

„Remote Keyboard Adapter und Zubehör“

(Release V1.00)

Letzte Änderung

Freitag, 18. Juni 2021

Inhaltsverzeichnis

1 Allgemein.....	2
2 Remote Keyboard Adapter.....	3
2.1 Bootloader flashen (optional).....	4
2.2 Übertragen des HEX Files / Firmware.....	6
2.3 Master/Slave Mode des HC-05 Bluetooth Modul einstellen.....	7
2.4 Prüfung.....	8
3 Bluetooth 64 Keyboard Platine.....	9
3.1 Bootloader Flashen.....	10
3.2 Übertragen des HEX Files / Firmware.....	12
3.3 Einstellen der Fuses.....	13
3.4 Master/Slave Mode des HC-05 Bluetooth Modul einstellen.....	14
3.5 Prüfung.....	15
3.6 Verbinden.....	16
4 64 MACRO Keyboard.....	17
4.1 Bootloader Flashen.....	18
4.2 Übertragen des HEX Files / Firmware.....	20
4.3 Einstellen der Fuses.....	21
4.4 Master/Slave Mode des HC-05 Bluetooth Modul einstellen.....	22
4.5 Prüfung.....	23
4.6 Verbinden.....	24

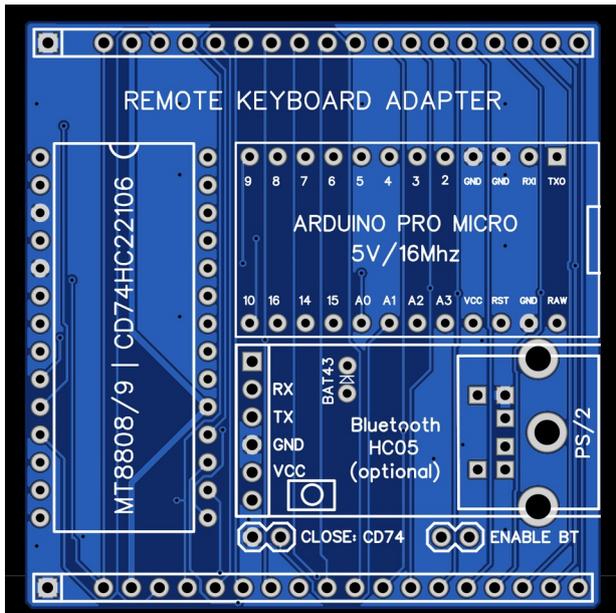
1 Allgemein

Der **Remote Keyboard Adapter** ist das Verbindungsstück zwischen C64 und **PS/2**-, **USB**- und **Bluetooth**-Verbindungen. Mit dessen Hilfe ist es möglich Eingaben, die normalerweise über das große C64 Keyboard mittels Stiftleisten-Matrix empfangen werden, über PS/2-, USB oder Bluetooth-Geräte zu empfangen.

Das kann ein **PS/2-Keyboard**, die **Bluetooth 64 Keyboard Platine** oder das **Macro-Keyboard** sein, sowie die **Android App** mit der man sich über Bluetooth und USB mit der **Remote Keyboard Adapter** Platine verbinden kann.

2 Remote Keyboard Adapter

Der **Remote Keyboard Adapter** ist mit einem Arduino Pro Micro ausgestattet, der die gesamte Kommunikation aller Schnittstellen (**PS/2**, **USB**, **Bluetooth**) für den C64 übernimmt.



