

# **„ uLightBox64 Keyboard BT/USB Remote Board“**

(Release V1.00)

Letzte Änderung

Dienstag, 8. März 2022

# Inhaltsverzeichnis

- 1 Allgemein.....2
- 2 uLightBox64 Keyboard BT/USB Remote Board.....3
  - 2.1 Verbinden / Pairing.....4
  - 2.2 Übertragen des HEX Files / Firmware.....5
  - 2.3 Prüfung.....6
  - 2.4 Bootloader flashen (optional).....7

# 1 Allgemein

Das **uLightBox64 Keyboard BT/USB Remote Board** ist das Verbindungsstück zwischen C64 und zu einem normalen Bluetooth oder normalen USB Keyboard.

Für die Übertragung der Bluetooth Signale werden zwei bestimmte Komponenten benötigt:

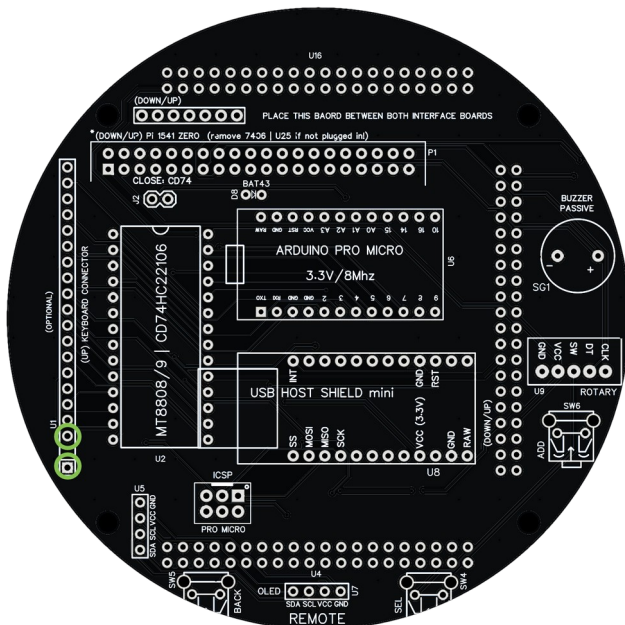
1. Mini USB Host Shield Platine
2. CSR Bluetooth USB Dongle (darauf achten das dies ein „CSR“ Dongle ist!)



Die Stiftleisten für den Pro Micro und das USB Host Shield sollten mit kleinen runden Stiftleistenpins ausgestattet werden, so passen diese in einen Sockel, und auch beim direkten einlöten lassen diese sich so einfacher wieder auslöten.

## 2 uLightBox64 Keyboard BT/USB Remote Board

Das uLightBox64 Keyboard BT/USB Remote Board ist mit einem Arduino Pro Micro ausgestattet, der die Kommunikation über Bluetooth sowie USB für den C64 übernimmt.



Damit die empfangenen Signale an den C64 weitergeleitet werden können, wird ein MT8808 oder ein CD74HC22106 IC benötigt. Je nachdem welches IC man bestückt, sollte der **Jumper CLOSE: CD74** offen oder geschlossen werden

## 2.1 Verbinden / Pairing

Wird ein Bluetooth Keyboard zum ersten Mal verwendet (oder wird der USB Bluetooth CSR Dongle getauscht), muss diese mit der Platine/Dongle gepaart werden (Pairing). Dazu den C64 ausschalten, an dem anderen Ende der Platine ein normales C64 Keyboard mit 20 Pin Stecker anschließen und die „Restore“ Taste gedrückt halten (oder die grün umkreisten Pins kurzschließen). Anschließend dann den C64 einschalten. Somit weiß das Keyboard BT/USB Remote Board das ein neues Bluetooth Keyboard angemeldet wird. Anschließend die Restore Taste loslassen (oder den Kurzschluss der grün umkreisten Pins aufheben)

Hier die einzelnen Schritte in Listen-Form:

1. C64 ausschalten
2. Restore Taste gedrückt halten (oder die grün umkreisten Pins kurzschließen)
3. C64 einschalten
4. Restore Taste loslassen (den Kurzschluss der grün umkreisten Pins aufheben)
5. Bluetooth Keyboard einschalten und auf der Rückseite „Scan“ Taste drücken
6. Auf dem BT Keyboard nacheinander die Tasten 1,2,3,4 und dann die "Return"-Taste drücken

Diesen Vorgang muss nur einmal erledigt werden, und jedes Mal neu, wenn eine andere Bluetooth Tastatur verbunden wird, oder der USB CSR BT Dongle getauscht wird.

Wird der C64 ausgeschaltet und wieder eingeschaltet ist die Bluetooth Verbindung natürlich unterbrochen. Dazu einfach auf der Bluetooth Tastatur eine Taste drücken (nicht die „Scan“ Taste auf der Rückseite drücken!), und nach ca. 2-3 Sekunden ist die Bluetooth Tastatur erneut mit dem C64 verbunden.

## 2.2 Übertragen des HEX Files / Firmware

Das HEX File / Firmware wird über den USB Anschluss des Arduino ProMicro übertragen.

Dazu kann ein HEX Uploader Programm wie zum Beispiel: **HEX Uploader**, **Xloader**, **Arduino Builder**, ... verwendet werden.

Aber es kann auch AVRdude über das Terminal mit vorher installiertem AVRdude benutzt werden, wobei die Kommandozeile etwa so aussehen sollte:

```
stty -f /dev/tty.usbmodem144101 1200; sleep 3; avrdude -v -p atmega32u4 -c avr109 -P  
/dev/cu.usbmodem144101 -b57600 -D -Uflash:w:"C64_Remote_BT_Keyboard_Adapter.hex":i
```

(**Wichtig:** die serielle USB-Schnittstelle „/dev/tty.usbmodem144101“ sollte durch die richtige ersetzt werden, welches der Arduino ProMicro im jeweiligen Betriebssystem gerade verwendet).

Eine erfolgreiche Übertragung sollte so wie rechts zu sehen aussehen:

```
Programmer Type : butterfly  
Description      : Atmel AppNote AVR109 Boot Loader  
  
Connecting to programmer: .  
Found programmer: Id = "CATERIN"; type = S  
Software Version = 1.0; No Hardware Version given.  
Programmer supports auto addr increment.  
Programmer supports buffered memory access with buffersize=128 bytes.  
  
Programmer supports the following devices:  
Device code: 0x44  
  
avrdude: devcode selected: 0x44  
avrdude: AVR device initialized and ready to accept instructions  
  
Reading | ##### | 100% 0.00s  
  
avrdude: Device signature = 0x1e9587 (probably m32u4)  
avrdude: safemode: hfuse reads as D8  
avrdude: safemode: efuse reads as CB  
avrdude: reading input file "Remote_Keyboard_Adapter.hex"  
avrdude: writing flash (9822 bytes):  
  
Writing | ##### | 100% 0.79s  
  
avrdude: 9822 bytes of flash written  
avrdude: verifying flash memory against Remote_Keyboard_Adapter.hex:  
avrdude: load data flash data from input file Remote_Keyboard_Adapter.hex:  
avrdude: input file Remote_Keyboard_Adapter.hex contains 9822 bytes  
avrdude: reading on-chip flash data:  
  
Reading | ##### | 100% 0.09s  
  
avrdude: verifying ...  
avrdude: 9822 bytes of flash verified  
  
avrdude: safemode: hfuse reads as D8  
avrdude: safemode: efuse reads as CB  
avrdude: safemode: Fuses OK (E:CB, H:D8, L:FF)  
  
avrdude done. Thank you.
```

