

Bedienungsanleitung XpressNet-Stellpult

Stand: 14.11.2021

Version: 1.0

Ersteller: Andreas Schulze

1 Inhalt

1	Inhalt.....	2
2	Änderungsübersicht	3
3	Systemvoraussetzungen.....	4
4	Bauteile.....	5
	4.1 XpressNet-Shield	5
	4.2 LED-Treiber	5
5	Platinenbestückung	6
	5.1 XpressNet-Shield	6
	5.1.1 Platinenkorrektur	6
	5.1.2 Stiftleisten auf der Unterseite	6
	5.1.3 Buchsenleiste für Micro-SD-Karten-Leser	7
	5.1.4 Sonstige Bauteile	7
	5.2 LED-Treiber	8
6	Verkabelung.....	9
	6.1 Verbindung der Platinen	9
	6.2 Anschluss der Taster.....	10
	6.3 Anschluss der LEDs	10
	6.4 Anschluss des Tasters zur Ermittlung der LED-Adressen.....	12
	6.5 Anschluss eines optionalen LCD-Displays.....	12
	6.6 Anschluss der Spannungsversorgung	13
7	Inbetriebnahme	13
8	Anpassung der Konfigurationsdatei	14
	8.1 Übersicht	14
	8.2 Beschreibung der Konfigurationsmöglichkeiten	16
	8.2.1 Kunde.....	16
	8.2.2 xpressnet	16
	8.2.3 zeiten	17
	8.2.4 tastenkonfig.....	17
	8.2.5 ledadressen	17
9	Software	18
	9.1 Arduino-IDE	18
	9.2 Bibliotheken.....	18
	9.3 Erweiterungen	18

2 Änderungsübersicht

Änderungsdatum	Änderung	Verantwortlicher

Ansprechpartner

schulze-modellbau.de

Andreas Schulze

Sackmannstraße 30

30453 Hannover

E-Mail: andreas@schulze-modellbau.de

3 Systemvoraussetzungen

Das hier dargestellte Arduino-XpressNet-Stellpult wird als Bausatz angeboten. Die Bestückung der Platinen erfolgt in Eigenregie.

Für den Aufbau sind Lötkenntnisse erforderlich. Desweiteren muss der Umgang mit der Arduino Entwicklungsumgebung geläufig sein.

Wie der Name bereits andeutet, erfolgt der Anschluss des Stellpults an den XpressNet-Bus. Ein Anschluss an andere Bus-Systeme wie z.B. LocoNet ist nicht möglich.

Zum jetzigen Zeitpunkt wurde das Stellpult an folgenden Zentralen getestet:

- Roco Multimaus 10764
- Digikeijs DR5000

4 Bauteile

Zur Bestückung der Platinen sind die nachfolgend beschriebenen Bauteile notwendig.

4.1 XpressNet-Shield

Anzahl	Bezeichnung
1	MAX 485 CSA
1	LED grün, SMD 1206
1	LED rot, SMD 1206
1	Widerstand 120, SMD 1206
1	Widerstand 330, SMD 1206
1	Widerstand 1k, SMD 1206
1	Widerstand 10k, SMD 1206
1	Widerstand 100k, SMD 1206
1	Trimpotentiometer 10k, SMD liegend 5mm
3	Kondensator X7R 100nF, SMD 0805
1	Codierschalter 16 Positionen HEX, SMD 3+3
1	Stiftleiste 2,54mm 2x36 gerade
4	Stiftleiste 2,54mm 1x36 gerade
3	Buchsenleiste 2,54mm 1x36 gerade
1	Micro SD-Karte 4GB
1	Modular Einbaubuchse 6/6
1	Westernkabel 2x Stecker 6polig belegt, Länge nach Bedarf
1	USB-Ladegerät mit mindestens 1 Port
1	USB-Kabel A-Stecker auf B-Stecker, Länge nach Bedarf
1	Arduino kompatibles Mega 2560 R3 Board
1	Breakout Board für Micro-SD-Karte

Über den Link <https://www.reichelt.de/my/1840189> kann eine Bauteilliste für den Lieferanten Reichelt abgerufen werden.

4.2 LED-Treiber

Anzahl	Bezeichnung
1	IRLML 5203, MOSFET 30V 3A
1	MAX 7219 CNG, Displaytreiber Konstantstromregler
1	Widerstand 10k, SMD 1206
1	Widerstand 100k, SMD 1206
3	Kondensator X7R 100nF, SMD 0805
1	IC-Sockel 24polig schmal
8	Stiftleiste 2,54mm 2x8 gerade
8	Buchsenleiste 2,54mm 2x8 gerade
2	Stiftleiste 2,54mm 1x5 gerade
2	Buchsenleiste 2,54mm 1x5 gerade

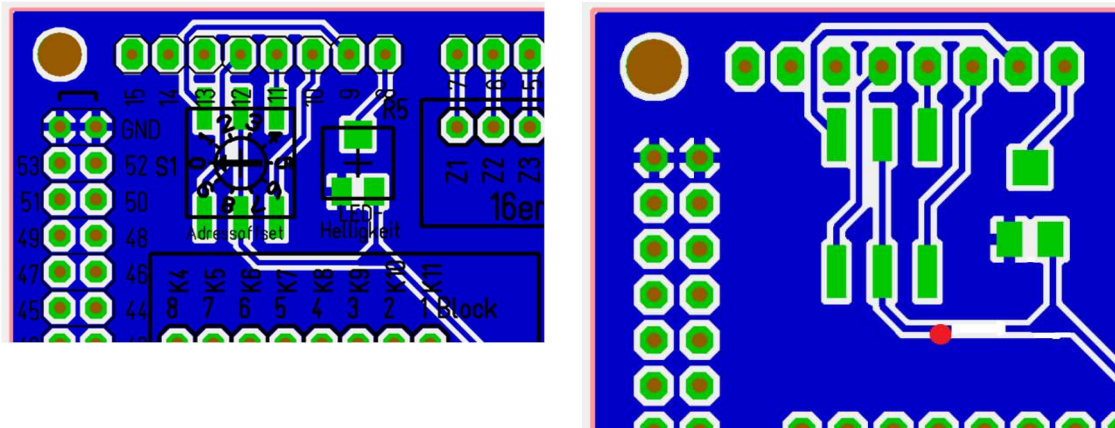
Über den Link <https://www.reichelt.de/my/1840199> kann eine Bauteilliste für den Lieferanten Reichelt abgerufen werden.

5 Platinenbestückung

5.1 XpressNet-Shield

5.1.1 Platinenkorrektur

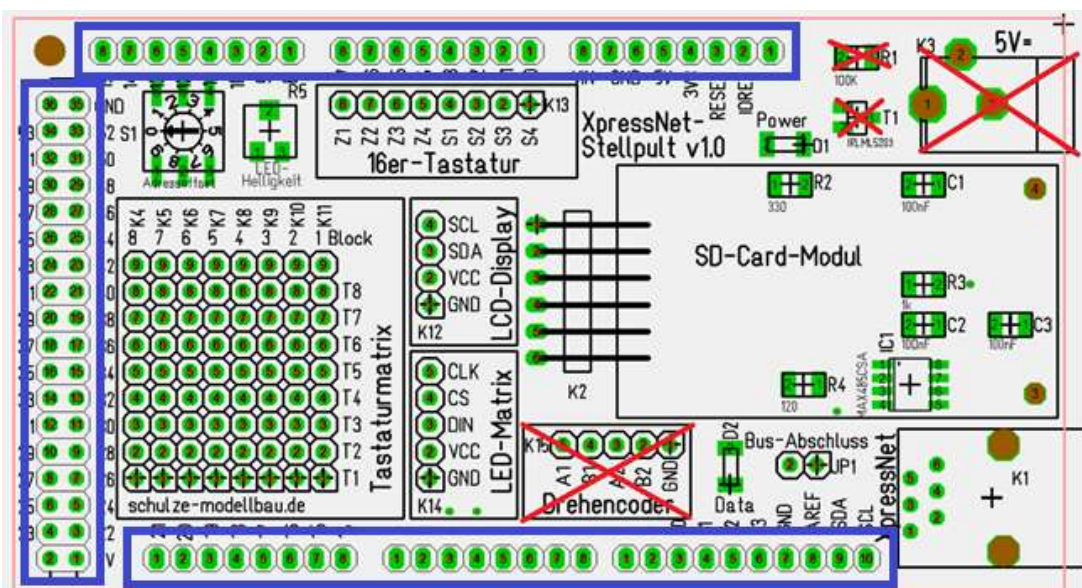
Auf der Platine muss zunächst eine kleine Korrektur erfolgen, um den Adresswahlschalter funktionsfähig zu bekommen.



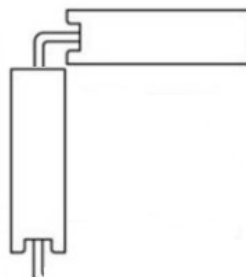
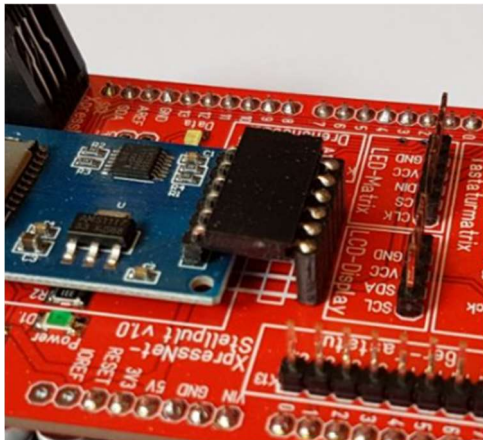
Die Leiterbahn muss unterbrochen werden und stattdessen die vom Adresswahlschalter kommende Leiterbahn mit der Massefläche verbunden werden. Hierzu muss der Lötstopplack an der betreffenden Stelle entfernt werden.

5.1.2 Stiftleisten auf der Unterseite

Die Bestückung des XpressNet-Shields besteht im Wesentlichen aus Stiftleisten. Hierbei ist darauf zu achten, dass die Stiftleisten im Randbereich auf der Unterseite der Platine bestückt werden müssen, damit die Platine auf das Arduino-Board gesteckt werden kann (in der folgenden Abbildung blau markiert).

