

IRMP auf STM32 - Bauanleitung

Copyright (C) 2014-2015 Jörg Riechardt

Inhaltsverzeichnis

- [1 Vorwort](#)
- [2 Kauf](#)
- [3 Firmware bauen \(Linux\)](#)
- [4 Firmware bauen \(Windows\)](#)
- [5 Programmer](#)
- [6 Target](#)
- [7 Flashen \(Linux\)](#)
- [8 Flashen \(Windows\)](#)
- [9 Anschlüsse](#)
- [10 Wakeup anlernen](#)
- [11 Testen \(Linux\)](#)
- [12 Testen \(Windows\)](#)
- [13 Aufsteckplatine](#)
- [14 Bootloader \(Linux\)](#)
- [15 Automatisches Starten/Beenden von irmpircd\(*\) beim An/Abstecken mit systemd](#)
- [16 Minimalistischer Aufbau für erfahrene Bastler](#)

Vorwort

Diese Anleitung ist für zwei ST-Links. Man kann auch ein Developer Board und einen USB-Seriell-TTL Adapter nehmen. Die ST-Links im USB Stick Gehäuse sind wie rote ST-Links beschaltet.

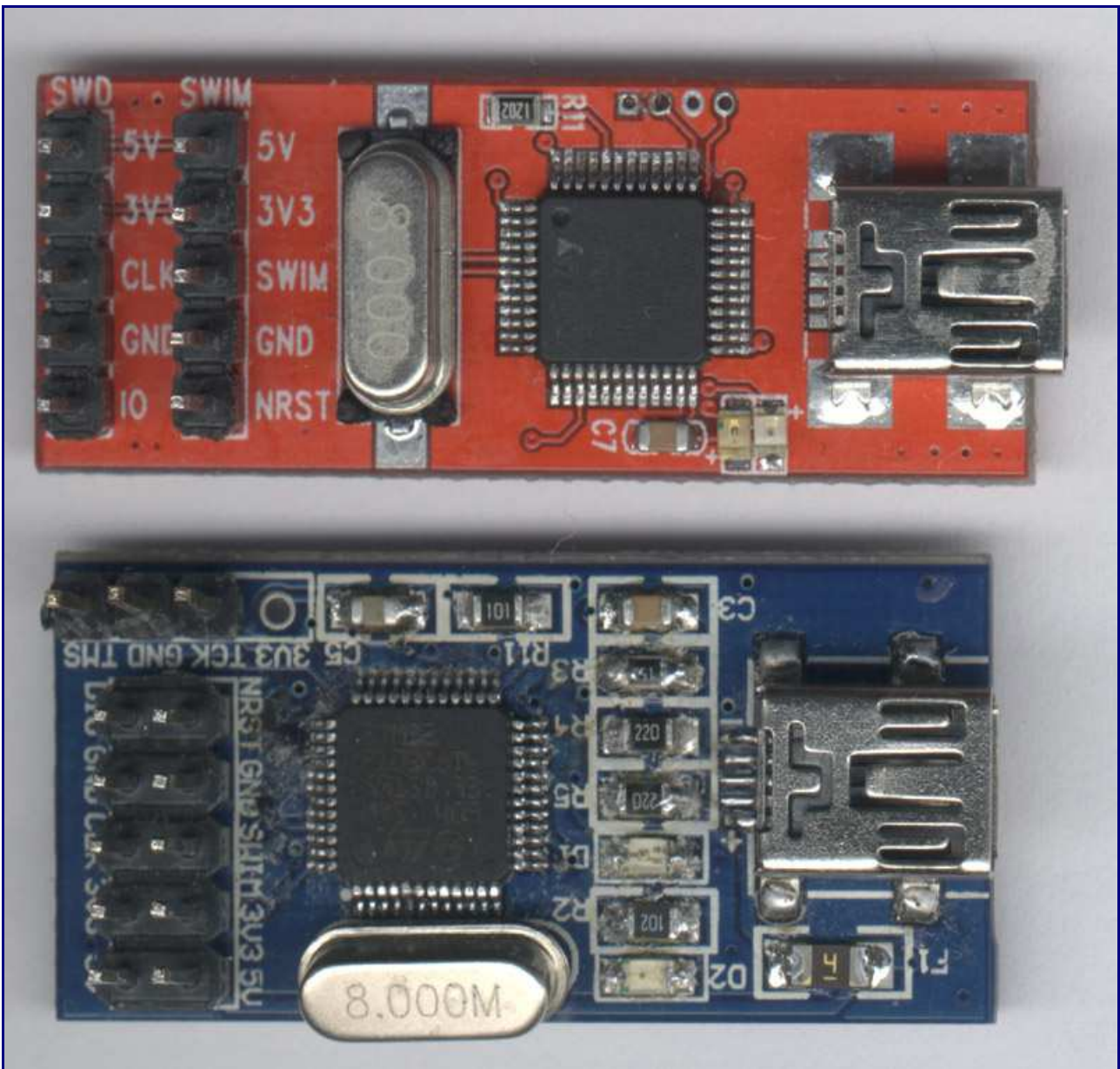
Kauf

Man braucht zwei ST-Link V2 Emulatoren.

Eine ebay Suche nach "ST-Link V2 Emulator" zeigt, was es gerade gibt.

Ich habe meine Roten beim Verkäufer e_goto für 4.99 \$ gekauft, also ca. 4 €.

Die Blauen gibt es bei <http://www.aliexpress.com/item/STLINK-V2-STM8-STM32-emulator-programmer-mini-STLINK-downloader/1446941850.html>



In den Blauen habe ich 3 Pins zusätzlich eingelötet für TCK und TMS
 Beim Blauen sind CLK und TCK sowie DIO und TMS vertauscht

Firmware bauen (Linux)

gcc-arm-none-eabi wird zum Kompilieren gebraucht. Wenn es nicht schon über die Distribution verfügbar ist, kann man es sich von <https://launchpad.net/gcc-arm-embedded/+download> holen.

```
git clone git://github.com/j1rie/IRMP\_STM32.git IRMP_STM32
```

```
cd IRMP_STM32
```

```
cd F103
```

```
./make
```

Firmware bauen (Windows)

