



Tom simple Handwheel

V3

Mit Mesa SmartSerial Schnittstelle
realisiert durch Talla83

- * Danke an Tom er hat dieses einfache Handbedienteil entwickelt / konstruiert und gedruckt.
- * Ich hab mir Gedanken gemacht wie ich dieses Handrad am besten an die Maschine bekomme.
- * Denn es herrscht wie so oft IO Mangel. Daher habe ich mich für Schnittstelle "sserial" entschieden.
- * Das Beschreibung zum Protokoll findet man zum Beispiel im Manual einer 7i73 oder anderen SmartSerial Cards von Mesa.

- * Links:
- * <https://www.thingiverse.com/thing:3684937>
- * <http://www.talla83.de/linuxcnc/config.htm>

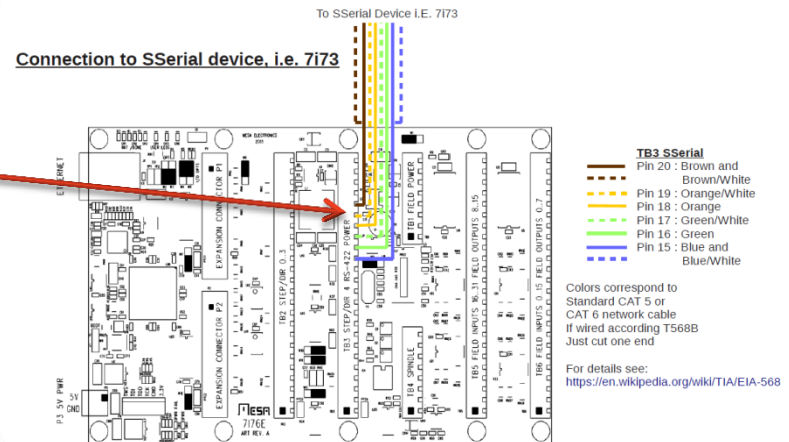
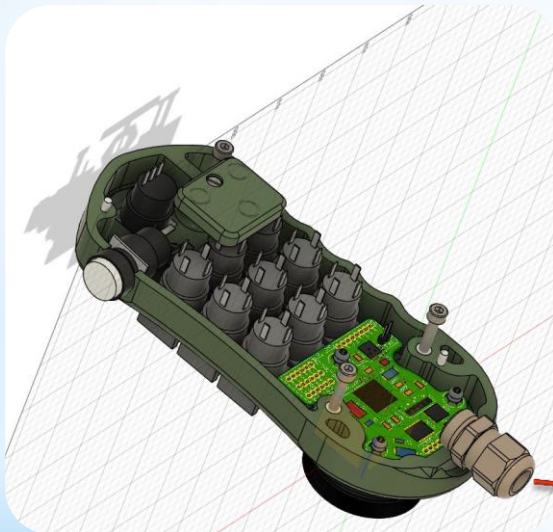
- * <http://linuxcnc.org/docs/html/man/man9/sserial.9.html>
- * <https://github.com/LinuxCNC/linuxcnc/blob/master/src/hal/drivers/mesa-hostmot2/sserial.c>

* Tom simple
Handwheel

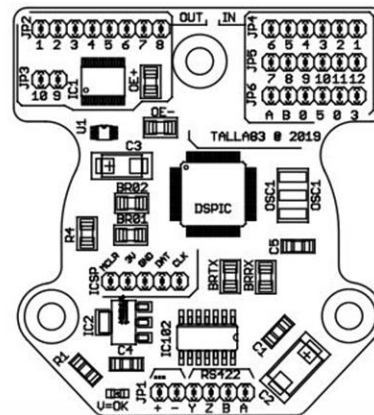
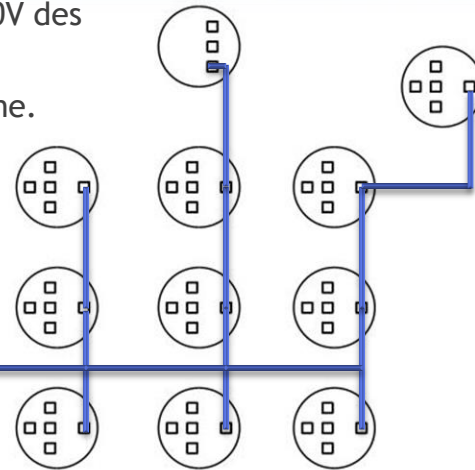
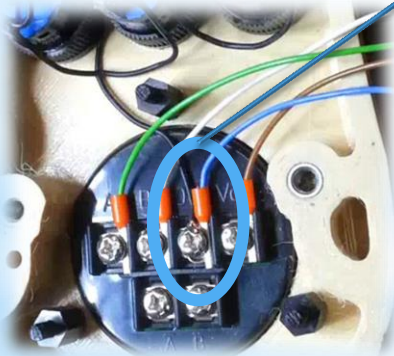
- * Kunststoffgehäuse (3D Druck - <https://www.thingiverse.com/thing:3684937>)
- * Standard China Handrad 60mm (Encoder 5V und A + B Signal)
- * Taster BENKPAK (16mm in Viereckig oder Rund)