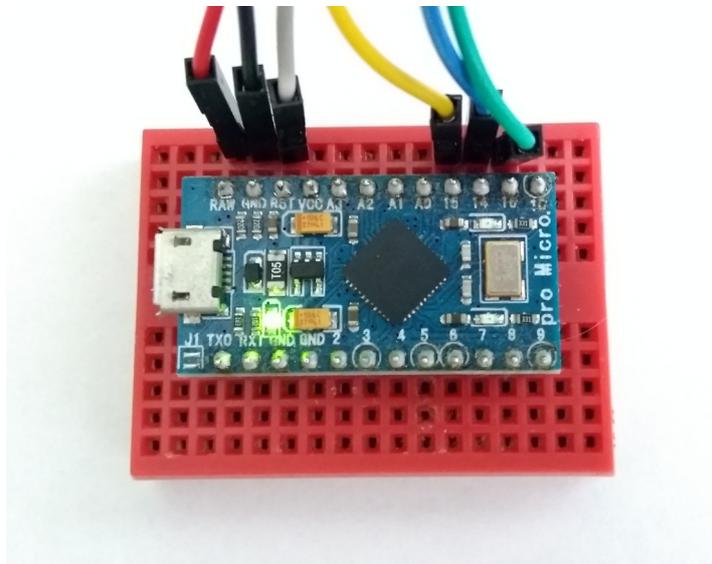
Alle Schnittstellen sind gleichberechtigt, und können alle **gleichzeitig** genutzt werden. Es spielt dabei keine Rolle ob der **ENABLE BT** Jumper gesteckt ist oder nicht, die übrigen Schnittstellen funktionieren immer.

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 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.

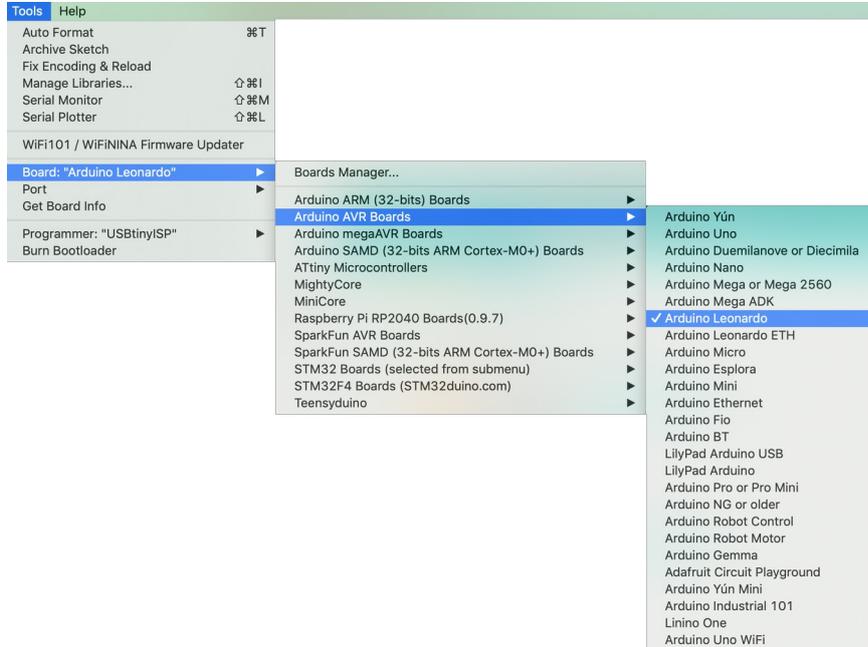


Pins

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

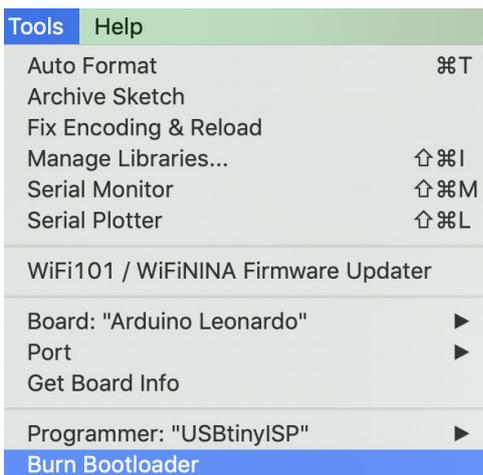
Der Vorgang selbst erfolgt über die installierte **Arduino IDE**

Dazu muss zuerst das richtige Gerät ausgewählt werden:



Anschließend erfolgt die Übertragung des Bootloaders über den Menüpunkt

„**Tools**“ → Burn Bootloader:



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:"Remote_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:

Ob alles richtig übertragen wurde kann man daran erkennen, das ein angeschlossenes PS/2 Keyboard an der Platine funktioniert.

```
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 Master/Slave Mode des HC-05 Bluetooth Modul einstellen

Der **Remote Keyboard Adapter** arbeitet immer im **Slave Mode**. Jedoch ist nicht immer davon auszugehen das der richtige Mode im HC-05 Bluetooth Modul auch eingestellt ist, vor allem wenn man die HC-05 Module ab und zu wechselt.

Um den richtigen Mode möglichst halb automatisiert umstellen zu können, wurde eine Routine in der Firmware des **Remote Keyboard Adapters** eingebaut, die die Umstellung durch kurzes halten des **MODE Taster** auf dem HC-05 Bluetooth Modul selbstständig erledigt.

Dazu folgendermaßen vorgehen:

1. Das C64 Board ausschalten
2. Den **MODE Taster** des Bluetooth Moduls HC-05 drücken und halten
3. Das C64 Board einschalten, und den **MODE Taster** nach 1 Sekunde loslassen
(das Bluetooth Modul blinkt nun zuerst langsam, stellt den Mode um, führt einen eigenen Reset durch, und blinkt dann schnell)

Dann ist der richtige Mode ist eingestellt.

Sollte es nach Punkt 3 beim langsamen Blinken bleiben, war der richtige Mode bereits eingestellt.

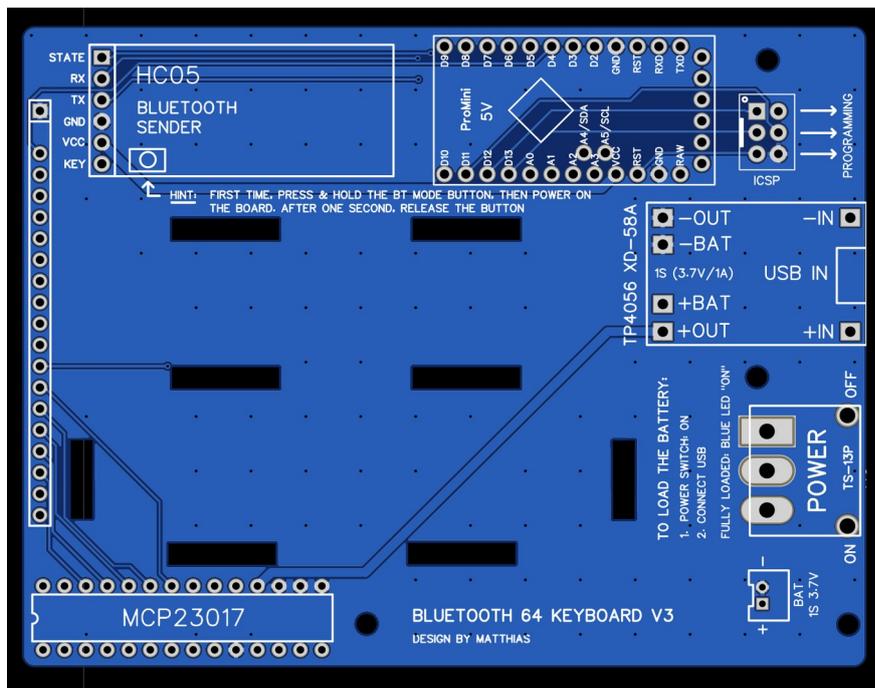
4. Nun das C64 Board ausschalten, so das das Bluetooth Modul keinen Strom mehr bekommt, und dann anschließend das C64 Board wieder einschalten.

2.4 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.

3 Bluetooth 64 Keyboard Platine

Die **Bluetooth 64 Keyboard** Platine ist mit einem Arduino Pro **Mini** ausgestattet, der zusammen mit dem Bluetooth Modul HC-05 als **Sender** fungiert.



Anders als beim **Remote Keyboard Adapter**, arbeitet die **Bluetooth 64 Keyboard** Platine nicht mit 5V, sondern nur mit 3.7V. Deshalb müssen neben dem Bootloader auch die **Fuses** eingestellt werden, die den **BOD-Level auf 2.7V** anstatt 4V stellen.

Für alle folgenden Schritte sollte der 3.7V Lipo Akku angeschlossen sein, und der Powerschalter auf „On“ stehen

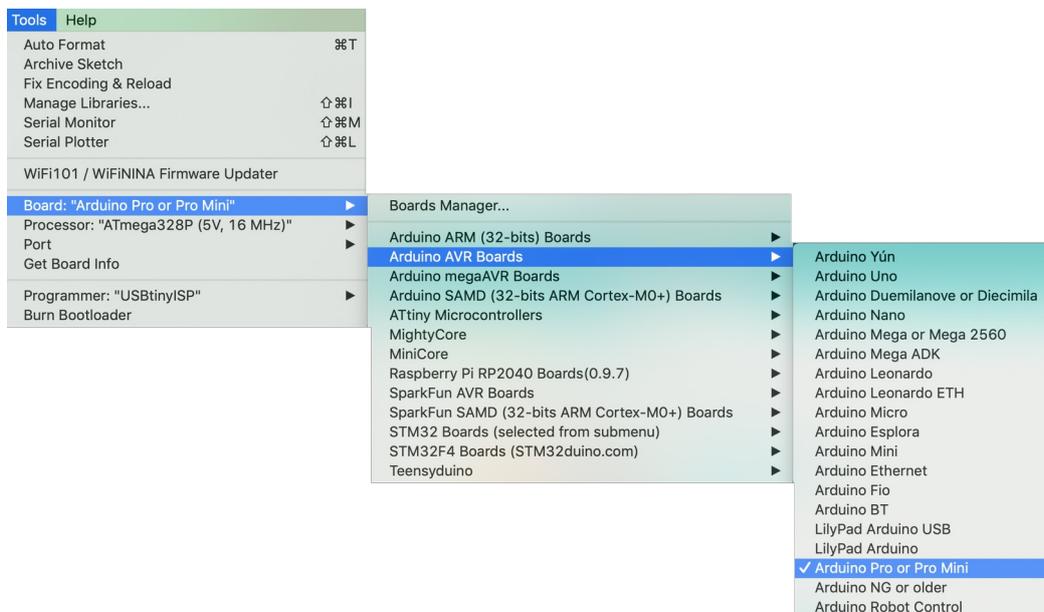
3.1 Bootloader Flashen

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

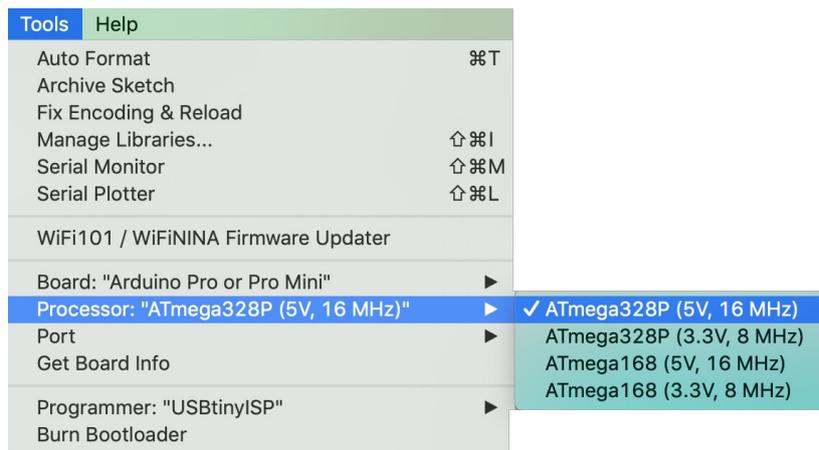


Der Vorgang selbst erfolgt über die installierte **Arduino IDE**

Dazu muss zuerst das richtige Gerät ausgewählt werden:

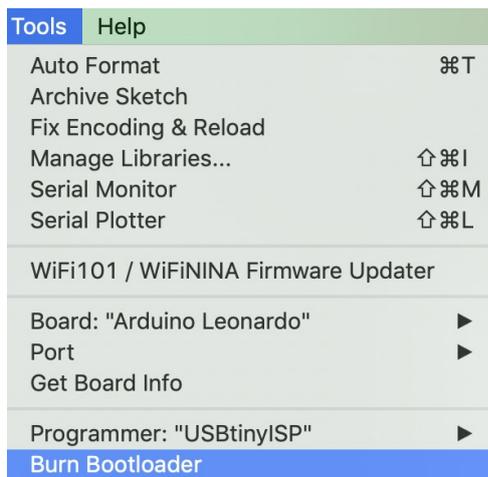


Und der richtige Typ:



Anschließend erfolgt die Übertragung des Bootloders über den Menüpunkt

„**Tools**“ → Burn Bootloader:



3.2 Übertragen des HEX Files / Firmware

Das HEX File / Firmware wird ebenfalls über die auf der Platine vorhandene **6-Pin ICSP Verbindung** mit einem **ICSP Programmer** übertragen werden. Diesmal jedoch nicht über die Arduino IDE, sondern über die **Kommandozeile/Terminal**.



Dazu muss AVRdude installiert sein. Die Kommandozeile sollte etwa so aussehen:

```
avrdude -v -p atmega328p -c usbtiny -D -U flash:w:Firmware.hex
```

(Wichtig: „usbtiny“ sollte durch den eigenen verwendeten ICSP Programmer Namen ersetzt werden)

Eine erfolgreiche Übertragung sollte so wie rechts zu sehen aussehen:

```
avrdude: Using SCK period of 10 usec
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: safemode: hfuse reads as DA
avrdude: safemode: efuse reads as FD
avrdude: reading input file "Firmware.hex"
avrdude: input file Firmware.hex auto detected as Intel Hex
avrdude: writing flash (7540 bytes):

Writing | ##### | 100% 6.70s

avrdude: 7540 bytes of flash written
avrdude: verifying flash memory against Firmware.hex:
avrdude: load data flash data from input file Firmware.hex:
avrdude: input file Firmware.hex auto detected as Intel Hex
avrdude: input file Firmware.hex contains 7540 bytes
avrdude: reading on-chip flash data:

Reading | ##### | 100% 5.25s

avrdude: verifying ...
avrdude: verification error, first mismatch at byte 0x000e
0xc4 != 0xe4
avrdude: verification error; content mismatch

avrdude: safemode: hfuse reads as DA
avrdude: safemode: efuse reads as FD
avrdude: safemode: Fuses OK (E:FD, H:DA, L:FF)

avrdude done. Thank you.
```

3.3 Einstellen der Fuses

Dazu muss AVRdude installiert sein. Die Kommandozeile sollte etwa so aussehen:

```
avrdude -p atmega328p -c usbtiny -U efuse:w:0xfd:m
```

(Wichtig: „usbtiny“ sollte durch den eigenen verwendeten ICSP Programmierer Namen ersetzt werden)

Eine erfolgreiche Übertragung sollte wie folgt aussehen:

```
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: reading input file "0xfd"
avrdude: writing efuse (1 bytes):

Writing | ##### | 100% 0.00s

avrdude: 1 bytes of efuse written
avrdude: verifying efuse memory against 0xfd:
avrdude: load data efuse data from input file 0xfd:
avrdude: input file 0xfd contains 1 bytes
avrdude: reading on-chip efuse data:

Reading | ##### | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of efuse verified

avrdude: safemode: Fuses OK (E:FD, H:DA, L:FF)