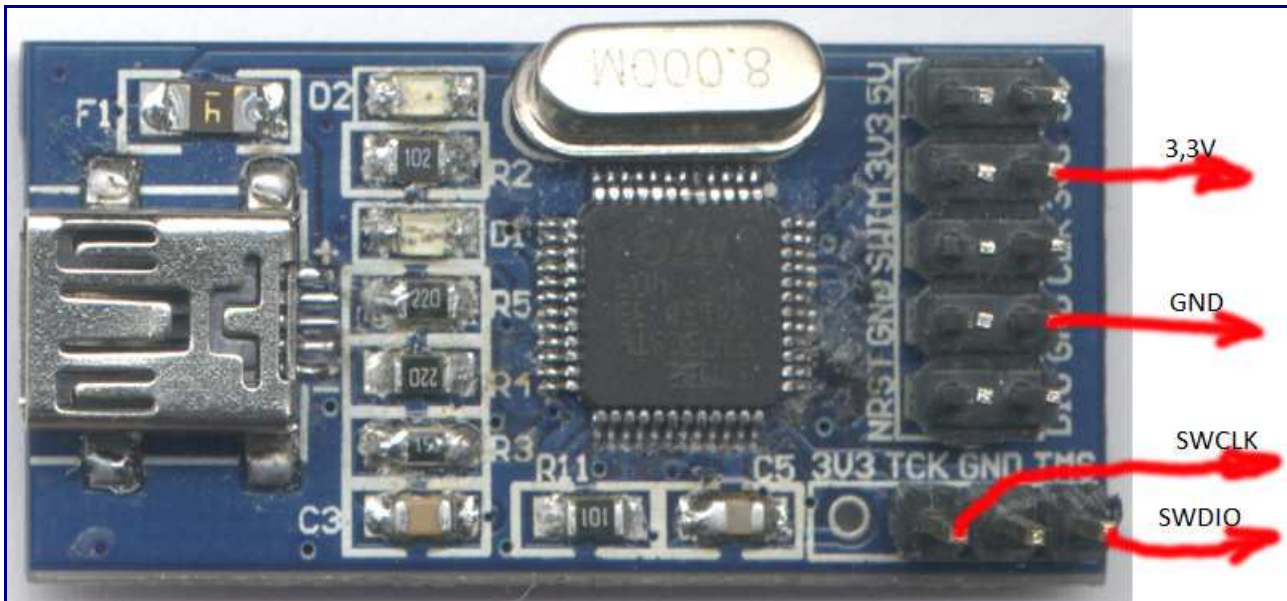
https://github.com/j1rie/IRMP_STM32

IRMP_STM32-master.zip herunter laden und auspacken

in scripts prepare.bat ausführen

IR.coproj in CoCoX laden und kompilieren

