

IRMP on STM32 - a USB IR receiver/sender/powerswitch with wakeup-timer

A remote control receiver with many functions based on cheap hardware.

Copyright (C) 2014-2015 Jörg Riechardt

Contents

- [16 Introduction](#)
- [17 Functions](#)
- [18 Software Linux](#)
- [19 Software Windows](#)
- [20 Download](#)
- [21 Pin-Out](#)
- [22 Firmware Compilation](#)
- [23 Firmware Flashing](#)
- [24 Other STM32 Microcontrollers](#)
- [25 Images](#)
- [26 TODO](#)
- [27 Thanks to](#)
- [28 Discussion](#)
- [29 Construction manual](#)

Introduction

For 4 - 8 € you can buy on ebay a ST-Link emulator or a STM32F103 development board, manufactured in China usually. Onto the STM32F103 mikrocontroller on it an open source firmware with many functions will be flashed.

Functions

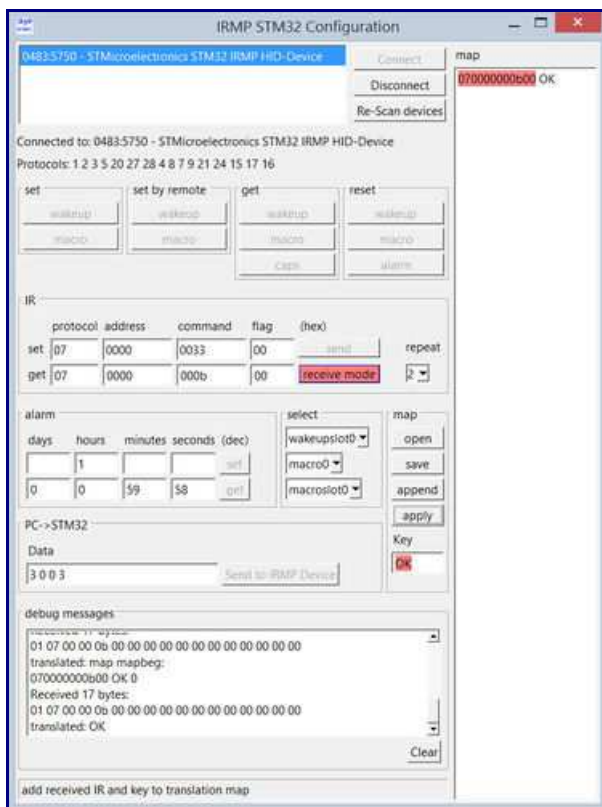
- connection via USB
- registers as HID device, no driver needed, appears as /dev/hidraw or "Custom HID device"
- transfers the data via USB2 in hardware with fullspeed
- IR receiver (ca. 40 protocols decoded in hardware)
- switch PC on via remote control from S3 (STR) and S4 (STD) via USB or from S3, S4 and S5 (SoftOff) via motherboard-switch (+5V needed, on USB or from PSU or motherboard)
- switch PC on via build in timer from S3 (STR) and S4 (STD) via USB or from S3, S4 and S5 (SoftOff) via motherboard-switch (+5V needed, on USB or from PSU or motherboard)
- IR sender (ca. 40 protocols)
- the configuration is stored in the emulated eeprom
- within eeprom size any number of macros with any depth and any number of wakeups: $\text{MACRO_SLOTS} \times (\text{MACRO_DEPTH} + 1) + \text{WAKE_SLOTS} < 85$ for F103, < 170 for F105
- bootloader for comfortable firmware update

Software Linux

- GUI config-tool stm32IRconfig_gui for configuration: set, get and reset wakeup codes, macros, alarm time, send IR and show the received IR code. It is possible to program wakeups and macros via remote control. Create the translation table with remote control and mouse, as well test and edit. Comprehensive debug messages.
- command line program stm32IRconfig for configuration: set, get and reset wakeup code, macros, alarm time, send IR and show the received IR code. It is possible to program wakeups and macros via remote control.
- stm32IRalarm for setting and reading alarm time via script
- irimplrcd is a daemon, running in background as an independent lirc server and passing the IR codes/events on to the application <https://github.com/realglotzi/irimplrcd>
- irctl for configuration: <https://github.com/olebowle/irctl>

Software Windows

- GUI config-tool stm32IRconfig_gui (same as linux)
- command line programm stm32IRconfig (same as linux)
- stm32IRalarm for setting and reading alarm time via script
- MediaPortal plugin: <https://github.com/pikim/HIDIRT-host/tree/master/hidirt.MePo>



Download

https://github.com/jlrie/IRMP_STM32

The sources are complete only after running the prepare script, the script downloads the ST- and IRMP-sources, extracts and patches them.

This makes it possible to put the patches under GPL, without interfering with the original licences. It is easier to see, what has been changed, too.

Pin-Out

IR send diode - PB6

logging - PB10 = Tx, connect to Rx of USB-serial-TTL

F103:

IR TSOP – PB11

power on switch motherboard - PB14 and via USB

toggle LED - PB13

Wakeup reset - PB12

USB disconnect - PB15

F105:

IR TSOP - PC6

power on switch motherboard - PB7 and via USB

toggle LED - PB12

Wakeup reset - PB8

Firmware Compilation

Linux: arm-none-eabi-gcc

Windows: Coocox CoIDE (uses arm-none-eabi-gcc too)

<http://www.olliv.eu/2013/stm32-this-and-that/#coocoxinstall>

Firmware Flashing

flash firmware with USB-serial-TTL (STM32F103):

connect ground and 5V resp. 3,3V, Rx - Tx , Tx - Rx

BOOT0 jumper on 3,3V, (caution, on some boards BOOT0 is labeled wrongly as BOOT1, see images for position), reset

`./stm32flash -v -w /path/IR.bin /dev/ttyUSB0`

BOOT0 back to ground, reset, reconnect USB (or USB jumper off, on)

flash firmware via USB (STM32F105):

BOOT0 jumper on 3,3V, reset, reconnect USB (or USB low shortly)

`./dfu-util -a 0 -s 0x8000000 -D /path/IR.bin`

BOOT0 back to ground, reset, reconnect USB (or USB low shortly)

flash firmware via ST-Link:

SWDIO -> PA13, SWCLK -> PA14, Gnd - Gnd, 3,3V - 3,3V

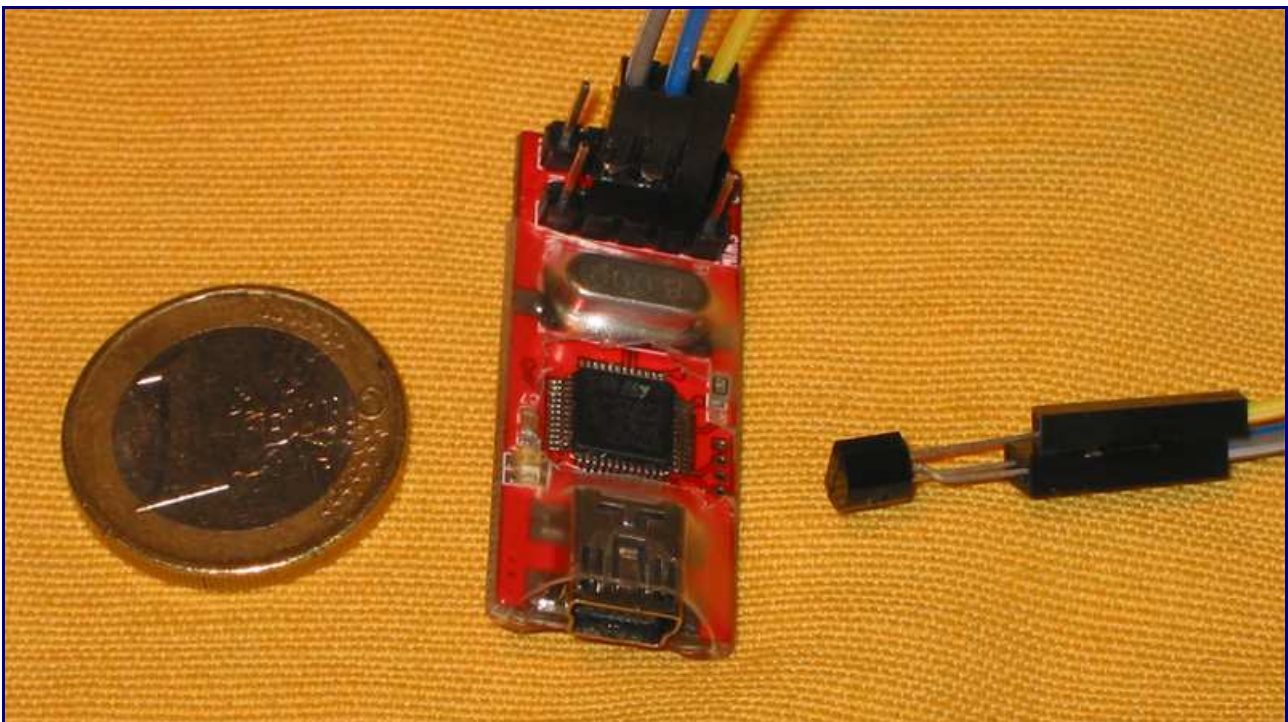
Other STM32 Microcontrollers

- the STM32F105 firmware is easily customizable to STM32F107, STM32F2xx and STM32F4xx, as the same USB library is used
- the STM32F103 firmware is easily customizable to STM32L1xx and STM32F3xx, as the same USB library is used