- * Zusätzliche Schnittstellenplatine ermöglicht einfache Einbindung in Systeme mit Mesakarten und LinuxCNC.
- * <http://linuxcnc.org/docs/html/man/man9/sserial.9.html>
- * Zum Beispiel bietet eine 7i76e direkt einen SmartSerial Kanal
- * https://youtu.be/Y1_VYIPNtuk (Video : Test an einer Maschine)

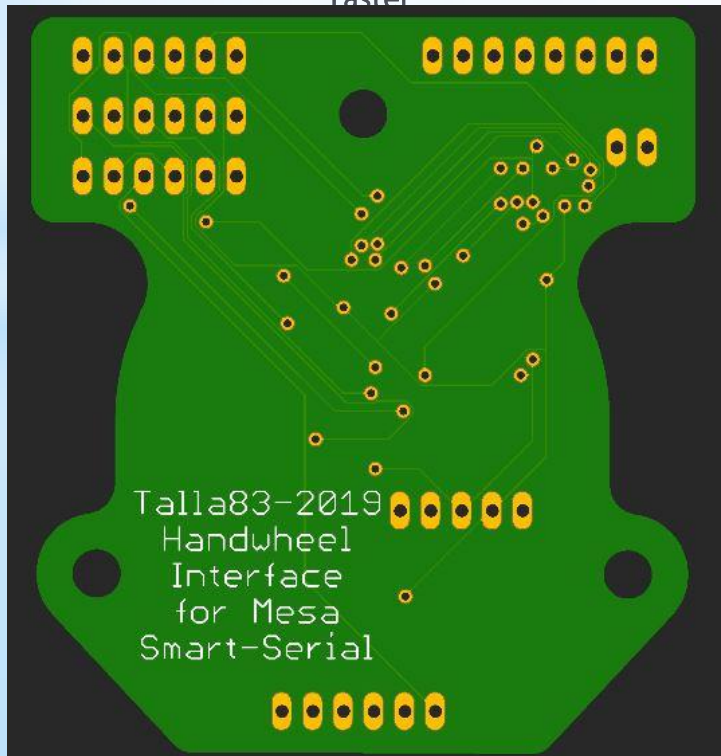


- * Verkabelung Teil 1
- * Alle Taster werden mit GND verbunden.
- * Dazu kann man alle Taster zusammen verbinden und anschließend auf den 0V des Encoder führen.
- * Vom Encoder geht es später zur Platine.