Programmer

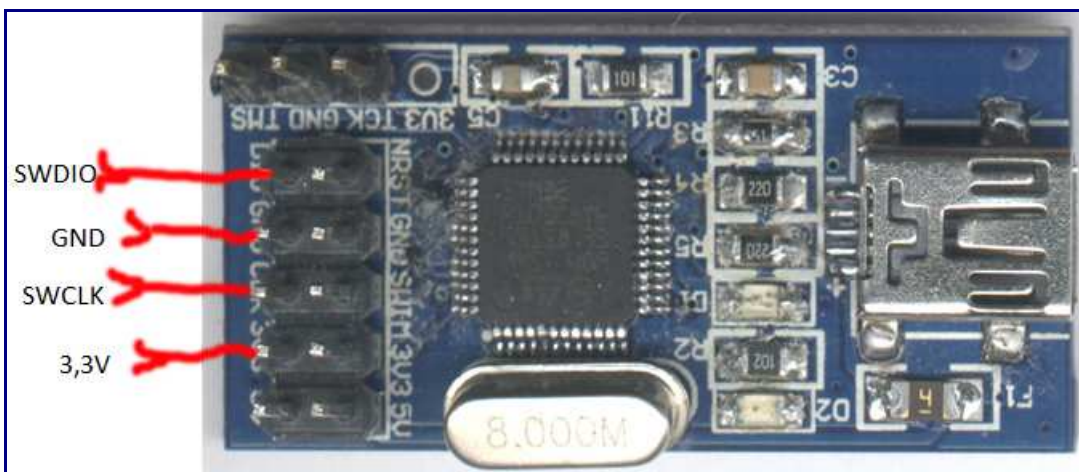


Der Blaue als Programmer

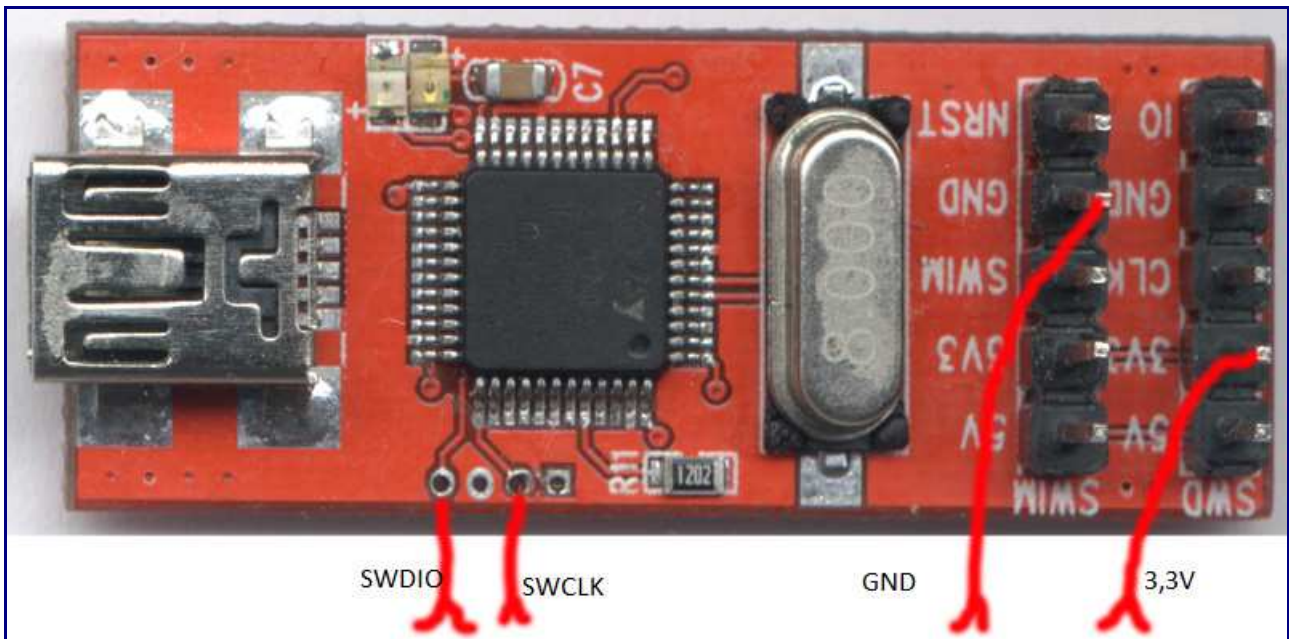


Der Rote als Programmer

Target



Der Blaue als Target



Der Rote als Target.

Damit der TSOP an SWIM funktioniert, muss möglicherweise der PullUp an SWIM entfernt werden (Bild unten). Ich habe einen (uralten) TSOP, der mit PullUp nicht funktioniert. Ein TSOP 34136 funktioniert aber auch mit dem Widerstand.

Wenn man einen Roten als Programmer und einen Blauen als Target nimmt, muss man nichts löten (wenn man auf den Bootloader verzichtet). Oder man kann auch einfach während des Flashens zwei Kontaktstifte an SWDIO und SWCLK pressen.

Flashen (Linux)

Programmer und Target werden verbunden, SWCLK - SWCLK, SWDIO - SWDIO, 3,3V - 3,3V, Gnd -Gnd.

Für das erste Flashen ist es nötig, den Controller mit folgendem openocd-Befehl zu entsperren:
`$ openocd -f interface/stlink-v2.cfg -f target/stm32f1x_stlink.cfg -c "init" -c "halt" -c "stm32f1x unlock 0" -c "shutdown"`

Danach kann die Firmware geflasht werden:

```
git clone git://github.com/texane/stlink.git
```

```
cd stlink
```

```
./autogen.sh
```

```
./configure
```

```
./make
```

```
./st-flash erase
```

```
./st-flash --reset write /IRMP_STM32/F103C8/IR.bin 0x8000000
```

In manchen Fällen ist es nötig, zu Beginn des Flashens den ST-Link zu resettet. Dazu muss die Stelle im Bild auf Masse gezogen werden und kurz nach Absetzen des Flash Befehls wieder los gelassen werden.

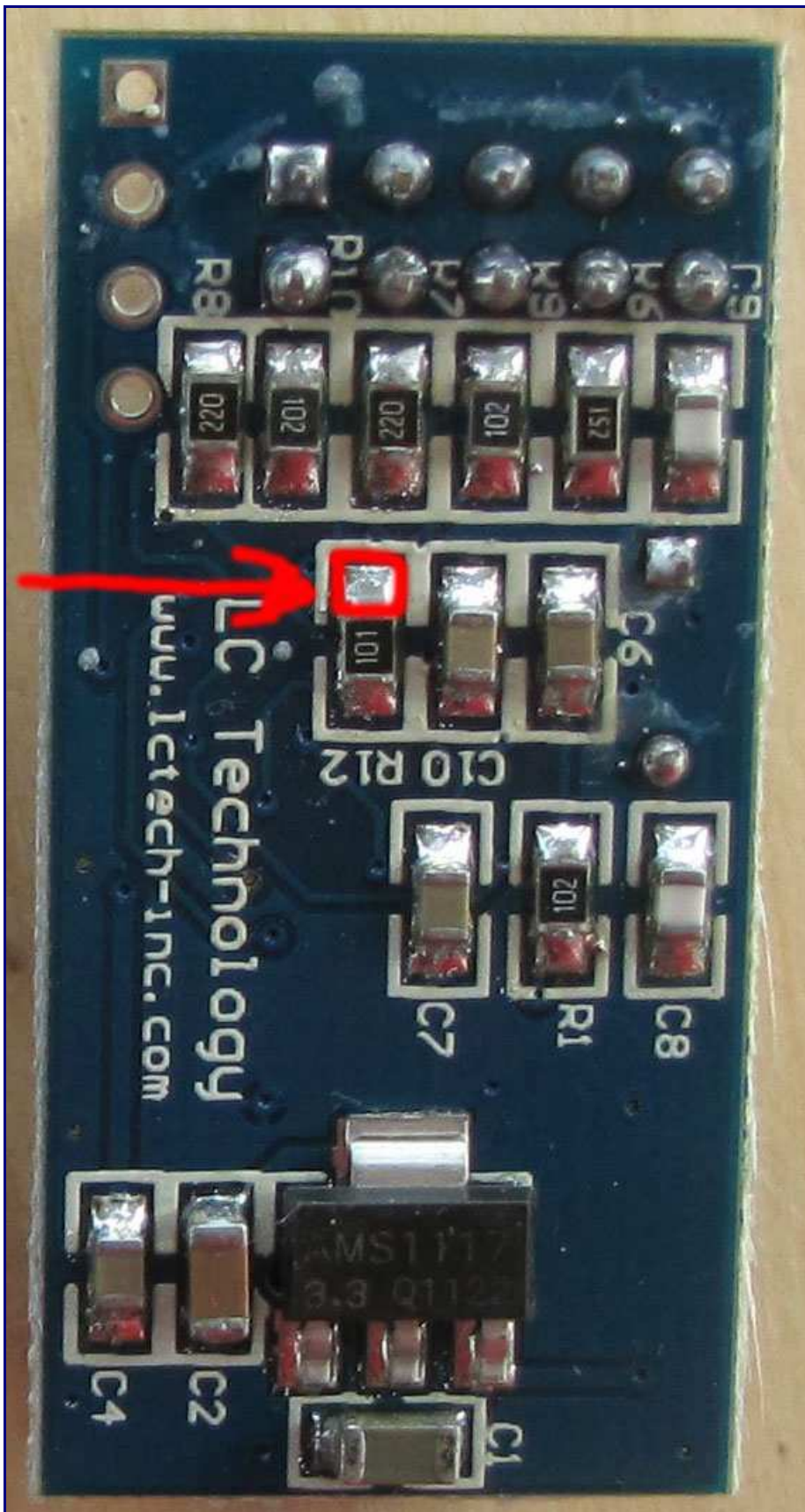
Flashen (Windows)

