: mySmartUSB Light

4.1.1. Allgemein

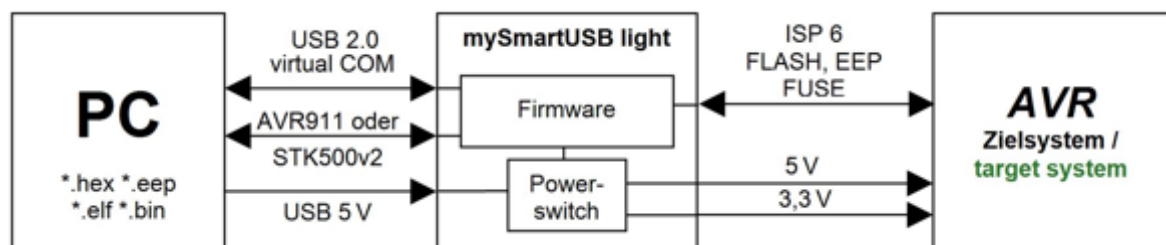
Der mySmartUSB Light Programmer ist ein reiner ISP-Programmer für Atmel AVR – Mikrocontroller im USB Stick Design. Er ist besonders schnell durch double buffering und hat Auto – ISP Speed.

Die Einbindung in das Betriebssystem und die Programmiersoftware läuft völlig Transparent über einen virtuellen COM – Port.

Die Standard Firmware des Programmers ist STK500 V2, man kann aber auch alternativ AVR911 / 910 / 109 darauf spielen.

Mit der STK500 V2 Firmware besitzt er einen automatischen Step-Down-Modus: für Controller, die auf langsameren Taktquellen unter 1 MHz umgeschaltet wurden und sich nicht mehr mit der Standard ISP Geschwindigkeit ansprechen lassen.

Ein großer Vorteil des mySmartUSB light Programmers ist, dass er besonders effizient mit den myAVR – Softwareprodukten arbeitet, aber auch direkt aus dem AVR – Studio oder auch mit BASCOM und anderen Entwicklungsumgebungen genutzt werden kann.



13

Abb. 4-2: Kommunikation des mySmartUSB Light

¹² Quelle

¹³ Quelle

4.1.2. Technische Spezifikationen

Parameter	Value	Unit
Versorgungsspannung	5	V
Betriebsstrom	10 – 20	mA
Betriebstemperatur	0 – 30	°C

Tabelle 4-1: Technische Spezifikationen von mySmartUSB Light

4.1.3. Programmierumgebung

Um einen kleinen Vorteil durch den Codewizard zu erhalten haben wir als Programmierumgebung das sogenannte Programm „CodevisionAVR“ gewählt.