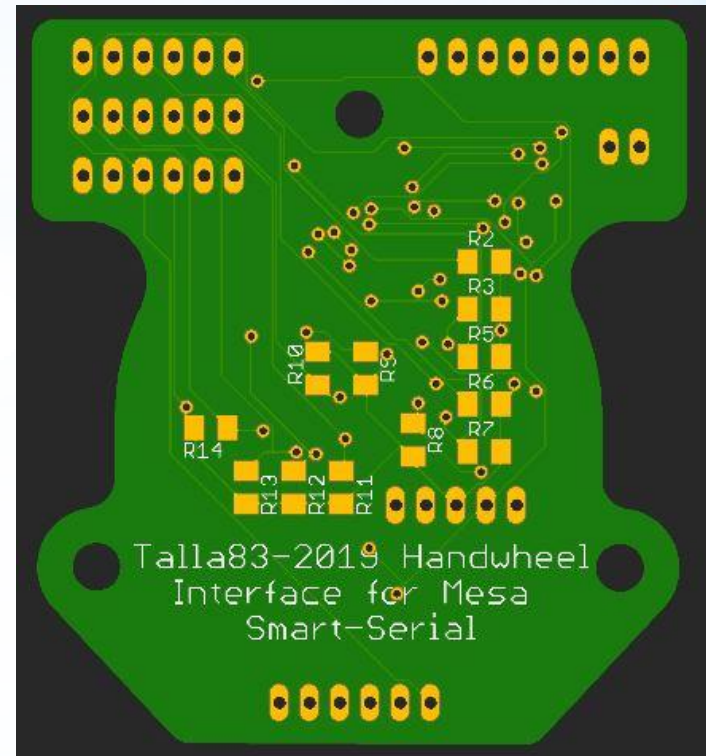




* V2 mit ext.
Widerstand am
Taster



V3 kommt dann ohne
den Widerstand aus

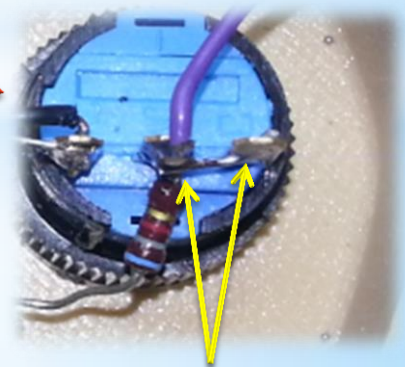
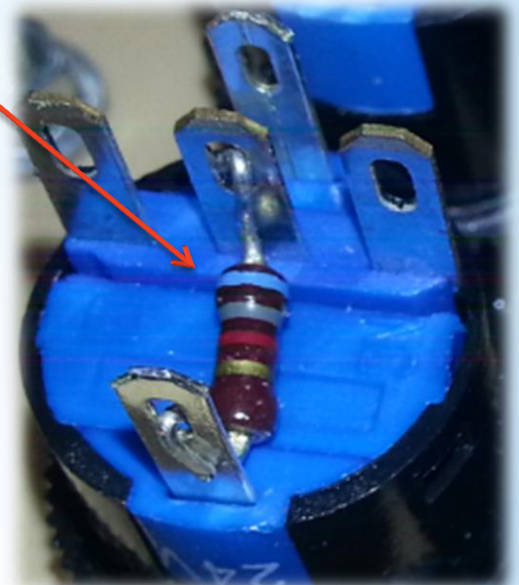


* **Verkabelung Teil 2**

- * Alle Taster bekommen einen Pulldown Widerstand der zwischen 1 und 10 Kiloohm liegen sollte.
- * Diese kann man geschickt direkt am Taster zwischen den Kontakten verbauen.
- * Wenn der Notaus auch als normaler Eingang abgefragt wird, braucht er somit auch einen Pulldown Widerstand.

Ab V3 sind alle Widerstände schon auf der Platine integriert !

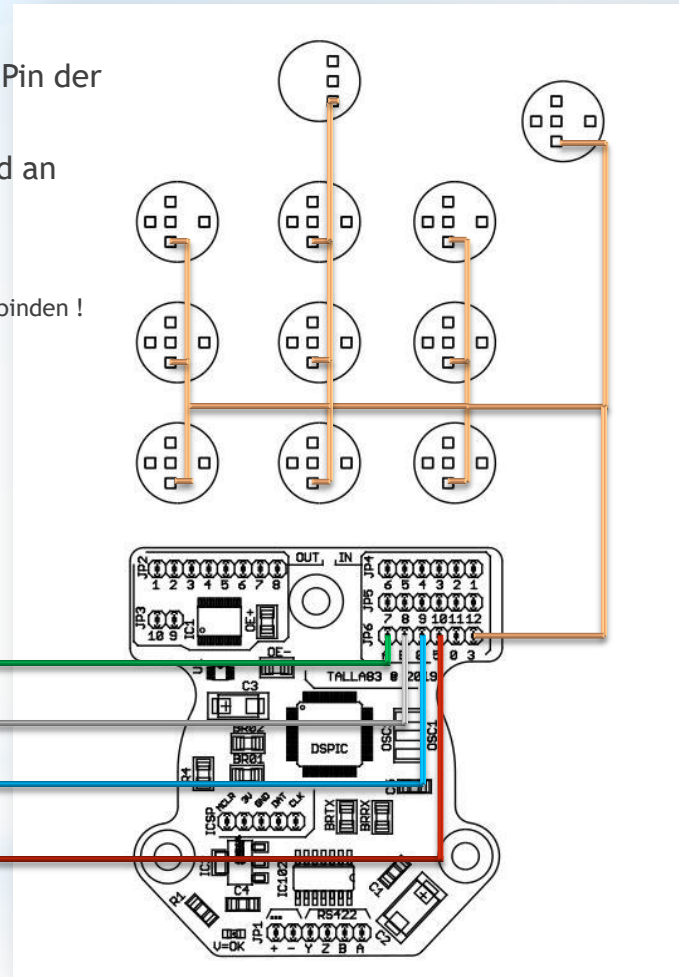
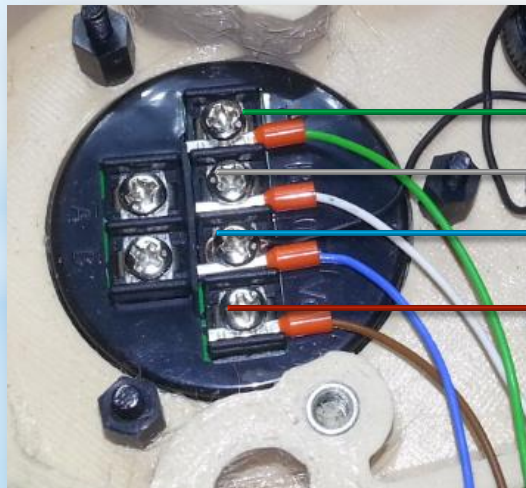
- * Am Encoder müssen nun 4 Leitungen angebracht werden, die dann zur Platine gehen.