Mit dem STM32 ST-LINK Utility.

Verbinden und resettet wie unter Linux.

Entsperren: -> Target -> Option Bytes, die Read Out Protection auf Disabled stellen.

Flashen: -> Target -> Program & Verify

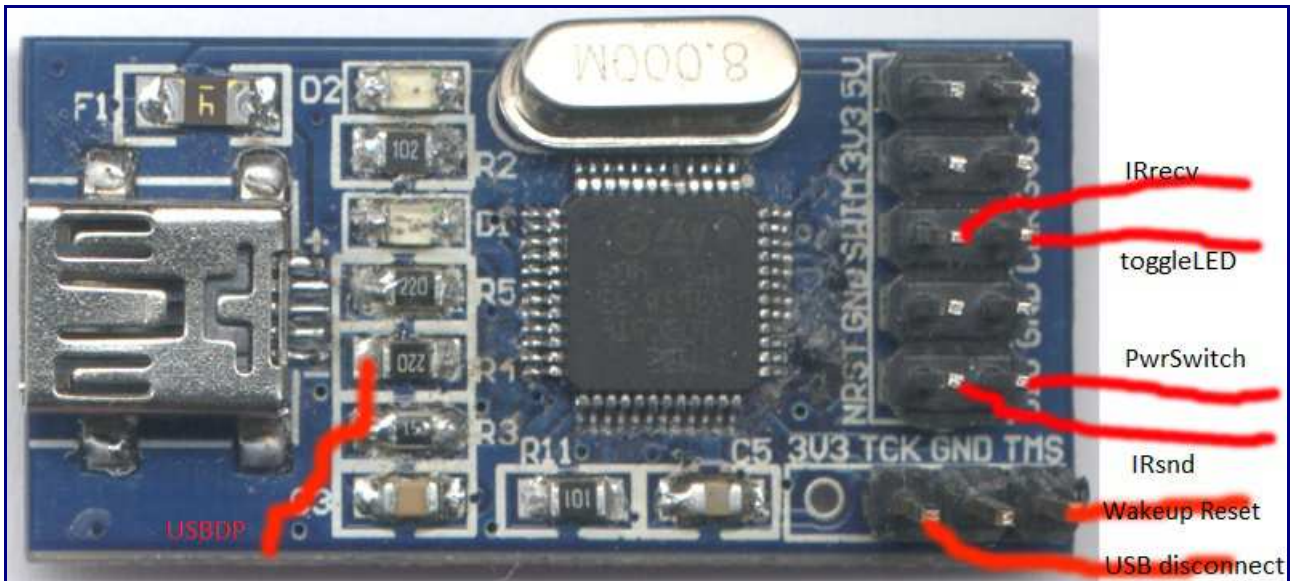


Für Reset das rot Umrahmte auf Masse ziehen.

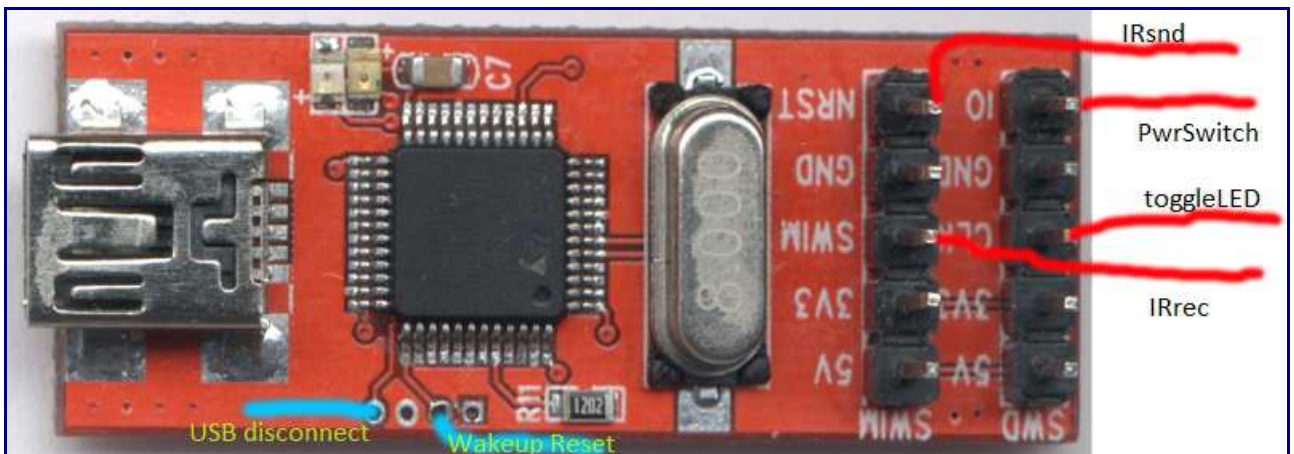


Für Reset das gelb Umrahmte auf Masse ziehen. Grün umrahmt der PullUp an SWIM.

Anschlüsse



Den Blauen verbinden.



Den Roten verbinden.

Wakeup anlernen

Falls das erste Wakeup leer ist, werden automatisch die ersten empfangenen IR Daten ins erste Wakeup gespeichert.

Für das erneute Anlernen zieht man den Wakeup Reset Pin kurz auf Masse und drückt in den nächsten 5 Sekunden die Taste, die angelernt werden soll.

Man kann die neuen Wakeup IR Daten auch manuell oder per Fernbedienung mit dem Konfigurationsprogramm eingeben.

Testen (Linux)

Mit stm32IRconfig_gui: ST-Link anschliessen, stm32IRconfig_gui starten, "receive mode" drücken und testen.

Mit stm32IRconfig: ST-Link anschliessen, stm32IRconfig starten, m eingeben für Monitor-Modus und testen.

Mit irimplircd: ST-Link anschliessen, irimplircd und irw starten und testen.

Testen (Windows)

Mit stm32IRconfig_gui: ST-Link anschliessen, stm32IRconfig_gui starten, "receive mode" drücken und testen.

Mit stm32IRconfig: ST-Link anschliessen, stm32IRconfig starten, m eingeben für Monitor-Modus und testen.

Mit SimpleHIDWrite: ST-Link anschliessen, in SimpleHIDWrite auswählen und testen.

<http://janaxelson.com/hidpage.htm#tools>

Aufsteckplatine

Es gibt eine Aufsteckplatine. Mit dem Steckverbinder auf der einen Seite passt sie auf den Blauen, und mit dem Steckverbinder auf der anderen Seite passt sie auf den Roten.