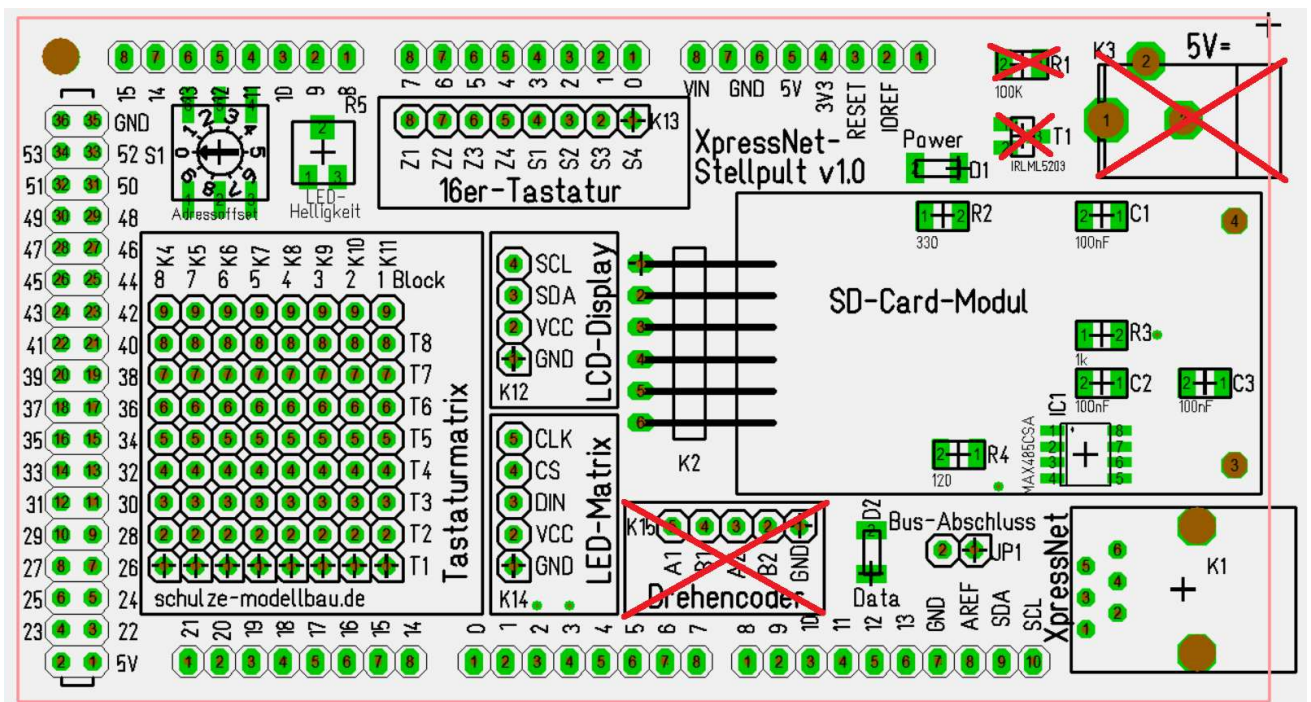


5.1.3 Buchsenleiste für Micro-SD-Karten-Leser



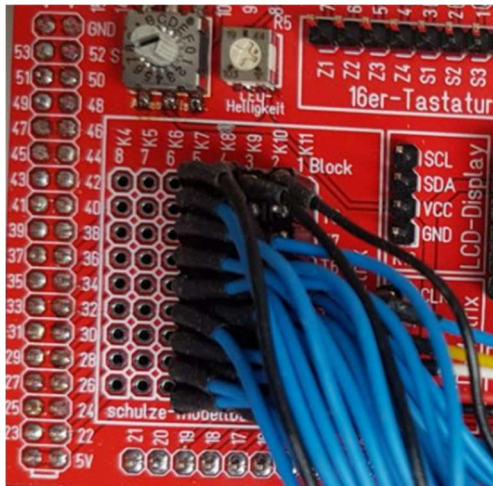
Die Winkelsteckverbindung K2 wird aus einer Buchsenleiste und einer Winkelbuchsenleiste zusammengelötet. Alternativ kann auch eine Buchsenleiste mit langen Anschlüssen verwendet werden.

5.1.4 Sonstige Bauteile



Die Teile K3, R1 und T1 der optionalen Spannungsversorgung brauchen nicht bestückt werden, wenn die Spannungsversorgung über das Arduino-Board läuft.