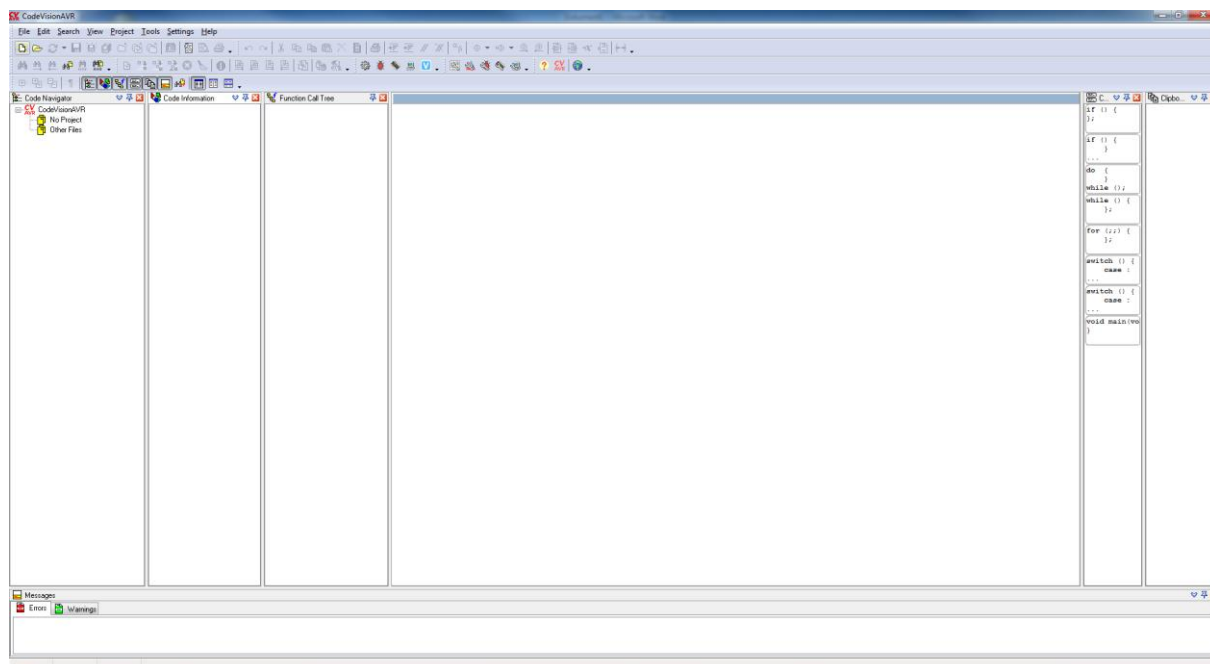


Abb. 4-3: Codevision Entwicklungsumgebung

Im Codewizard kann man die Ports des Mikrocontrollers leichter konfigurieren und LCDs und andere Peripherieteile besser initialisieren.

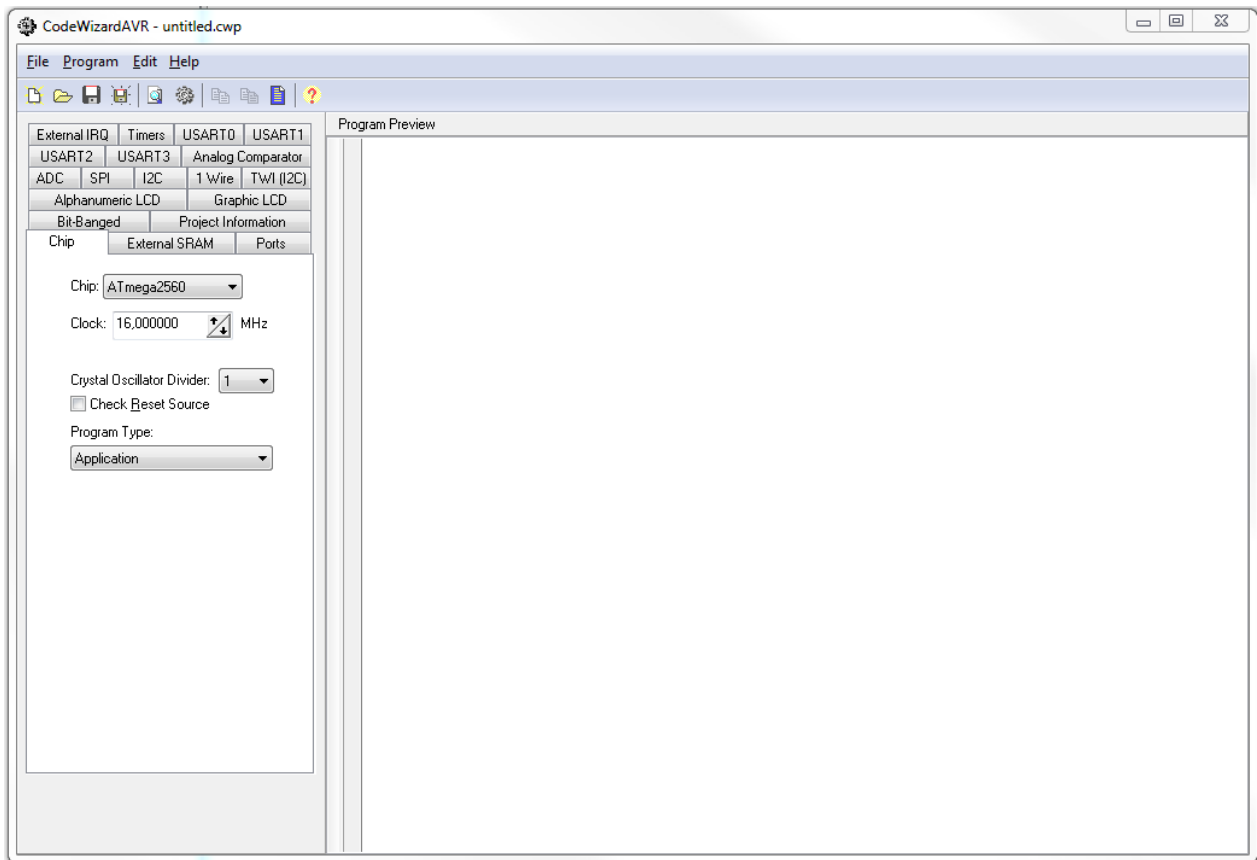


Abb. 4-4: Codewizard

Da man vom Codevision keine Verbindung mit dem ATmega2560 herstellen kann, mussten wir noch ein weiteres Programm für die Programmierung hinzufügen und von diesem das HEX-File an den Mikrocontroller schicken:

Das myAVR ProgTool ist ein freies Werkzeug zum Programmieren von AVR-Mikrocontrollern. Man kann den Programmspeicher, den EEPROM und die Fuse-/Lock-Bits der unterstützten AVR-Mikrocontroller komfortabel programmieren und auslesen.