[Schaltplan](#)

<http://www.vdr-portal.de/board73-marktplatz/board75-verkaufen/p1220193-usb-ir-receiver-inkl-einschalter-open-source-sw-hw/#post1220193>

Bootloader (Linux)

```
git clone -b unify_platforms git://github.com/jlrie/maple-bootloader.git
```

```
cd maple-bootloader
```

```
make Platform=Dev (Developer Board) bzw.
```

```
make Platform=Red (Roter) bzw.
```

```
make Platform=Blue (Blauer)
```

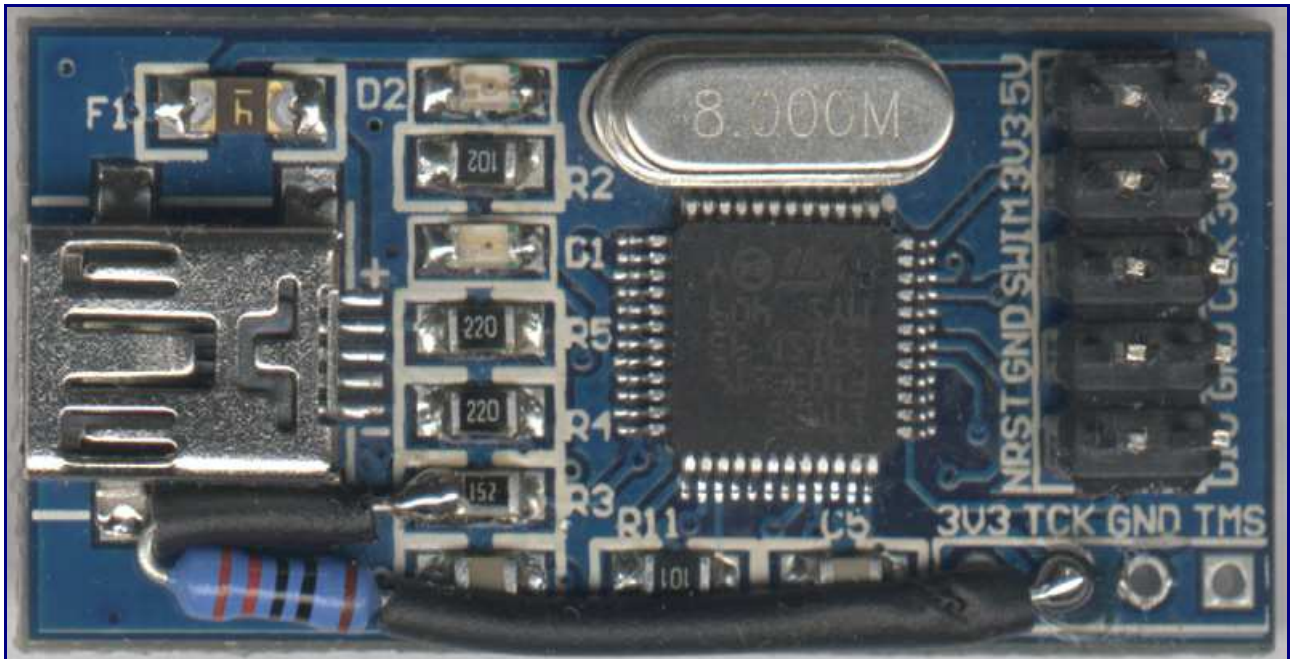
```
Dann ./build/maple_boot.bin flashen.
```

```
Danach wird dann so geflasht:
```