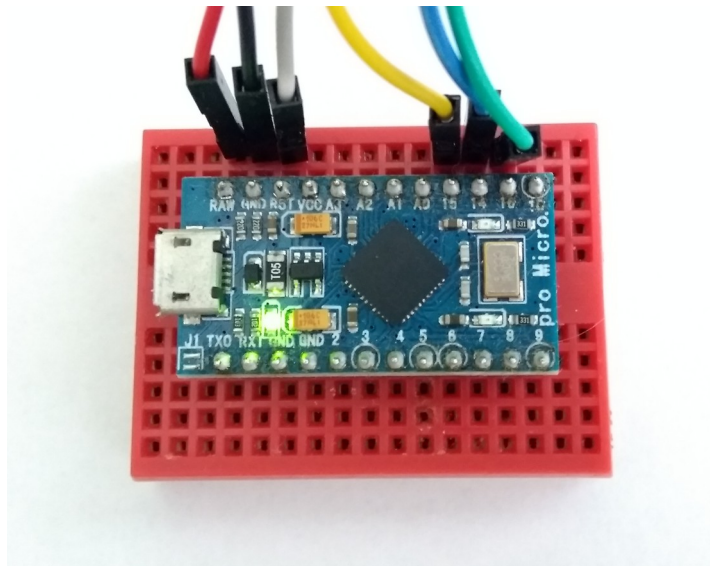
## 2.3 Prüfung

Um sehen zu können ob die Firmware und alles andere richtig überspielt und eingestellt wurde, blinkt die Platine 3x nach etwa 2 Sekunden nach dem Einschalten, sowie bei jedem Tastendruck.

## 2.4 Bootloader flashen (optional)

Eigentlich nur selten notwendig, aber im Fall der Fälle immer zu empfehlen, um auf Nummer sicher zu gehen und den Arduino ProMicro in den Auslieferungszustand zu versetzen.

Der **Bootloader** kann nur über eine **6-Pin ICSP Verbindung** mit einem **ICSP Programmer** und der **Arduino IDE** übertragen werden.



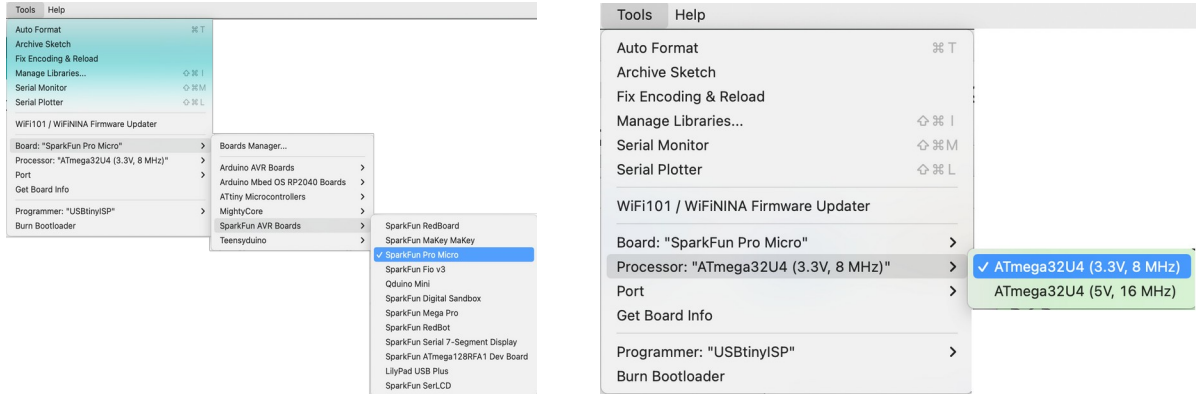
### Pins

RAW (Rot)	5V
GND (Schwarz)	Masse
RST (Weiß)	Reset
15 (Gelb)	SCK (Clock)
14 (Blau)	MISO
16 (Grün)	MOSI



Der Vorgang selbst erfolgt über die installierte **Arduino IDE**

Dazu muss zuerst das richtige Gerät ausgewählt werden:



Anschließend erfolgt die Übertragung des Bootloaders über den Menüpunkt

„**Tools**“ → Burn Bootloader:

