IRMP auf STM32 - ein USB IR Empfänger/Sender/Einschalter mit Wakeup-Timer

Ein IR Fernbedienungsempfänger mit vielen Funktionen auf Basis preiswerter Hardware.

Inhaltsverzeichnis

Einleitung

Für ca. 4 - 8 € kann man auf ebay einen ST-Link Emulator oder ein STM32F103 Board kaufen, meistens in China hergestellt. Auf den darauf befindlichen STM32F103 Mikrocontroller wird eine Open Source Firmware mit vielen Funktionen geflasht.

Funktionen

- Anschluss über USB
- meldet sich als HID Device an, kein Treiber nötig, erscheint als „HID konformes Gerät“ oder /dev/hidraw
- überträgt die Daten per USB2 in Hardware mit Fullspeed
- IR Empfänger (ca. 40 Protokolle werden in HW dekodiert)
- PC mit Fernbedienung Einschalten aus S3 (STR) und S4 (STD) über USB oder aus S3, S4 und S5 (SoftOff) über Motherboard-Schalter (+5V nötig, auf USB oder vom Netzteil/Motherboard zugeführt)
- PC mit eingebautem Timer Einschalten aus S3 (STR) und S4 (STD) über USB oder aus S3,

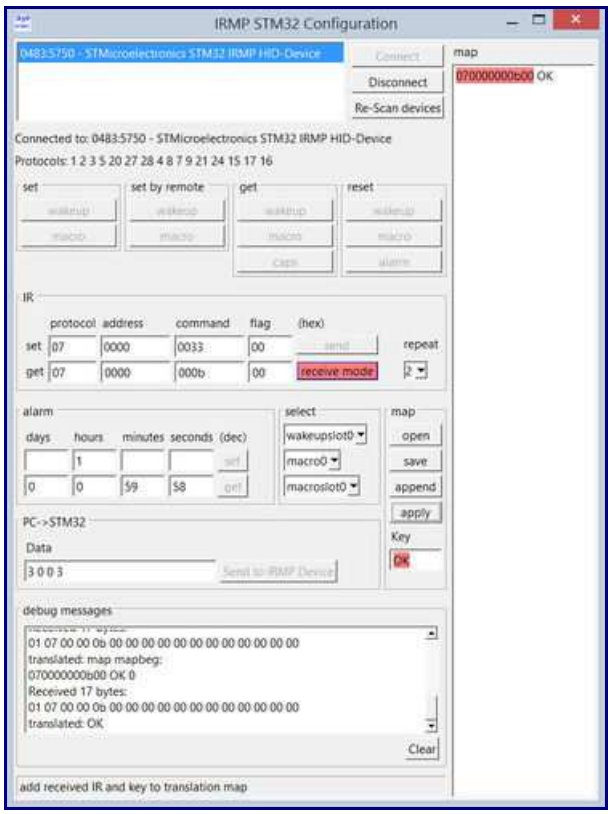
S4 und S5 (SoftOff) über Motherboard-Schalter (+5V nötig, auf USB oder vom Netzteil/Motherboard zugeführt)
- IR Sender (ca. 40 Protokolle)
- die Konfiguration wird im emulierten Eeprom gespeichert
- im Rahmen der Größe des Eeproms beliebig viele Makros mit beliebiger Tiefe und beliebig viele Wakeups: MACRO_SLOTS x (MACRO_DEPTH + 1) + WAKE_SLOTS < 85 bei F103, < 170 bei F105
- Bootloader für bequemes Firmware Update