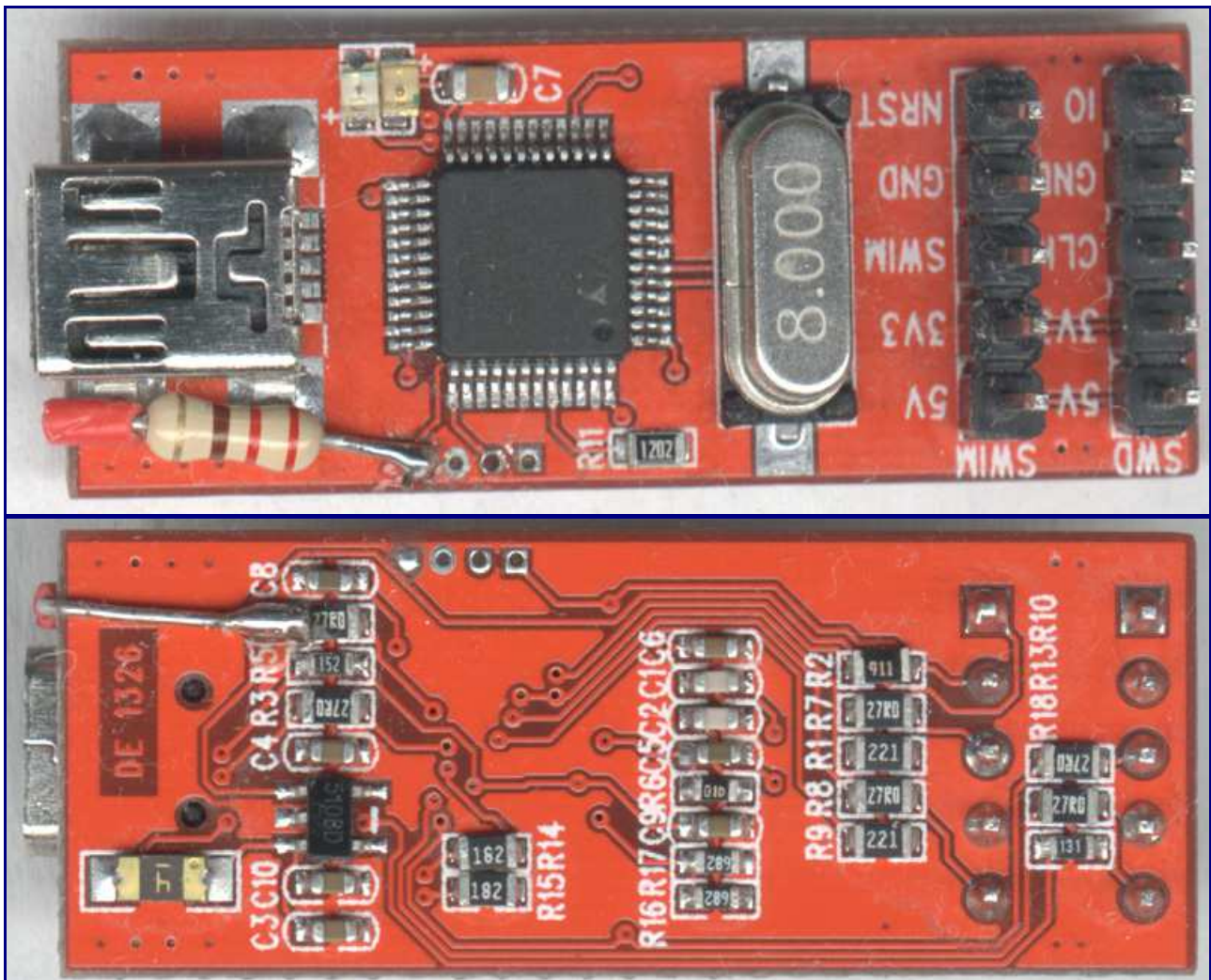
```
Zuerst /path/FlashDFUSe.sh /path/neueFirmware.bin* ausführen,
```

```
dann den ST-Link anstecken.
```

(*) Die Firmware muss mit Bootloader=1 kompiliert sein.



Der Blaue mit Pulldown Widerstand zwischen USB DP und USB Disconnect Pin.



Der Rote mit Pulldown Widerstand zwischen USB DP und USB Disconnect Pin.

Automatisches Starten/Beenden von `irmplircd(*)` beim An/Abstecken mit `systemd`

80-irmp.rules aus `IRMP_STM32/irmplircd/` nach `/etc/udev/rules.d/` kopieren.

`irmplircd.service` aus `IRMP_STM32/irmplircd/` nach `/etc/systemd/system/` kopieren.

Service aktivieren mit `'systemctl enable irmplircd.service'`.

Die Pfade zu `irmplircd` und `irmp_stm32.map` in `irmplircd.service` falls nötig anpassen.

Die Übersetzungstabelle `irmp_stm32.map` kann mit `stm32IRconfig_gui` erzeugt und angepasst werden.

(*) <https://github.com/realglotzi/irmplircd>

Minimalistischer Aufbau für erfahrene Bastler

Nur für IR Empfang und PC Einschalten. Der TSOP ist direkt mit den entsprechenden Pins verbunden und der aktive Einschaltpin des Mainboards wird über den Widerstand (220 Ohm) herunter gezogen (SimpleCircuit = 1). Wer weiß, was er tut, kann den Widerstand auch weg lassen. Bitte vorsichtig sein und nicht das Motherboard beschädigen!