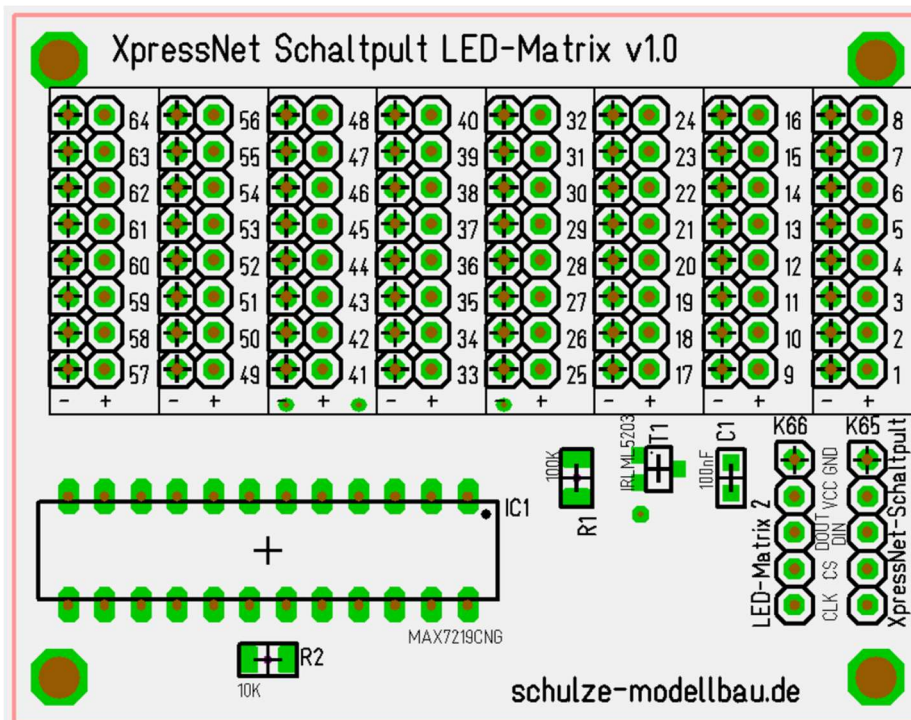
Ebenso wird die Stiftleiste K15 für den Drehencoder nicht benötigt.



Je nach Menge der anzuschließenden Taster können auch Stiftleisten der Tastaturmatrix unbestückt bleiben.

5.2 LED-Treiber

Auch bei der LED-Treiberplatine sind überwiegend Stiftleisten zu bestücken. Je nach Menge der anzuschließenden LEDs können auch Stiftleisten der LED-Matrix unbestückt bleiben.

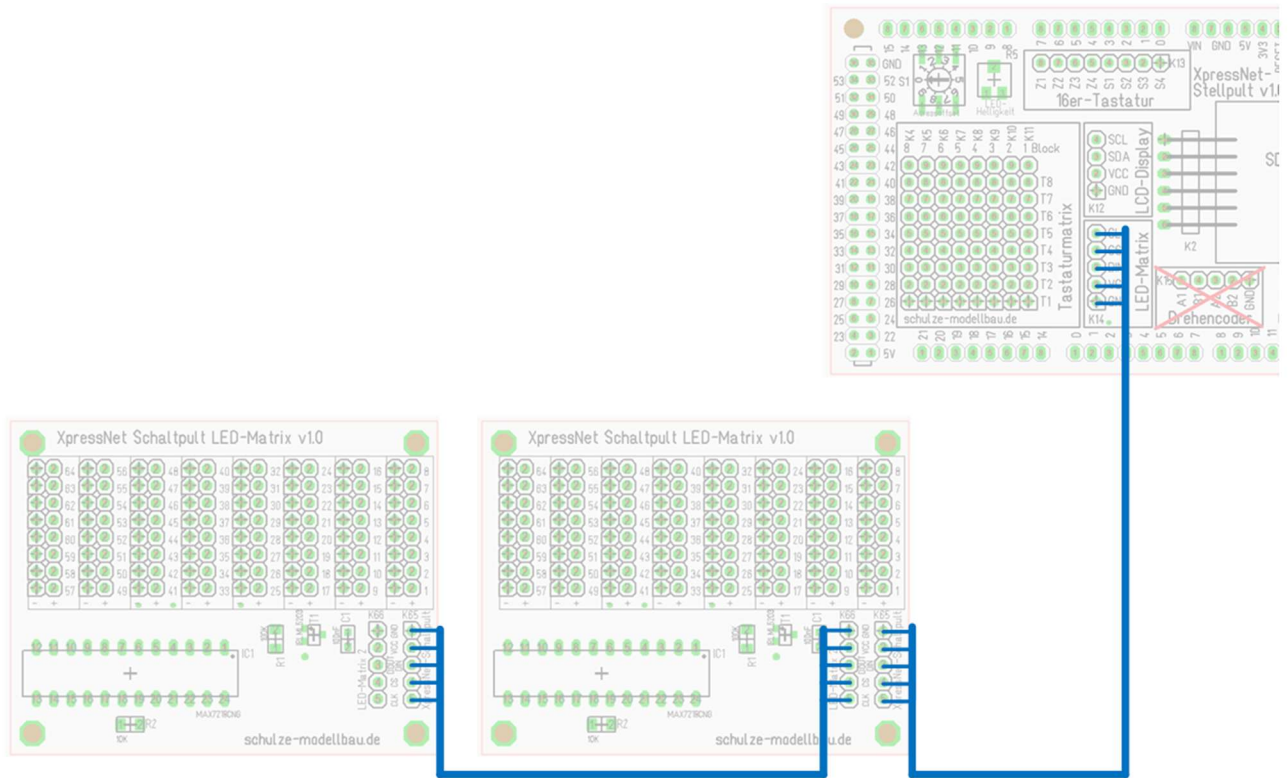


Beim Einsetzen des IC unbedingt auf die richtige Ausrichtung achten (hier im Bild Kerbe nach rechts).

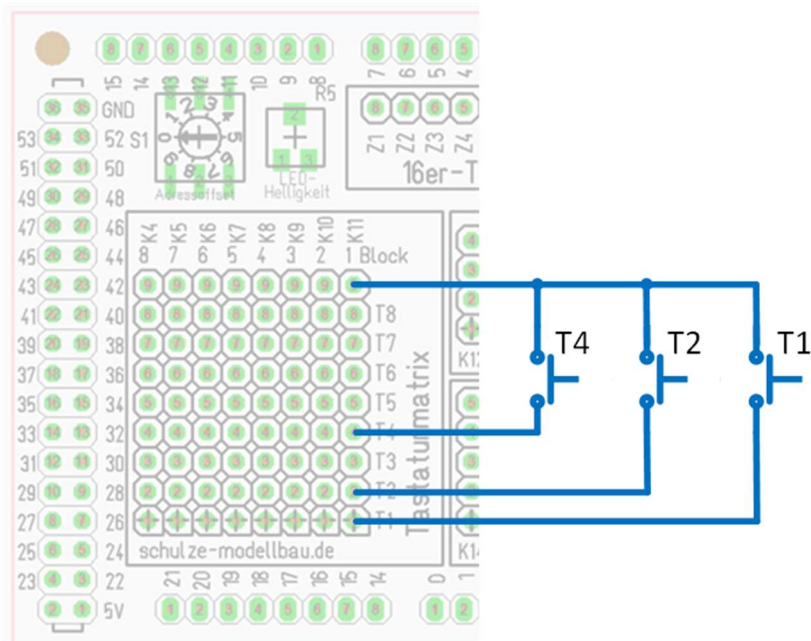
6 Verkabelung

6.1 Verbindung der Platinen

Die XpressNet-Shield-Platine und die LED-Treiberplatine werden über eine 5polige Verbindung verbunden. Hierbei sind die Ein- und Ausgänge der LED-Treiberplatine zu beachten.



6.2 Anschluss der Taster

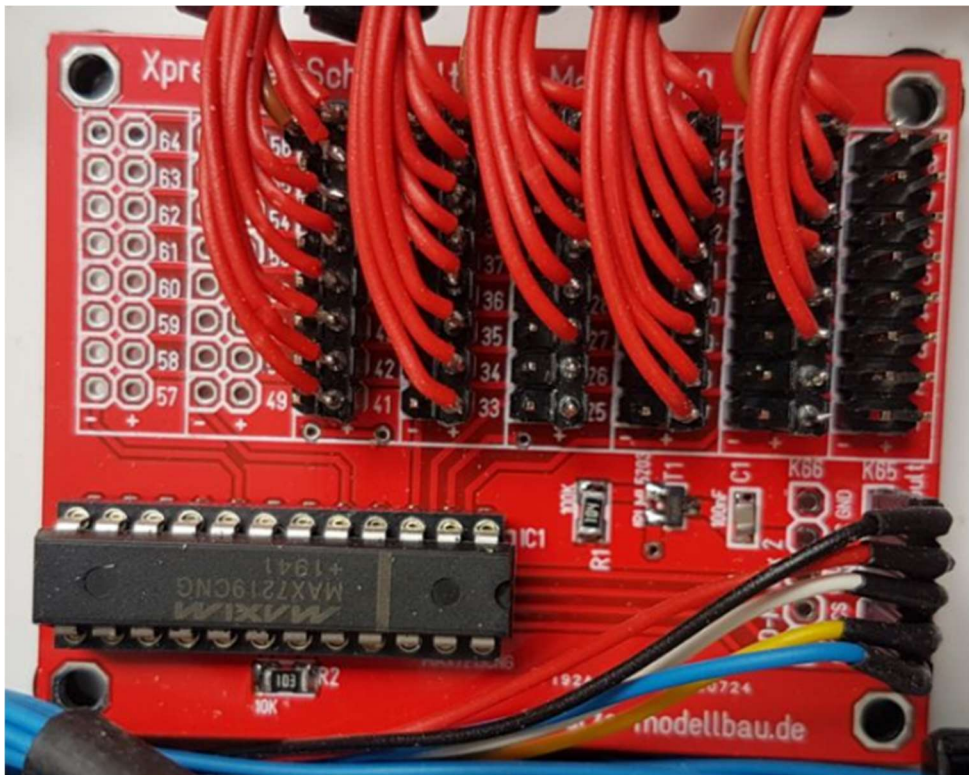
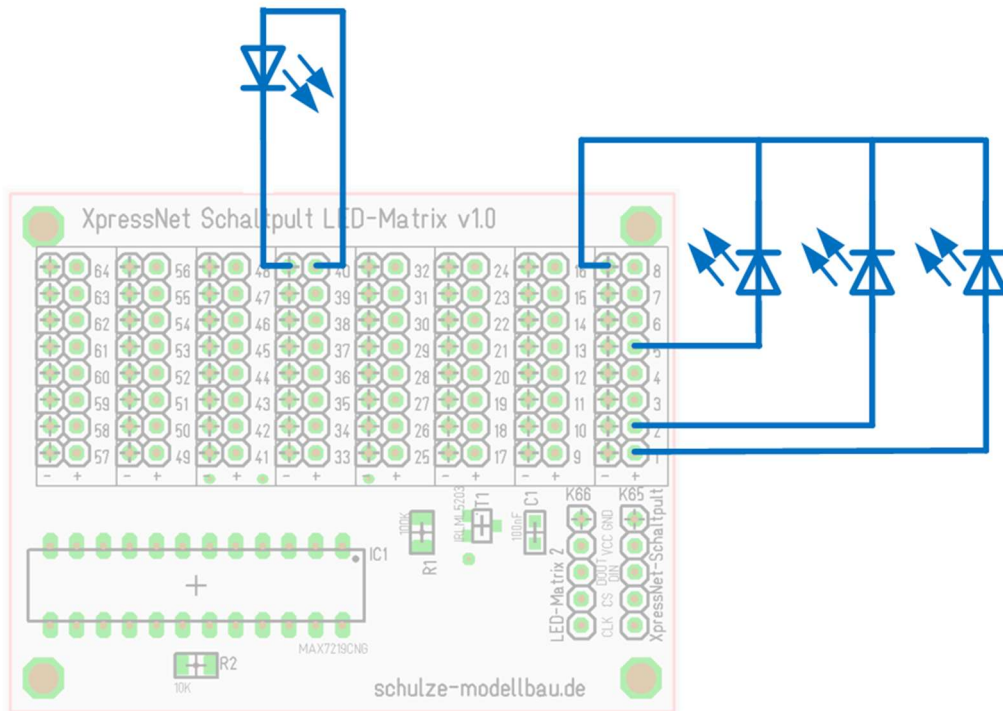