Software Linux

- GUI Konfig-Tool stm32IRconfig_gui für die Konfiguration: Wakeup Code, Makros, Alarmzeit setzen, auslesen und zurücksetzen, IR senden und Anzeigen der empfangenen IR Codes. Wakeups und Makros können auch per Fernbedienung progammiert werden. Erstellen der Übersetzungstabelle mit Fernbedienung und Maus, sowie testen und bearbeiten. Detaillierte Ausgaben zur Analyse.
- Kommandozeilenprogramm stm32IRconfig für die Konfiguration: Wakeup Code, Makros, Alarmzeit setzen, auslesen und zurücksetzen, IR senden und Anzeigen der empfangenen IR Codes. Wakeups und Makros können auch per Fernbedienung progammiert werden.
- stm32IRalarm zum Alarmzeit setzen und auslesen per Skript
- irmplircd ist ein Daemon, der im Hintergrund als eigenständiger lirc Server läuft und die IR Codes/events an die Anwendung weitergibt https://github.com/realglotzi/irmplircd
- irctl für die Konfiguration: https://github.com/olebowle/irctl

Software Windows

- GUI Konfig-Tool stm32IRconfig_gui (wie Linux)
- Kommandozeilenprogramm stm32IRconfig (wie Linux)
- stm32IRalarm zum Alarmzeit setzen und auslesen per Skript
- MediaPortal Plugin: https://github.com/pikim/HIDIRT-host/tree/master/hidirt.MePo



Download

https://github.com/j1rie/IRMP_STM32
Die Quellen sind erst nach dem Durchlauf des prepare Skripts vollständig, das Skript lädt ST- und IRMP-Quellen herunter, packt sie aus und patcht sie.
Das hat den Vorteil, dass die Patche unter GPL gestellt werden können, ohne den Original Lizenzen in die Quere zu kommen.
Man sieht auch besser, was geändert wurde.

Pinbelegung

IR Sendediode - PB6
Logging - PB10 = Tx, verbinden mit Rx vom USB-seriell-TTL
F103:
IR TSOP - PB11
Einschalter Motherboard - PB14 und über USB
Toggle LED - PB13
Wakeup reset - PB12
USB disconnect - PB15
F105:
IR TSOP - PC6
Einschalter Motherboard - PB7 und über USB
Toggle LED - PB12
Wakeup reset - PB8

Firmware kompilieren

Linux: mit arm-none-eabi-gcc
Windows: mit Coocox CoIDE (benutzt auch arm-none-eabi-gcc)

http://www.olliw.eu/2013/stm32-this-and-that/#coocoxinstall

Firmware flashen

Firmware Flashen USB-seriell-TTL (STM32F103):
Ground und 5V bzw. 3,3V verbinden, Rx an Tx , Tx an Rx
BOOT0 Jumper auf 3,3V (Vorsicht, auf manchen Boards ist BOOT0 fälschlich mit BOOT1 beschriftet, Position siehe Fotos), Reset
./stm32flash -v -w /Pfad/IR.bin /dev/ttyUSB0
BOOT0 wieder auf Ground, Reset, USB neu anschliessen (oder USB Jumper ab, an)

Firmware Flashen über USB (STM32F105):
BOOT0 Jumper auf 3,3V, Reset, USB neu anschliessen (oder USB kurz auf low)
./dfu-util -a 0 -s 0x8000000 -D /Pfad/IR.bin
BOOT0 wieder auf Ground, Reset, USB neu anschliessen (oder USB kurz auf low)