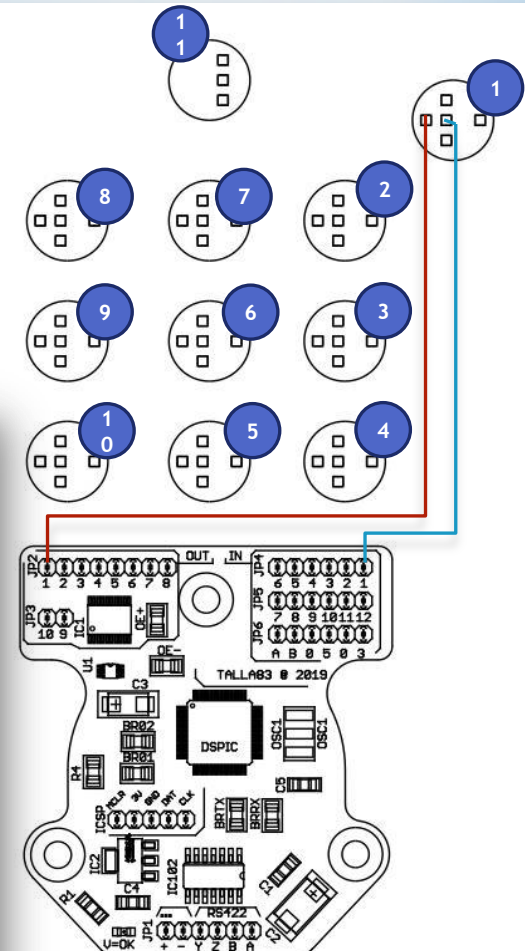
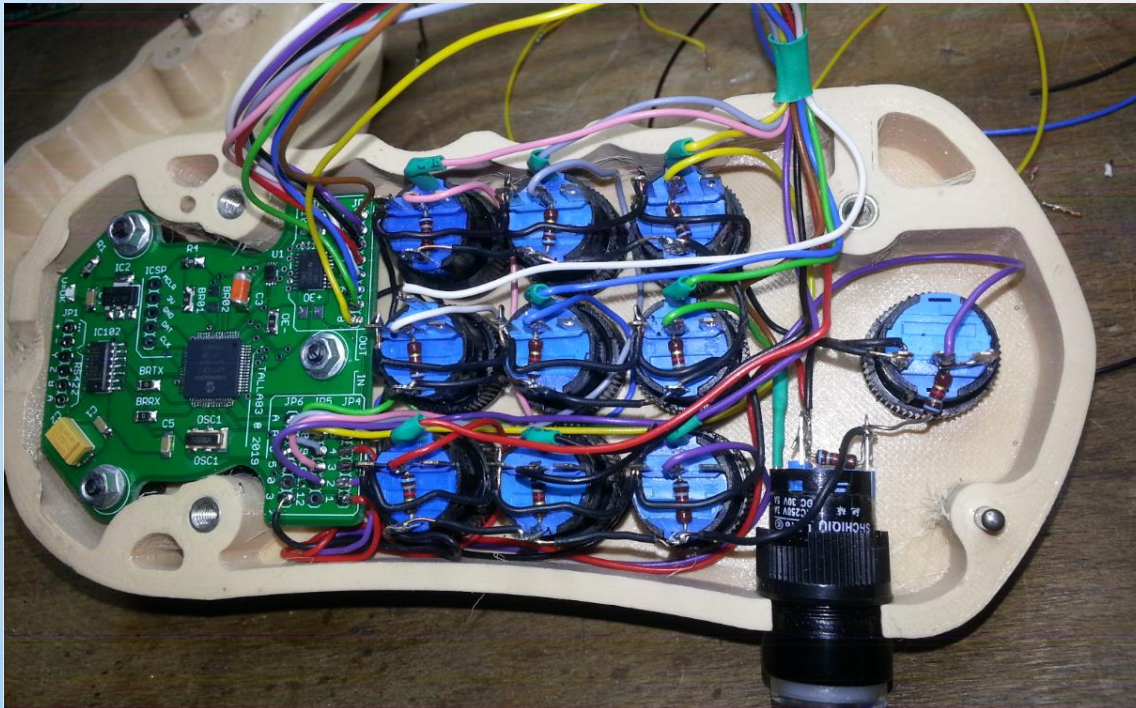
Müssen verbunden werden.

- * Verkabelung Teil 3
- * Alle Taster werden mit 3V verbunden.
- * Dazu kann man alle Taster zusammen verbinden und anschließend auf den 3V Pin der Platine legen.
- * Der Encoder mit seinen 4 Leitungen wird an die Pins A, B, 0 und 5 angeschlossen.