avrdude done. Thank you.
```

Anschließend sollte die Platine vom Strom getrennt und wieder verbunden werden.

3.4 Master/Slave Mode des HC-05 Bluetooth Modul einstellen

Die **Bluetooth 64 Keyboard Platine** arbeitet immer im **Master Mode**. Jedoch ist nicht immer davon auszugehen das der richtige Mode im HC-05 Bluetooth Modul auch eingestellt ist, vor allem wenn man die HC-05 Module ab und zu wechselt.

Um den richtigen Mode möglichst halb automatisiert umstellen zu können, wurde eine Routine in der Firmware der **Bluetooth 64 Keyboard Platine** eingebaut, die die Umstellung durch kurzes halten des **MODE Taster** auf dem HC-05 Bluetooth Modul selbstständig erledigt.

Dazu folgendermaßen vorgehen:

1. Die Platine mit dem Schalter auf „OFF“ ausschalten
2. Den **MODE Taster** des Bluetooth Moduls HC-05 drücken und halten
3. Die Platine mit dem Schalter auf „On“ einschalten, und den **MODE Taster** nach 1 Sekunde loslassen

(das Bluetooth Modul blinkt nun zuerst langsam, stellt den Mode um, führt einen eigenen Reset durch, und blinkt dann schnell)

Dann ist der richtige Mode ist eingestellt.

Sollte es nach Punkt 3 beim langsamen Blinken bleiben, war der richtige Mode bereits eingestellt.

4. Nun die Platine noch einmal ausschalten, so das das Bluetooth Modul keinen Strom mehr bekommt, und dann anschließend die Platine wieder einschalten.

3.5 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.

3.6 Verbinden

Ist soweit alles richtig übertragen und eingestellt, kann die **Bluetooth 64 Keyboard** Platine mit dem **Remote Keyboard Adapter** verbunden werden.

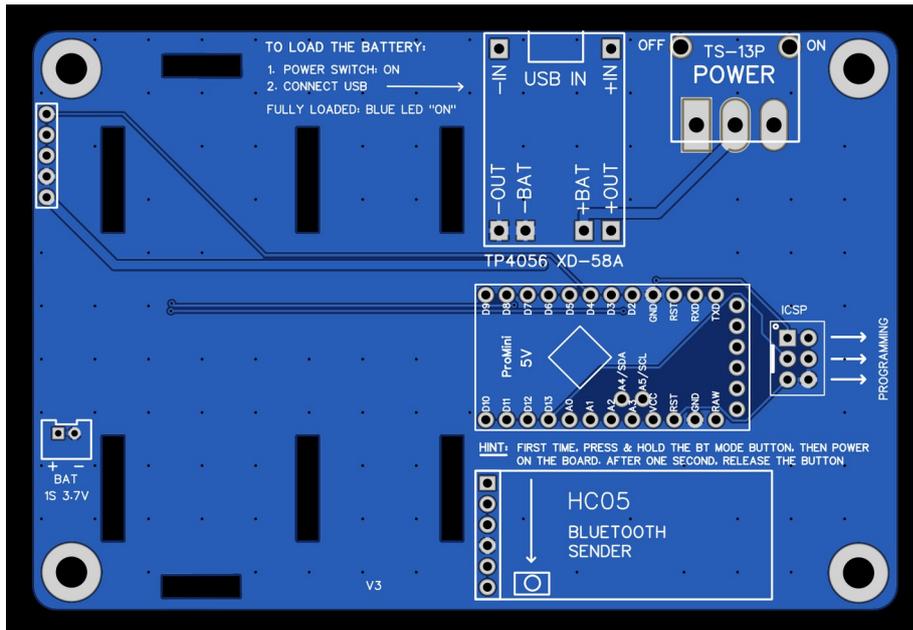