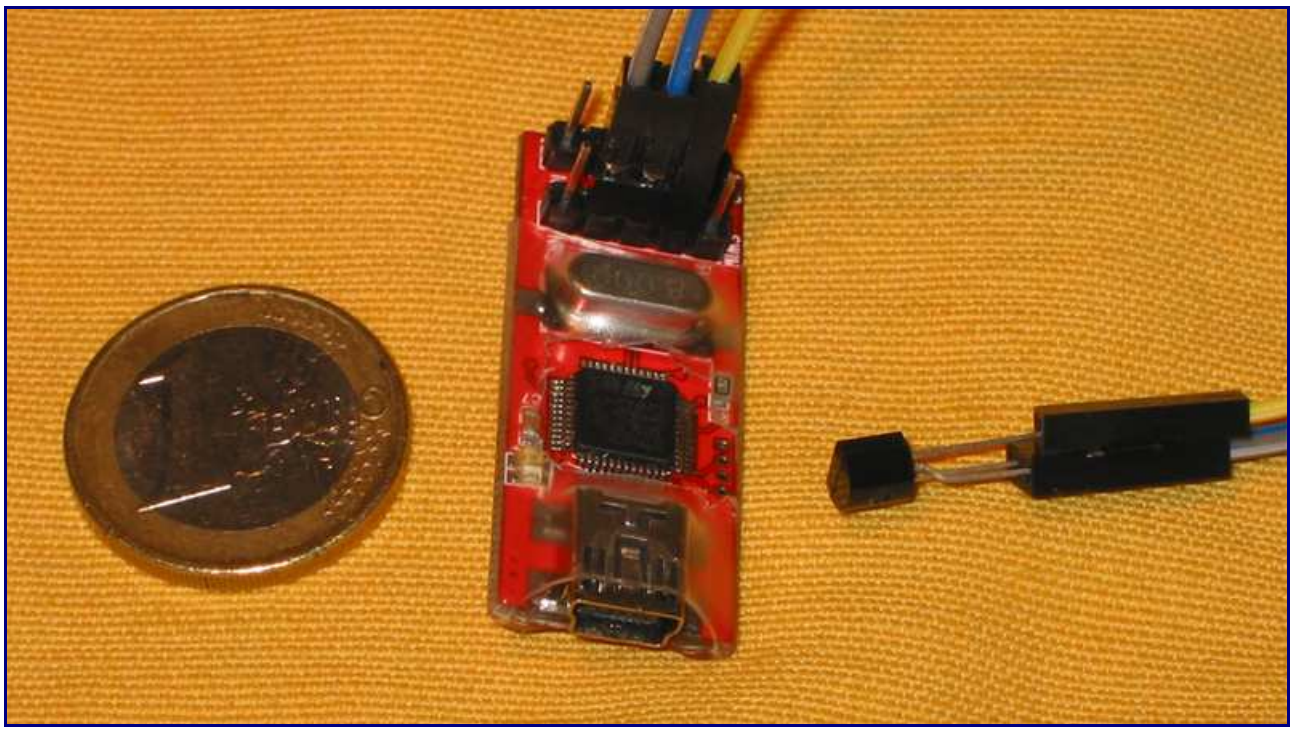
Firmware Flashen mit ST-Link:
Programmer SWDIO -> Ziel PA13/TMS , Programmer SWCLK -> Ziel PA14/TCK , Gnd - Gnd, 3,3V – 3,3V