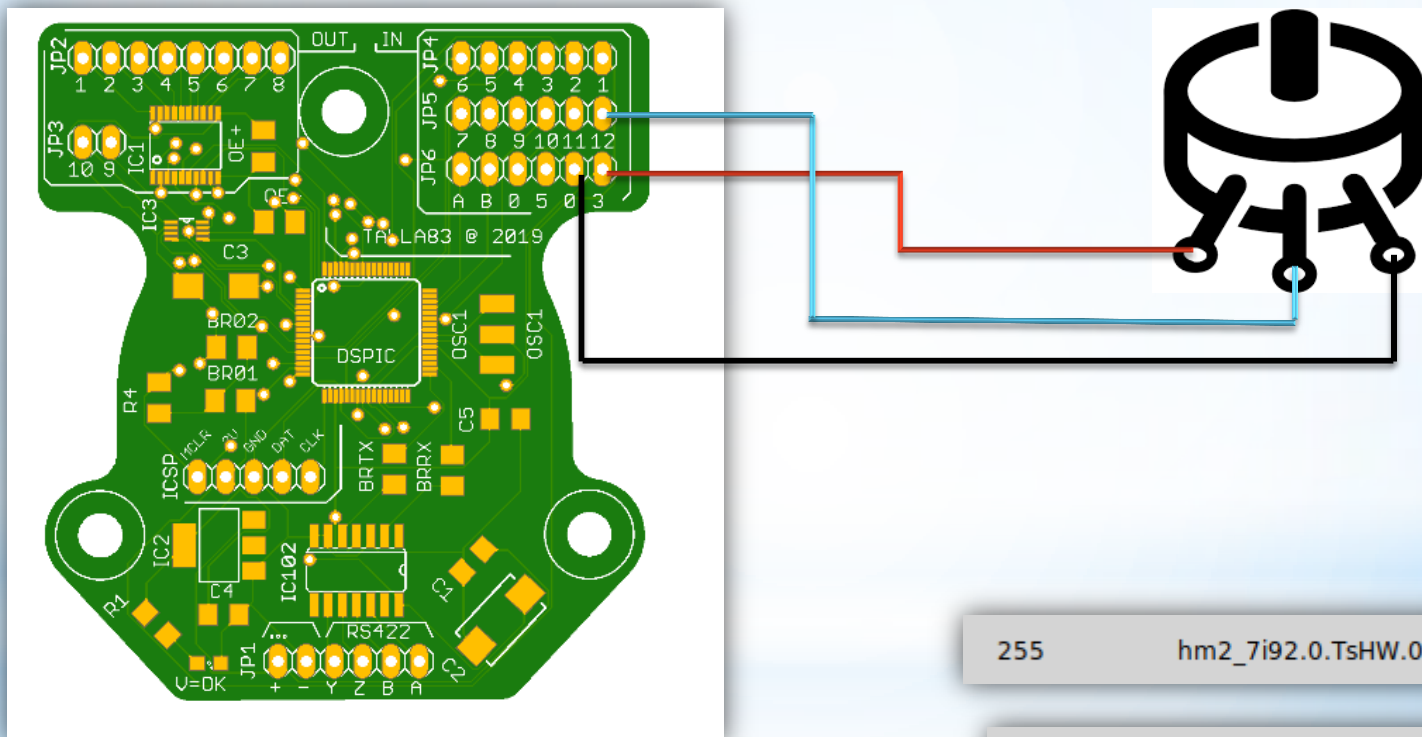
Achtung 5V+ Leitungen **nicht** mit den 3V+ Leitungen verbinden !



- * Verkabelung Teil 4
- * Auf der Platine gibt es links den Bereich „OUT“ und rechts den Bereich „IN“.
- * OUT = LED +
- * IN = Taster
- * Es wird immer 1 zu 1 verbunden.
- * Taster 1 NO Kontakt auf „IN“ 1
- * Taster 1 LED + auf „OUT“ 1
- * Und so weiter.....



- * **Verkabelung Teil 4a**
- * Funktion Analogeingang ist auf Pin 12 verfügbar
- * 0 bis 255 ist dann der Wert im LinuxCNC (Datatyp U32)