Die Taster des ersten Blocks (wie oben abgebildet) haben die Nummern 1-8. Der zweite Block umfasst die Tastennummern 9-16 usw.

Die Taster können in beliebiger Folge angeschlossen werden. Die Zuordnung der Tasten zu den Weichenadressen erfolgt über die Konfigurationsdatei auf der SD-Karte.

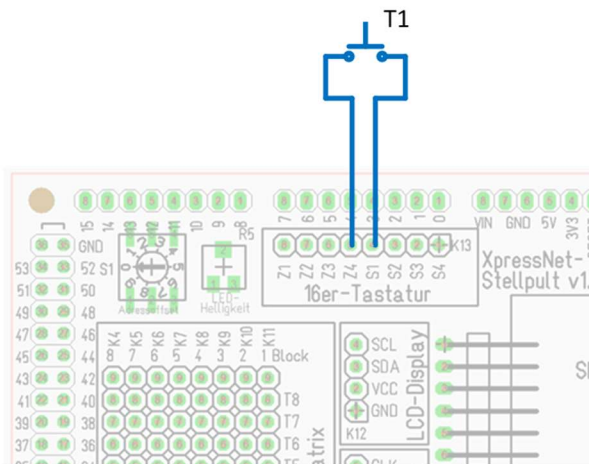
6.3 Anschluss der LEDs

Die LEDs werden ohne Vorwiderstand angeschlossen, da die Regelung hier vom LED-Treiber übernommen wird. Die LEDs können entweder einzeln an den jeweiligen Plus- und Minus-Anschluss der Platine angeschlossen werden oder man fasst den Minus-Anschluss für einen 8er-Block LEDs zusammen.



6.4 Anschluss des Tasters zur Ermittlung der LED-Adressen

Mit Hilfe eines Tasters können die einzelnen LED-Adressen von 1 beginnend durchgetastet werden. Jeder Tastendruck zählt die LED-Adresse um eins hoch und schaltet die zugehörige LED ein. Über ein angeschlossenes LCD-Display kann die aktuelle LED-Adresse abgelesen werden. Dies vereinfacht die Zuordnung der LEDs zu den Weichenadressen erheblich.

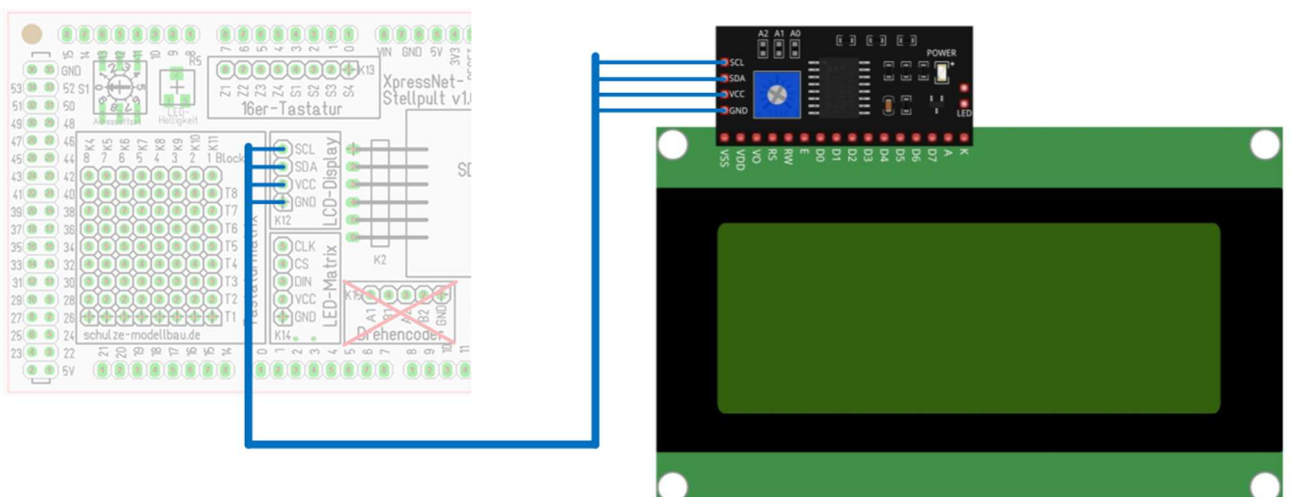


Da der Taster nur zeitweise benötigt wird, kann man eine 2polige Buchsenleiste direkt an den Taster löten.