Abb. 4-5: Einstellung des Programmers

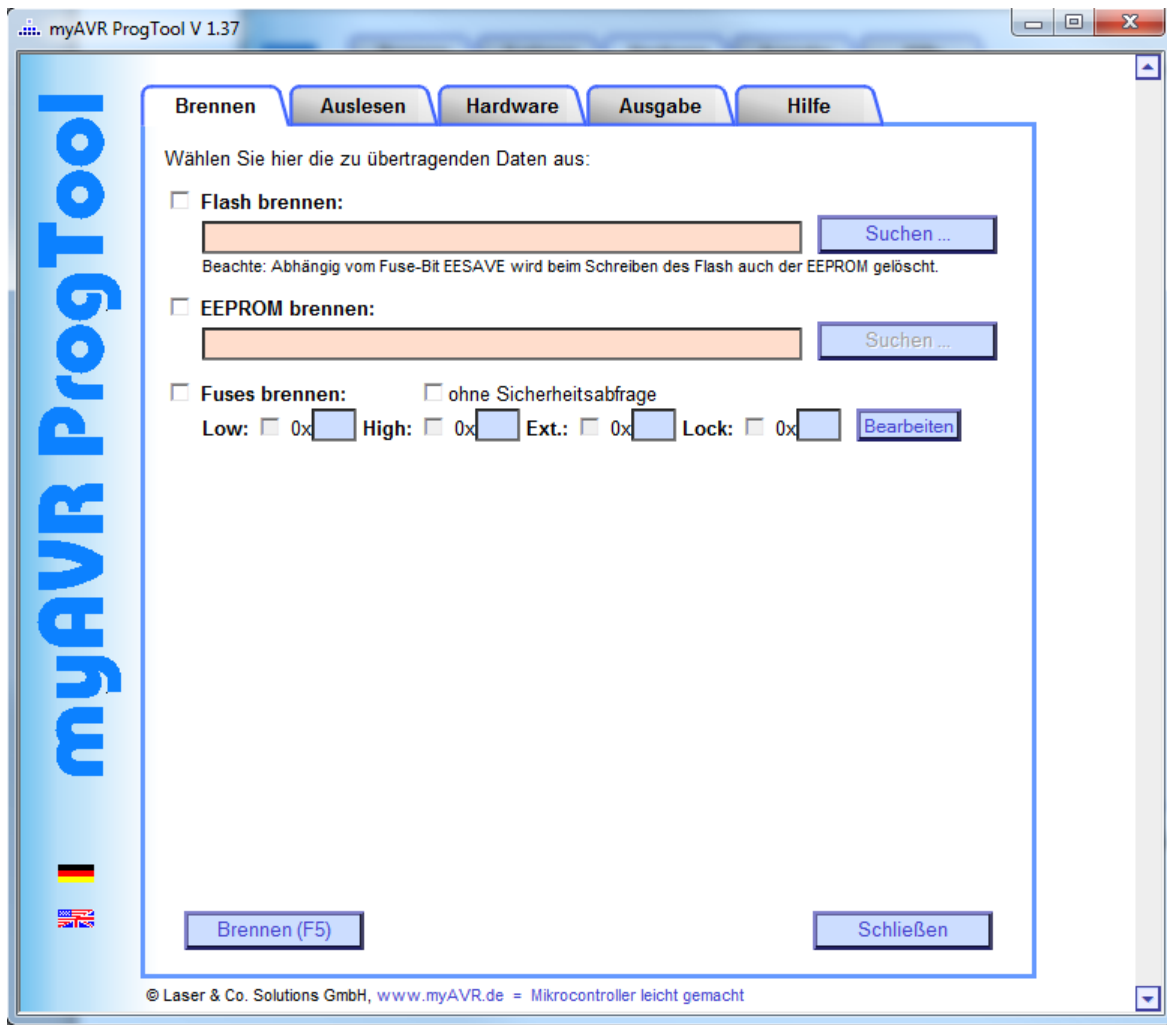


Abb. 4-6: Brennen des Programmes auf den Mikrocontroller

Nach dem Brennen des Programmes auf den Mikrocontroller, wird dieses sofort wiedergegeben und kann durch erneutes Brennen einer HEX-Datei auf den Controller verändert werden.

5. Kostenaufstellung

Menge	Einheit	Artikelbezeichnung	Einzelpr.	Gesamtpr.
1	Stk	LCD 4x40	€21,46	€21,46
1	Stk	Opto-isolator (6N137)	€1,68	€1,68
1	Stk	Arduino Mega2560	€45,67	€45,67
16	Stk	Schalter mit LED (MEC 3FTL680)	€3,62	€57,92
1	Ve	DIN Buchse 5-Pol	€4,19	€4,19
16	Stk	Abdeckkappe (mit Fenster MEC 1E031)	€0,71	€11,36
8	Stk	Inkrementalgeber	€5,45	€43,60
1	Stk	Platine ätzen (303x104)	€133,89	€133,89
1	Stk	Gleichrichter (BI25C)	€0,81	€0,81
5	Stk	Female Steckerleisten	€0,68	€2,04
5	Stk	Male Steckerleiste	€1,79	€5,37
11	Stk	Widerstand	€0,08	€0,88
2	Stk	Kondensator	€0,09	€0,18
1	Stk	Spannungsregler (LM7805)	€0,69	€0,69
1	Stk	Diode (1N41418)	€0,06	€0,06
22	Stk	SMD-LED	€0,12	€2,42

Tabelle 5-1: Kostenaufstellung

6. Probleme und Verworfenne Ideen

6.1. Design

Die ersten Probleme entstanden bereits in der 4. Klasse. Das von uns gewünschte Design konnte nicht so umgesetzt werden wie wir es wollten. In der 5. Klasse wurde das Design so angepasst dass man das Gerät einfach bedienen kann und der vorhandene Platz trotzdem effizient genutzt wird.

6.2. Schaltplan

Zuerst wurden im Schaltplan alle Bauelemente mit „Wire“ verbunden, dies führte zu einem sehr unübersichtlichen Plan, somit wurden alle Verbindungen aufgehoben und mit Bussen neu gemacht.

7. Abbildungsverzeichnis

<i>Abb. 1-1: Stefan Fötschl</i>	5
<i>Abb. 1-2: Daniel Rutte</i>	6
<i>Abb. 2-1: Blockschaltbild zur Kurzbeschreibung</i>	9
<i>Abb. 3-1: Ansicht der Taster, Drehregler etc.</i>	10
<i>Abb. 3-2: Arduion MEGA 2560</i>	15
<i>Abb. 3-3: Pinbelegung des Arduino-Boards</i>	15
<i>Abb. 3-4: AVR ATmega2560</i>	17
<i>Abb. 3-5: Pinbelegung des AVR ATmega2560</i>	18
<i>Abb. 3-6: LM7805</i>	20
<i>Abb. 3-7: Pinbelegung des LM7805</i>	20
<i>Abb. 3-8: Beschaltung des LM7805</i>	21
<i>Abb. 3-9: 6N139</i>	22
<i>Abb. 3-10: Pinbelegung des 6N139</i>	22
<i>Abb. 3-11: AS1100</i>	23
<i>Abb. 3-12: Pinbelegung des AS1100</i>	23
<i>Abb. 3-13: Adressen des AS1100</i>	24
<i>Abb. 3-14: Beschaltung des AS1100</i>	25
<i>Abb. 3-15: LC-Display 4x40</i>	26
<i>Abb. 3-16: Beschaltung des LCDs</i>	28
<i>Abb. 3-17: 74HCT14</i>	29
<i>Abb. 3-18: Pinbelegung des 74HCT14</i>	29
<i>Abb. 3-19: Beschaltung des MIDI -In/ -Out/ -Thru</i>	31
<i>Abb. 3-20: Symbol des Arduino MEGA 2560</i>	32
<i>Abb. 3-21: Package des Arduino MEGA 2560</i>	33
<i>Abb. 3-22: des Arduino MEGA 2560</i>	33
<i>Abb. 3-23: Symbol des AS1100</i>	34
<i>Abb. 3-24: Package des AS1100</i>	34
<i>Abb. 3-25: Device des AS1100</i>	34
<i>Abb. 3-26: Symbol des LC- Displays</i>	35
<i>Abb. 3-27: Package des LC- Displays</i>	35
<i>Abb. 3-28: Device des LC- Displays</i>	35
<i>Abb. 3-29: Symbol der LED</i>	36
<i>Abb. 3-30: Symbol des Tasters</i>	36
<i>Abb. 3-31: Package des Tasters ohne LED</i>	36
<i>Abb. 3-32: Package des Tasters mit LED</i>	36
<i>Abb. 3-33: Device des Tasters ohne LED</i>	37
<i>Abb. 3-34: Device des Tasters mit LED</i>	37
<i>Abb. 3-35: Symbol des Inkrementalgebers</i>	38

<i>Abb. 3-36: Symbol des Switches</i>	38
<i>Abb. 3-37: Package des Inkrementalgebers</i>	38
<i>Abb. 3-38: Device des Inkrementalgebers</i>	38
<i>Abb. 3-39: Gesamtschaltung</i>	39
<i>Abb. 3-40: Schaltung des Spannungsreglers</i>	40
<i>Abb. 3-41: Schaltung des LC- Displays</i>	40
<i>Abb. 3-42: Schaltung der LED- Matrix</i>	41
<i>Abb. 3-43: Schaltung des AS1100</i>	41
<i>Abb. 3-44: Schaltung der MIDI -In/ -Out/ -Thru</i>	42
<i>Abb. 3-45: Top- und Bottom- Layer</i>	43
<i>Abb. 3-46: Top- Layer</i>	44
<i>Abb. 3-47: Bottom- Layer</i>	45
<i>Abb. 3-48: Bestückungsplan Top</i>	46
<i>Abb. 3-49: Bestückungsplan Bottom</i>	47
<i>Abb. 3-50: Aufbau</i>	48
<i>Abb. 4-1: mySmartUSB Light</i>	49
<i>Abb. 4-2: Kommunikation des mySmartUSB Light</i>	49
<i>Abb. 4-3: Codevision Entwicklungsumgebung</i>	50
<i>Abb. 4-4: Codewizard</i>	51
<i>Abb. 4-5: Einstellung des Programmers</i>	52
<i>Abb. 4-6: Brennen des Programmes auf den Mikrocontroller</i>	53

8. Tabellenverzeichnis

<i>Tabelle 3-1: Technische Spezifikationen des Arduino Boards</i>	16
<i>Tabelle 3-2: Eigenschaften des AVR ATmega2560</i>	17
<i>Tabelle 3-3: Pinbelegung des Arduinos bzw. AVR</i>	19
<i>Tabelle 3-6: Technische Spezifikationen des AS1100</i>	24
<i>Tabelle 3-7: Technische Spezifikationen des LC- Displays</i>	26
<i>Tabelle 3-8: Pinbelegung des LC- Displays</i>	27
<i>Tabelle 4-1: Technische Spezifikationen von mySmartUSB Light</i>	50
<i>Tabelle 5-1: Kostenaufstellung</i>	54

9. Quellenangabe

QuelleF1: Online im Internet

<http://docs-europe.electrocomponents.com/webdocs/0e8b/0900766b80e8ba22.pdf>

QuelleF2: Online im Internet:

<http://shop.myavr.de/index.php?sp=article.sp.php&artID=200006>

QuelleF3: Online im Internet:

<http://shop.myavr.de/index.php?sp=article.sp.php&artID=200006>

QuelleR1: Bild des ATMEGA2560, Online im Internet:

http://home.roboticlab.eu/_media/images/avr/avr_atmega128_smd.jpg

QuelleR2: Datenblatt des ACR2560, Online im Internet:

<http://www.atmel.com/Images/doc2549.pdf>

QuelleR3: Bild des LM7805, Online im Internet

<http://www.learningaboutelectronics.com/images/LM7805.jpg>

QuelleR4: Datenblatt des LM7805, Online im Internet

<http://pdf1.alldatasheet.com/datasheet-pdf/view/82833/FAIRCHILD/LM7805.html>

QuelleR5: Bild des 6N139, Online im Internet:

http://www.ereshop.com/shop/bmz_cache/2/26d321aecabe732f98e805671dda2c3f.image.256x256.jpg

QuelleR6: Datenblatt des 6N139, Online im Internet:

<http://www.pselectronic.cz/pdf/1114/1114033.pdf>

QuelleR7: Datenblatt des AS1100, Online im Internet:

<http://pdf1.alldatasheet.com/datasheet-pdf/view/150033/AMSCO/AS1100.html>

QuelleR8: Datenblatt des AS1100, Online im Internet:

<http://pdf1.alldatasheet.com/datasheet-pdf/view/150033/AMSCO/AS1100.html>

QuelleR9: Bild des 74HCT14, Online im Internet:

<http://www.nuri.kilu.de/bilder/74HCT14/74HCT14.jpg>

QuelleR10: Datenblatt des 74HCT14, Online im Internet:

http://www.datasheetcatalog.org/datasheet/philips/74HC_HCT14_CNV_2.pdf

10. Eidesstattliche Erklärung

Wir erklären an Eides statt, dass wir diese Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht haben.

Graz, am

Stefan Fötschl

Graz, am

Daniel Rutte