Dazu den C64 mit gesteckter **Keyboard Remote Adapter** Platine einschalten, und auch die **Bluetooth 64 Keyboard** Platine einschalten.

Beide LEDs der jeweiligen HC-05 Bluetooth Module blinken nun schnell.

Nach spätestens 1 Minute sollten beide LEDs nun langsam blinken (verbunden).

4 64 MACRO Keyboard

Auch das **64 MACRO Keyboard** das aus 3 Platinen besteht, ist mit einem Arduino Pro **Mini** ausgestattet, der zusammen mit dem Bluetooth Modul HC-05 als **Sender** fungiert.



Anders als beim **Remote Keyboard Adapter**, arbeitet das **64 Macro Keyboard** nicht mit 5V, sondern nur mit 3.7V. Deshalb müssen neben dem Bootloader auch die **Fuses** eingestellt werden, die den **BOD-Level auf 2.7V** anstatt 4V stellen.

Für alle folgenden Schritte sollte der 3.7V Lipo Akku angeschlossen sein, und der Powerschalter auf „On“ stehen

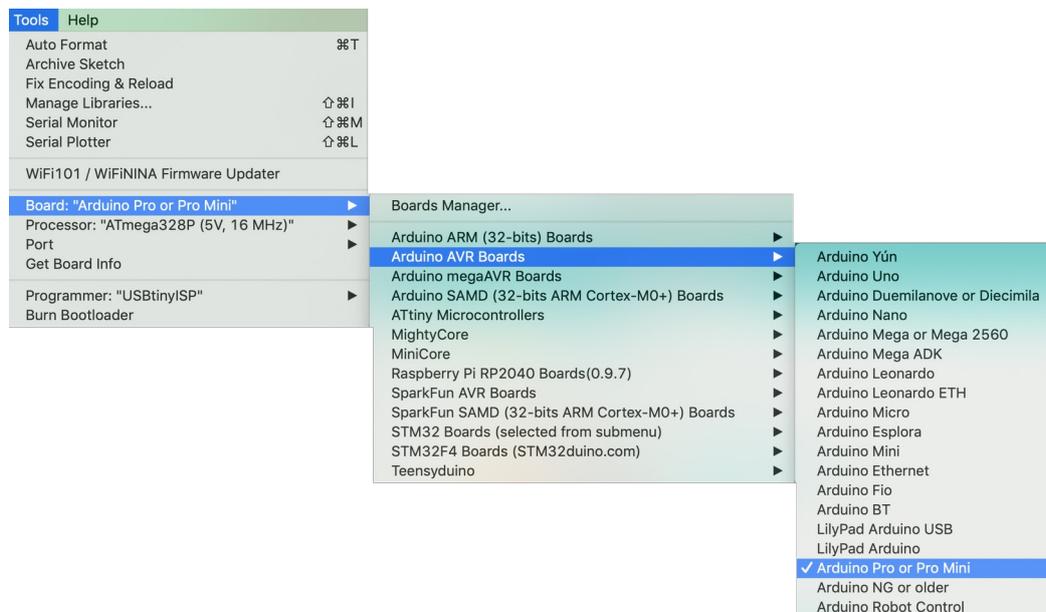
4.1 Bootloader Flashen

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

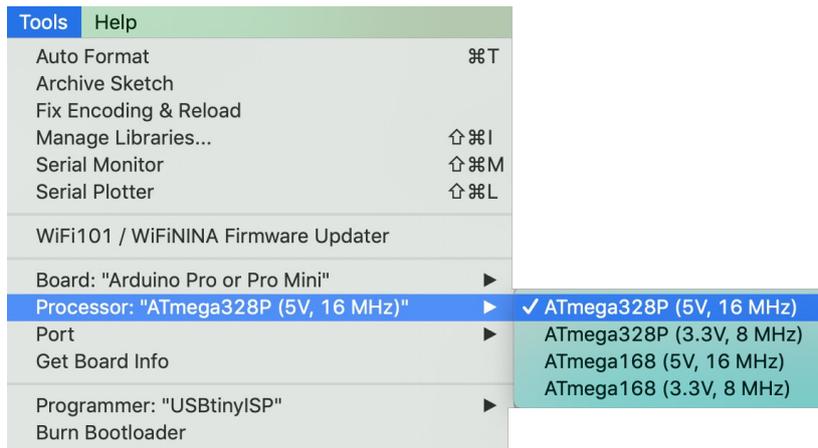


Der Vorgang selbst erfolgt über die installierte **Arduino IDE**

Dazu muss zuerst das richtige Gerät ausgewählt werden:

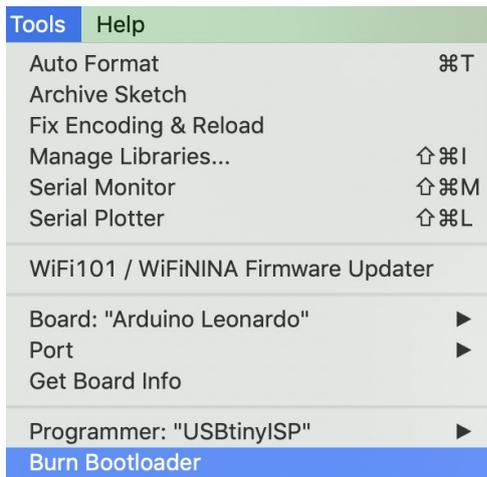


Und der richtige Typ:



Anschließend erfolgt die Übertragung des Bootloders über den Menüpunkt

„**Tools**“ → Burn Bootloader:



4.2 Übertragen des HEX Files / Firmware

Das HEX File / Firmware wird ebenfalls über die auf der Platine vorhandene **6-Pin ICSP Verbindung** mit einem **ICSP Programmer** übertragen werden. Diesmal jedoch nicht über die Arduino IDE, sondern über die **Kommandozeile/Terminal**.



Dazu muss AVRdude installiert sein. Die Kommandozeile sollte etwa so aussehen:

```
avrdude -v -p atmega328p -c usbtiny -D -U flash:w:Firmware.hex
```

(Wichtig: „usbtiny“ sollte durch den eigenen verwendeten ICSP Programmer Namen ersetzt werden)

Eine erfolgreiche Übertragung sollte so wie rechts zu sehen aussehen:

```
avrdude: Using SCK period of 10 usec
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: safemode: hfuse reads as DA
avrdude: safemode: efuse reads as FD
avrdude: reading input file "Firmware.hex"
avrdude: input file Firmware.hex auto detected as Intel Hex
avrdude: writing flash (7540 bytes):

Writing | ##### | 100% 6.70s

avrdude: 7540 bytes of flash written
avrdude: verifying flash memory against Firmware.hex:
avrdude: load data flash data from input file Firmware.hex:
avrdude: input file Firmware.hex auto detected as Intel Hex
avrdude: input file Firmware.hex contains 7540 bytes
avrdude: reading on-chip flash data:

Reading | ##### | 100% 5.25s

avrdude: verifying ...
avrdude: verification error, first mismatch at byte 0x000e
0xc4 != 0xe4
avrdude: verification error; content mismatch

avrdude: safemode: hfuse reads as DA
avrdude: safemode: efuse reads as FD
avrdude: safemode: Fuses OK (E:FD, H:DA, L:FF)

avrdude done. Thank you.
```

4.3 Einstellen der Fuses

Dazu muss AVRdude installiert sein. Die Kommandozeile sollte etwa so aussehen:

```
avrdude -p atmega328p -c usbtiny -U efuse:w:0xfd:m
```

(Wichtig: „usbtiny“ sollte durch den eigenen verwendeten ICSP Programmierer Namen ersetzt werden)

Eine erfolgreiche Übertragung sollte wie folgt aussehen:

```
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: reading input file "0xfd"
avrdude: writing efuse (1 bytes):

Writing | ##### | 100% 0.00s

avrdude: 1 bytes of efuse written
avrdude: verifying efuse memory against 0xfd:
avrdude: load data efuse data from input file 0xfd:
avrdude: input file 0xfd contains 1 bytes
avrdude: reading on-chip efuse data:

Reading | ##### | 100% 0.00s

avrdude: verifying ...
avrdude: 1 bytes of efuse verified

avrdude: safemode: Fuses OK (E:FD, H:DA, L:FF)

avrdude done. Thank you.
```

Anschließend sollte die Platine vom Strom getrennt und wieder verbunden werden.

4.4 Master/Slave Mode des HC-05 Bluetooth Modul einstellen

Das **64 Macro Keyboard** arbeitet immer im **Master Mode**. Jedoch ist nicht immer davon auszugehen das der richtige Mode im HC-05 Bluetooth Modul auch eingestellt ist, vor allem wenn man die HC-05 Module ab und zu wechselt.

Um den richtigen Mode möglichst halb automatisiert umstellen zu können, wurde eine Routine in der Firmware des **64 Macro Keyboards** eingebaut, die die Umstellung durch kurzes halten des **MODE Taster** auf dem HC-05 Bluetooth Modul selbstständig erledigt.

Dazu folgendermaßen vorgehen:

1. Die Platine mit dem Schalter auf „OFF“ ausschalten
2. Den **MODE Taster** des Bluetooth Moduls HC-05 drücken und halten
3. Die Platine mit dem Schalter auf „On“ einschalten, und den **MODE Taster** nach 1 Sekunde loslassen

(das Bluetooth Modul blinkt nun zuerst langsam, stellt den Mode um, führt einen eigenen Reset durch, und blinkt dann schnell)

Dann ist der richtige Mode ist eingestellt.

Sollte es nach Punkt 3 beim langsamen Blinken bleiben, war der richtige Mode bereits eingestellt.

4. Nun das 64 Macro Keyboard noch einmal ausschalten, so das das Bluetooth Modul keinen Strom mehr bekommt, und dann anschließend die Platine wieder einschalten.

4.5 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.

4.6 Verbinden

Ist soweit alles richtig übertragen und eingestellt, kann das **64 Macro Keyboard** mit dem **Remote Keyboard Adapter** verbunden werden.

Dazu den C64 mit gesteckter **Keyboard Remote Adapter** Platine einschalten, und auch das **64 Macro Keyboard** einschalten.

Beide LEDs der jeweiligen HC-05 Bluetooth Module blinken nun schnell.

Nach spätestens 1 Minute sollten beide LEDs nun langsam blinken (verbunden).