6.5 Anschluss eines optionalen LCD-Displays

Gerade bei der Inbetriebnahme und der Ermittlung der Taster- und LED-Adressen ist ein Display eine sinnvolle und wertvolle Hilfe. Beim Starten des Systems werden auf dem Display wichtige Informationen ausgegeben, die Hinweise auf ggf. vorhandene Probleme geben können.

Die Software ist auf ein 4x20-LCD-Display ausgelegt, das am I²C-Anschluss angeschlossen wird.



6.6 Anschluss der Spannungsversorgung

Die Spannung kann sowohl über einen 2.4mm Klinkenstecker in das Shield eingespeist werden oder über den USB-Anschluss des Arduino-Mega-Boards. Der Einfachheit halber empfehle ich den USB-Anschluss des Arduino-Boards. Vom Anschluss am 2.4mm Klinkenstecker des Arduino-Boards rate ich dringendst ab, da dies den Spannungsregler auf dem Arduino-Board überlastet.

7 Inbetriebnahme

- Über die Arduino-IDE die Software auf das Arduino-Board spielen.
- Das Shield auf das Arduino-Board stecken. Spannung über das USB-Netzteil und die USB-B-Buchse des Arduino-Boards einspeisen. Die grüne Power-LED auf dem Shield muss leuchten.
- Anschluss eines LCD-Displays und Einstecken der Micro-SD-Karte mit der Konfigurationsdatei. Spannung über das USB-Netzteil und die USB-B-Buchse des Arduino-Boards einspeisen. Im Display müssen die einzelnen Startschritte eingeblendet werden.
- Verbinden der LED-Treiberplatine(n) mit dem Shield. Spannung über das USB-Netzteil und die USB-B-Buchse des Arduino-Boards einspeisen. Im Display müssen die einzelnen Startschritte eingeblendet werden. Beim Punkt „LED-Test“ müssen die an die LED-Treiberplatine angeschlossenen LEDs aufleuchten.

8 Anpassung der Konfigurationsdatei

Bei der Bearbeitung der Datei muss berücksichtigt werden, dass eine Zeile maximal 120 Zeichen lang sein darf. Dies sollte man insbesondere bei Kommentaren beachten.

Eine detaillierte Beschreibung der einzelnen Konfigurationsmöglichkeiten erfolgt im Anschluss.

8.1 Übersicht

```
# Konfigurationsdatei für Arduino-XpressNet-Stellpult
# 26.08.2020 / 16.01.2021
#
# Änderungen:
# - 16.01.2021
#   - neuer Befehl G für sich gegenseitig auflösende Schaltungen (z.B. für Signale)
#     eingeführt

# Konfiguration des Kunden
[kunde]
# Name des Kunden (max 20 Zeichen)
name = Andreas Schulze
# Name der Anlage (max 20 Zeichen)
anlage = Demo-Stellpult

# Konfiguration des XpressNet
[xpressnet]
# Basisadresse dieses Moduls am XpressNet
# zu dieser Adresse wird die am Codierschalter eingestellte Adresse addiert
adresse = 10

# Offset für Weichenadresse (ggf. notwendig bei Roco)
offset = -1

# Konfiguration von Weichenschaltzeiten
[zeiten]
# Die Weichenschaltzeit definiert die Dauer des Weichenumlaufs (in ms)
schaltzeit = 500

# die Verzögerung definiert die Zeit zwischen zwei Weichenschaltungen
# beim Initialisieren
verzoegerung = 250

# Konfiguration der Tasten
# Aufbau:
# linke Zahl: Tastennummer (1 bis 64)
# rechts: Befehl mit Parametern wie nachstehend beschrieben
# 1 = U:21                Umschalten Weiche 21
# 1 = U:-21              Umschalten Weiche 21 invertiert
# 1 = S:21A              Schalten Weiche 21 abzweig
# 1 = S:21G              Schalten Weiche 21 gerade
# 1 = K:GO                Kommando Fahrstrom einschalten
# 1 = K:STOP             Kommando Nothalt
# 1 = K:INIT             Kommando alle Weichen in definierte Lage bringen
[tastenkongfig]
# Zeile 1 (Tasten 1-8)
1 =
2 =
3 =
4 = G:13A,13G,14G
5 = G:13G,13A,14G
```

```
6 = U:12
7 = G:14G,13A,13G
8 = U:11

# Zeile 2 (Tasten 9-16)
9 =
10 =
11 =
12 =
13 =
14 =
15 =
16 =

# Zeile 3 (Tasten 17-24)
17 =
18 =
19 =
20 =
21 =
22 =
23 =
24 =

# Zeile 4 (Tasten 25-32)
25 = K:STOP
26 =
27 =
28 =
29 = K:INIT
30 =
31 =
32 = K:GO

# Zeile 5 (Tasten 33-40)
33 =
34 =
35 =
36 =
37 =
38 =
39 =
40 =

# Zeile 6 (Tasten 41-48)
41 =
42 =
43 =
44 =
45 =
46 =
47 =
48 =

# Zeile 7 (Tasten 49-56)
49 =
50 =
51 =
52 =
53 =
54 =
55 =
56 =

# Zeile 8 (Tasten 57-64)
57 =
58 =
59 =
60 =
61 =
```

```
62 =
63 =
64 =

# Konfiguration der LED-Adressen
# pro Weiche werden in der Regel zwei LED-Adressen benötigt.
# linke Zahl: Weichenadresse mit Suffix (A)bzweig oder (G)erade
# rechte Zahl: LED-Adresse
[ledadressen]
    11G = 5
    11A = 6
    12G = 7
    12A = 8
    13G = 1
    13A = 2
    14G = 3

# Ende der Konfiguration
```