andere STM32 Mikrocontroller

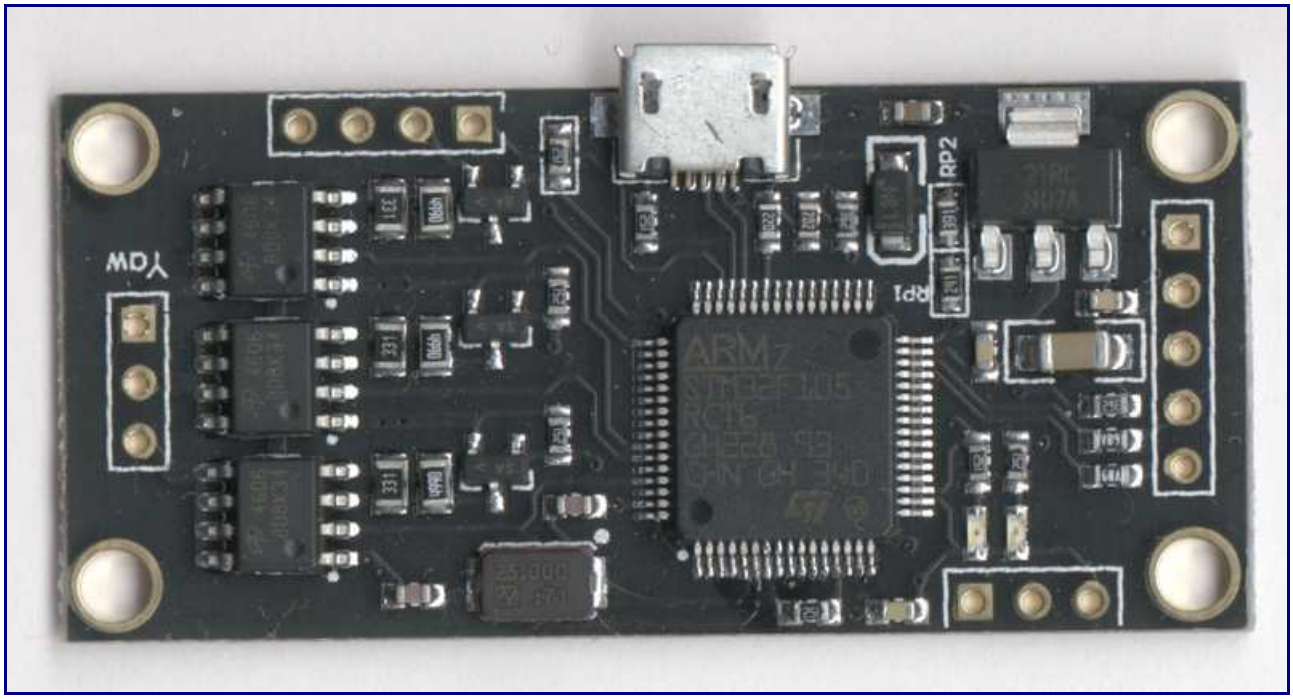
- die STM32F103 Firmware ist einfach anpassbar an STM32L1xx und STM32F3xx, da dieselbe USB Library genutzt wird
- die STM32F105 Firmware ist einfach anpassbar an STM32F107, STM32F2xx und STM32F4xx, da dieselbe USB Library genutzt wird

Fotos

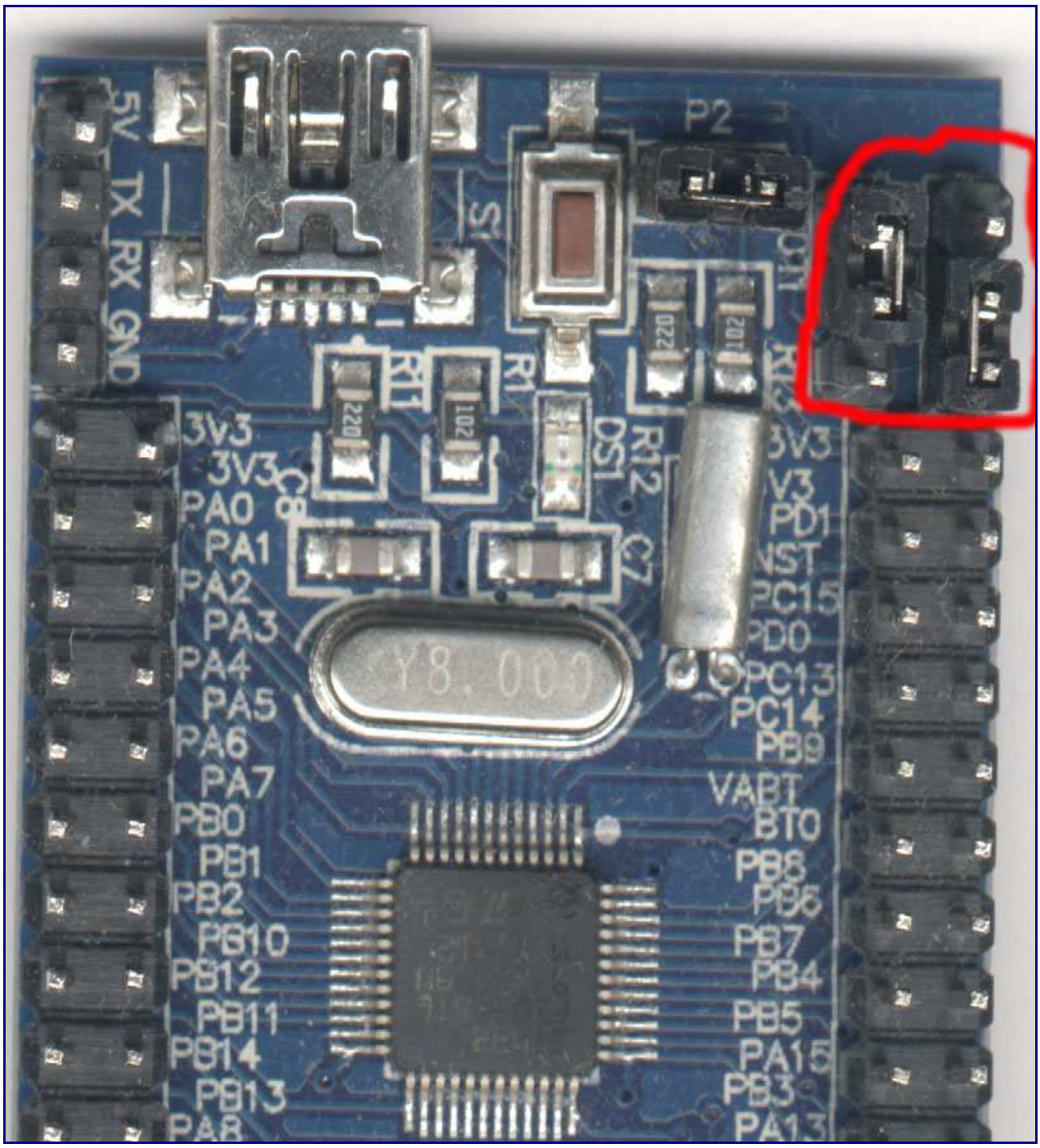
Auf dem STM32F103C8T6 Entwickler Board sind BOOT0 und BOOT1 vertauscht beschriftet. Auf dem roten ST-Link Emulator ist das Plastik etwas aufgeschnitten. Auf dem blauen ST-Link Emulator sind CLK und TCK sowie DIO und TMS vertauscht.