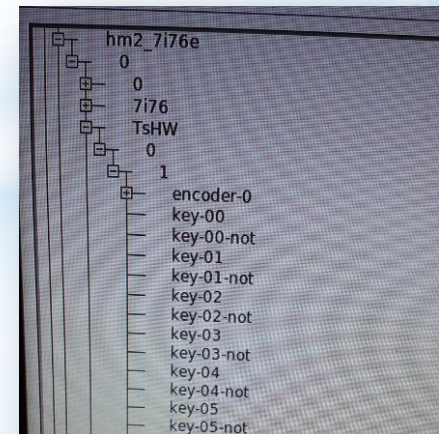
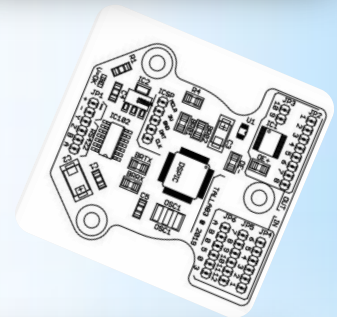
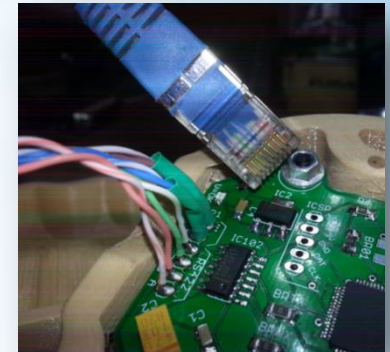
* Verkabelung Teil 5

- * Über die SmartSerial Leitung wird das ganze auch mit 5V versorgt.
- * Mesa verwendet ein RJ45 Kabel, es kann aber auch ein normales Kabel verwendet werden und direkt geklemmt.
- * Folgend ein Beispiel mit einer 7i76e und direkter Klemmung.

CAT5	PINS	SIGNAL DIR	CAT5 568B COLOR
1	RX-	TO TsHW A	ORANGE/WHITE
2	RX+	TO TsHW B	ORANGE
3	TX-	FROM TsHW Y	GREEN/WHITE
4	GND	TO TsHW -	BLUE
5	GND	TO TsHW -	BLUE/WHITE
6	TX+	FROM TsHW Z	GREEN
7	+5V	TO TsHW +	BROWN/WHITE
8	+5V	TO TsHW +	BROWN

TsHW	7i76e (TB3)
-	GND (15)
Z (TX+)	RX+ (16)
Y (TX-)	RX- (17)
B (RX+)	TX+ (18)
A (RX-)	TX- (19)
+	+5VP (20)

- * hm2_7i76e.TsHW.0.1.key-00 / hm2_7i76e.TsHW.0.1.led-00
- * hm2_7i76e.TsHW.0.1.encoder-0



- * Unter folgenden Link ein Beispiel wie man dieses in der HAL verknüpfen kann.
- * http://talla83.homepage.t-online.de/linuxcnc/TsHW_HAL.zip

```
TsHW_2_9 - Editor
Datei Bearbeiten Format Ansicht Hilfe
#####
# TsHW Talla simple Hand Wheel LinuxCNC Version 2.9
#####
# ilowpass und and2
#
# Wenn diese schon in einer anderen HAL geladen bzw. verwendet werden,
# dann muss dies in der Struktur beachtet werden.
# Denn doppeltes laden von Komponenten ist nicht möglich.
# Es müssen die loadrt Befehle auskommentiert werden und
# die loadrt Befehle in den anderen HAL Dateien mit diesen Namen
# ergänzt werden.
# addf kann hier bleiben :-))
#
# Damit der Zero der Achsen geht müssen in der INI die MDI Commands
# angelegt werden
#
# [HALUI]
# MDI_COMMAND = G10 L20 P1 X0
# MDI_COMMAND = G10 L20 P1 Y0
# MDI_COMMAND = G10 L20 P1 Z0
#
#
# Mit setp shw.enc-smooth.gain kann die Beschleunigung eingestellt werden.
# und mit setp shw.enc-smooth.scale die Übersetzung
#
#####
#####
## TsHW ##
#####

loadrt ilowpass names=shw.enc-smooth
addf shw.enc-smooth servo-thread

setp shw.enc-smooth.scale 1
setp shw.enc-smooth.gain 0.025

setp axis.x.jog-vel-mode 1
setp axis.y.jog-vel-mode 1
setp axis.z.jog-vel-mode 1

loadrt and2 names=shw.ena-x,shw.ena-y,shw.ena-z,shw.zero-x,shw.zero-y,shw.zero-z

addf shw.ena-x servo-thread
addf shw.ena-y servo-thread
addf shw.ena-z servo-thread

addf shw.zero-x servo-thread
addf shw.zero-y servo-thread
addf shw.zero-z servo-thread

loadrt mux8 names=shw.step-choice
addf shw.step-choice servo-thread

net key-0001 => shw.step-choice.sel0
net key-001 => shw.step-choice.sel1
net key-01 => shw.step-choice.sel2

setp shw.step-choice.in1 0.001
setp shw.step-choice.in2 0.01
```

```
TsHW_2_6 - Editor
Datei Bearbeiten Format Ansicht Hilfe
#####
# TsHW Talla simple Hand Wheel LinuxCNC Version 2.6
#####
# ilowpass und and2
#
# Wenn diese schon in einer anderen HAL geladen bzw. verwendet werden,
# dann muss dies in der Struktur beachtet werden.
# Denn doppeltes laden von Komponenten ist nicht möglich.
# Es müssen die loadrt Befehle auskommentiert werden und
# die loadrt Befehle in den anderen HAL Dateien mit diesen Namen
# ergänzt werden.
# addf kann hier bleiben :-))
#
# Damit der Zero der Achsen geht müssen in der INI die MDI Commands
# angelegt werden
#
# [HALUI]
# MDI_COMMAND = G10 L20 P1 X0
# MDI_COMMAND = G10 L20 P1 Y0
# MDI_COMMAND = G10 L20 P1 Z0
#
#
# Mit setp shw.enc-smooth.gain kann die Beschleunigung eingestellt werden.
# und mit setp shw.enc-smooth.scale die Übersetzung
#
#####
#####
## TsHW ##
#####

loadrt ilowpass names=shw.enc-smooth
addf shw.enc-smooth servo-thread

setp shw.enc-smooth.scale 1
setp shw.enc-smooth.gain 0.025

setp axis.0.jog-vel-mode 1
setp axis.1.jog-vel-mode 1
setp axis.2.jog-vel-mode 1

loadrt and2 names=shw.ena-x,shw.ena-y,shw.ena-z,shw.zero-x,shw.zero-y,shw.zero-z

addf shw.ena-x servo-thread
addf shw.ena-y servo-thread
addf shw.ena-z servo-thread

addf shw.zero-x servo-thread
addf shw.zero-y servo-thread
addf shw.zero-z servo-thread

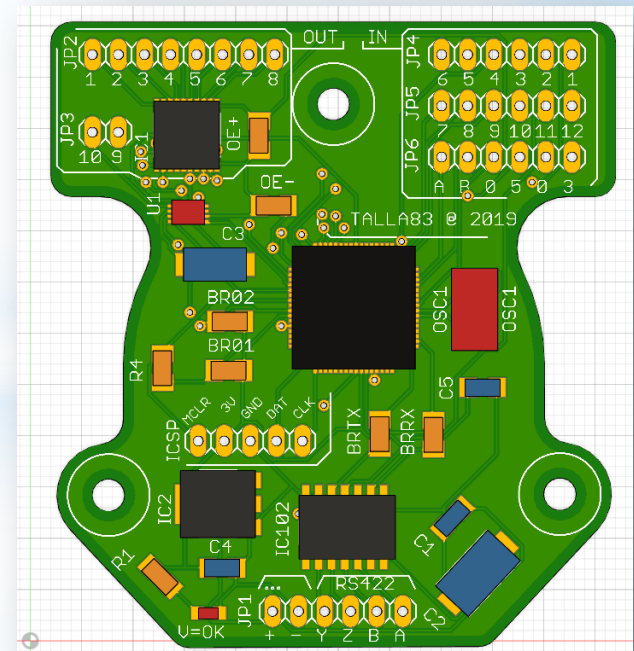
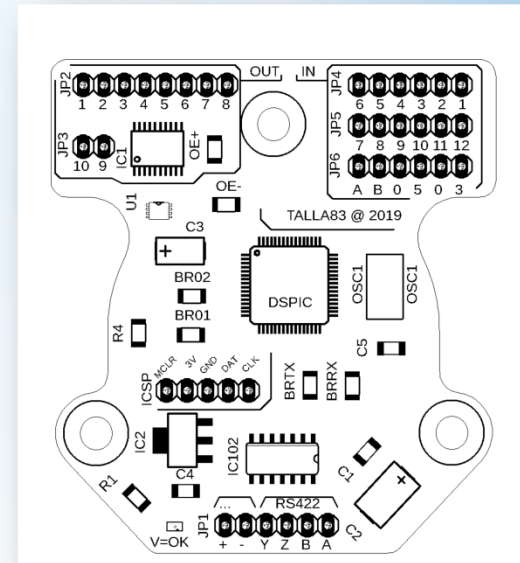
loadrt mux8 names=shw.step-choice
addf shw.step-choice servo-thread

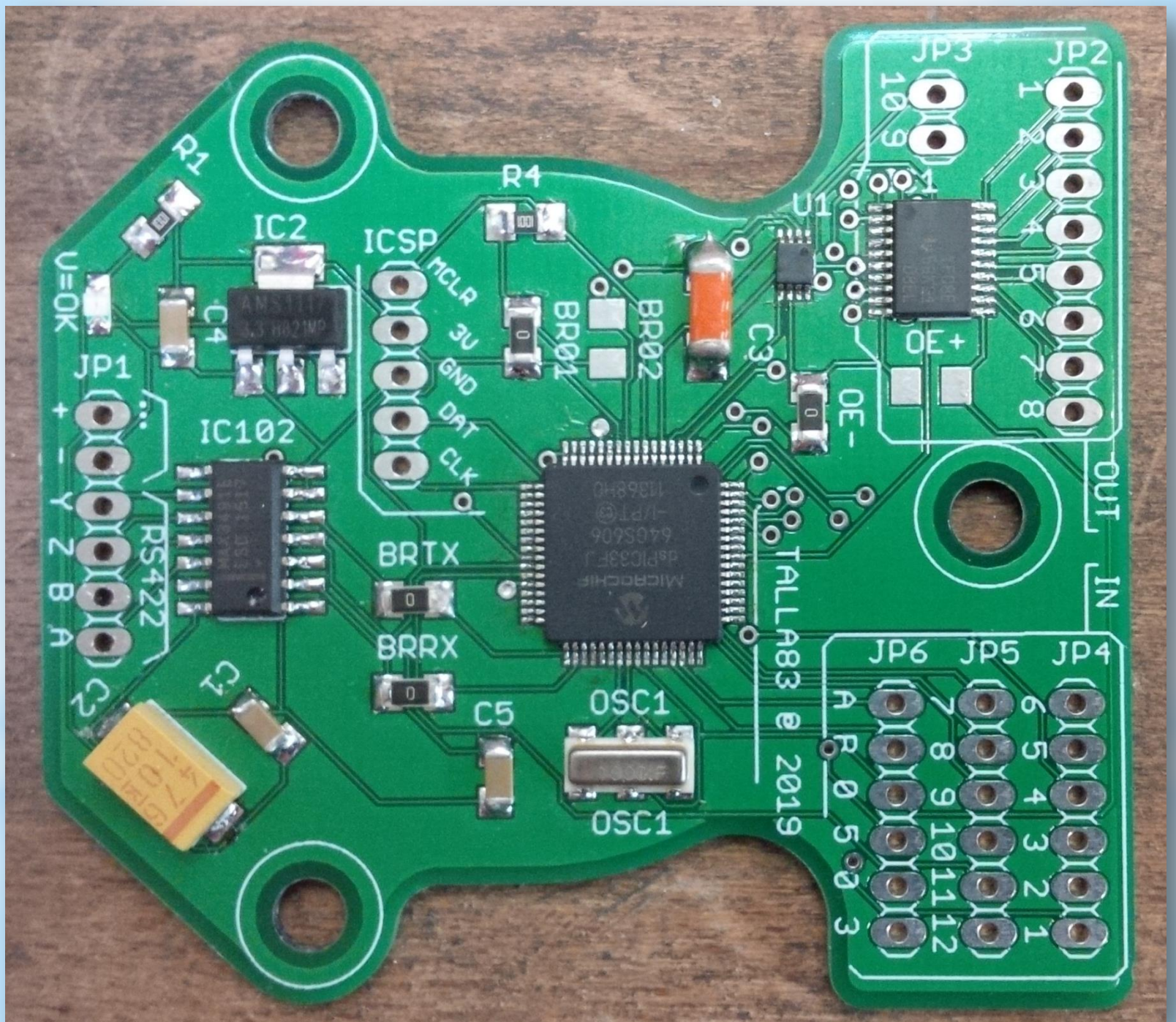
net key-0001 => shw.step-choice.sel0
net key-001 => shw.step-choice.sel1
net key-01 => shw.step-choice.sel2

setp shw.step-choice.in1 0.001
setp shw.step-choice.in2 0.01
```