8.2 Beschreibung der Konfigurationsmöglichkeiten

8.2.1 Kunde

In diesem Abschnitt werden Zeichenketten eingestellt, die beim Starten der Software auf dem Display erscheinen. Dies ist bei Verwendung mehrerer Stellpulte sinnvoll.

8.2.1.1 name

Beliebiger, max. 20 Zeichen langer Text. Erscheint beim Starten der Software in der 3. Zeile des Displays.

8.2.1.2 anlage

Beliebiger, max. 20 Zeichen langer Text. Erscheint beim Starten der Software in der 4. Zeile des Displays.

8.2.2 xpressnet

In diesem Abschnitt werden den XpressNet-Bus betreffende Einstellungen vorgenommen.

8.2.2.1 adresse

Basisadresse für den XpressNet-Bus. Zu diesem Wert wird der am Adresswahlschalter eingestellte Wert addiert. Der Adresswahlschalter kann Werte von 1 bis 16 annehmen.

Alle am XpressNet-Bus angeschlossenen Geräte nutzen eine eindeutige Adresse. Der Adressbereich reicht von 1 bis 32. Sollten zwei Geräte die gleiche Adresse nutzen wollen, führt dies zu Fehlfunktionen. In dem Fall den Adresswahlschalter verstellen und das Board neu starten.

8.2.2.2 offset

Der Offset für die Weichenadressen wird zu der Weichenadresse addiert, die in der Konfigurationsdatei angegeben ist.

8.2.3 zeiten

In diesem Abschnitt werden Schaltzeiten definiert.

8.2.3.1 schaltzeit

Die Weichenschaltzeit definiert die Dauer des Weichenumlaufs (in ms).

8.2.3.2 verzoegerung

Die Verzögerung definiert die Zeit zwischen zwei Weichenschaltungen beim Initialisieren (in ms).

8.2.4 tastenkonfig

In diesem Abschnitt werden die Befehle definiert, die bei Druck auf eine bestimmte Taste ausgeführt werden sollen.

Aufbau:

linke Zahl: Tastennummer (1 bis 64)

rechts: Befehl mit Parametern wie nachstehend beschrieben

1 = U:21	Umschalten Weiche 21
1 = U:-21	Umschalten Weiche 21 invertiert
1 = S:21A	Schalten Weiche 21 abzweig
1 = S:21G	Schalten Weiche 21 gerade
1 = K:GO	Kommando Fahrstrom einschalten
1 = K:STOP	Kommando Nothalt
1 = K:INIT	Kommando alle Weichen in definierte Lage bringen

8.2.5 ledadressen

In diesem Abschnitt werden die LED-Adressen definiert, die beim Schalten der angegebenen Adresse aktiviert werden soll.

Aufbau:

linke Zahl: geschaltete Adresse

rechte Zahl: LED-Adresse

11G = 5 bei Schalten der Adresse 11 gerade wird die LED-Adresse 5 aktiviert.

9 Software

Die zur Verfügung gestellte Software umfasst neben der von mir erstellten Software auch die mitgenutzten, benötigten Bibliotheken. In einigen Bibliotheken wurden Änderungen vorgenommen, um die gewünschte Funktionalität zu erzielen.

9.1 Arduino-IDE

Die Softwareentwicklung hat mit der Arduino-IDE in der Version 1.8.12 stattgefunden. Andere Versionen können unter Umständen zu Fehlern beim Compilieren führen.

9.2 Bibliotheken

Um die gewünschte Funktionalität zu erreichen, nutze ich einige Bibliotheken, die im folgenden beschrieben sind.

Name	Beschreibung
LiquidCrystal_I2C	wird für die Ansteuerung des I ² C-LCD-Displays benötigt
KeypadInt	Diese Bibliothek basiert auf der Keypad-Bibliothek, die aber nur maximal 26 Tasten (Buchstaben A-Z) verarbeiten kann. Die vorgenommenen Änderungen erweitern die verarbeitbaren Tasten auf den Datentyp INT
IniFile	wird für das Auslesen der Konfigurationsdatei von der Micro-CD-Karte benötigt.
XpressNet	wird für die Kommunikation auf dem XpressNet-Bus benötigt. Diese Bibliothek wurde erweitert, um den Serial_Port_2 nutzen zu können
LedControl	wird für die Ansteuerung des LED-Multiplexers benötigt

9.3 Erweiterungen