Am Roten eine TSOP „Attrappe“ und 1 € zum Größenvergleich.
An den St-Links muss eventuell der PullUp Widerstand (Roter) bzw. PullDown Widerstand (Blauer) an SWIM entfernt werden. Oder man wählt eine Funktion, die dadurch nicht gestört wird.



Mit diesem STM32F105 Board, das ursprünglich für die Kamera-Aufhängungs-Ansteuerung gedacht war, fing die Entwicklung an.



Jumper Position für Flashen über Tx/Rx (auch wenn die Beschriftung vertauscht ist)

TODO

Einbinden in weitere Windows HTPC Anwendungen.

Danke an

Frank Meyer für IRMP und IRSND. [1]
Uwe Becker für das Anpassen von IRMP und IRSND an den STM32F4xx
und seinen USB-HID für den STM32F4xx. [2]
Seine Arbeit war hilfreich, um mit dem STM32F105 anfangen zu können.
Andrew Kambaroff für seinen USB-HID für den STM32F103. [3]

Seine Arbeit war hilfreich, um mit dem STM32F103 anfangen zu können.
Ole Ernst für Code Review, Linux Makefile und Linux Download-Auspack-Patch-Skript, viel bessere Makro Implementierung und neues Protokoll. [4]

[1] https://www.mikrocontroller.net/articles/IRMP
[2] http://mikrocontroller.bplaced.net/wordpress/?page_id=744
[3] http://sysmagazine.com/posts/208026/
[4] https://github.com/olebowle

Diskussion

Meinungen, Verbesserungsvorschläge, Kritik und Ähnliches kann im http://www.vdr-portal.de/board18-vdr-hardware/board13-fernbedienungen/123572-irmp-auf-stm32-ein-usb-ir-empf%C3%A4nger-sender-einschalter-mit-wakeup-timer/ geäussert werden.
https://www.mikrocontroller.net/topic/347290 scheint eingeschlafen zu sein.

Bauanleitung

https://www.mikrocontroller.net/articles/IRMP_auf_STM32_-_Bauanleitung

Viel Spaß mit IRMP auf STM32!