* Bestückung:

Part	Value	Package
BR01	0 Ohm	M1206
BR02	nicht bestückt	M1206
BRRX	0 Ohm	M1206
BRTX	0 Ohm	M1206
C1	100n	C1206
C2	47µF	CT7343
C3	22µF	CT6032
C4	100n	C1206
C5	100n	C1206
DSPIC	DSPIC33FJ64GS606	TQFP64-10X10
IC1	TXS0108	PW_R-PDSO-G20
IC2	AMS1117 3,3V	SOT223
IC102	MAX3491	SO14
ICSP	nicht bestückt	1X05
JP1	nicht bestückt	1X06
JP2	nicht bestückt	1X08
JP3	nicht bestückt	1X02
JP4	nicht bestückt	1X06
JP5	nicht bestückt	1X06
JP6	nicht bestückt	1X06
OE+	nicht bestückt	M1206
OE-	0 Ohm	M1206
OSC1	10 Mhz	CSTCC10M
R1	1K	R1206
R4	1K	M1206
U1	TXS0102	SOP50P310X90-8N
V=OK	SML0805	SML0805





- * Nur zur Info ! Aktuell gibt es kein offizielles HEX File !
- * Der Microcontroller wird mit einem Programmiergerät verbunden.
- * Danach kann das HEX File übertragen werden.
- * https://youtu.be/L_EpoNiAqbA (Video vom Bestücken und der Programmübertragung)

 MPLAB X IPE v5.15