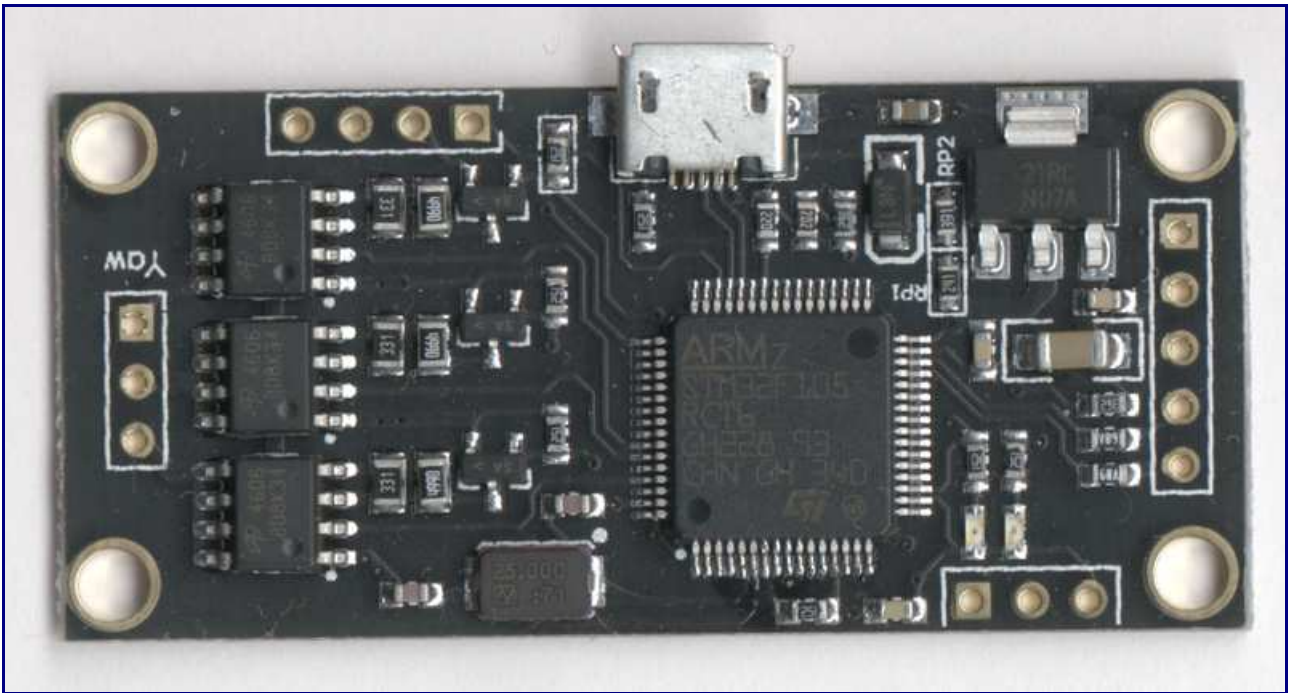
Images



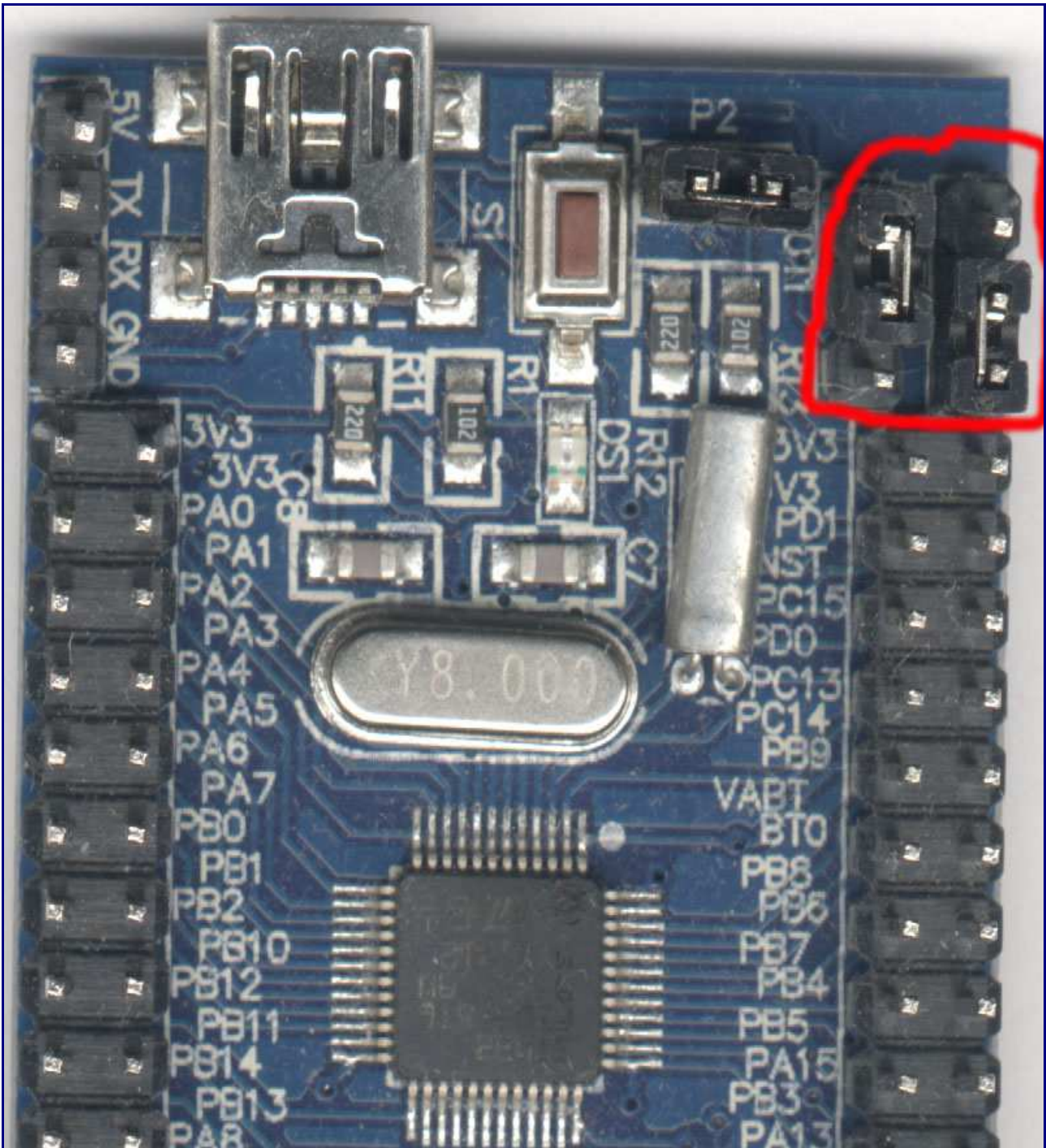
On the STM32F103C8T6 developer board BOOT0 and BOOT1 labels are exchanged. On the red ST-Link emulator the plastic is cut open some. On the blue ST-Link emulator CLK and TCK are exchanged and DIO und TMS as well.



Attached to the red a TSOP „dummy“ and 1 € for sizecomparison.
On the St-Links it might be necessary to remove the pullup resistor (red) or the pulldown resistor (blue) on SWIM. Or use a function, which is not disturbed by them.



With this STM32F105 board, intended as camera gimbal controller, the development started.



Jumper position for flashing via Tx/Rx (also if labeled incorrectly)

TODO

Integration into further Windows HTPC applications.

Thanks to

Frank Meyer for IRMP und IRSND. [1]

Uwe Becker for adapting IRMP and IRSND to the STM32F4xx and his USB-HID for the STM32F4xx. [2]

His work helped me in getting started with the STM32F105.

Andrew Kambaroff for his USB-HID for the STM32F103. [3]

His work helped me in getting started with the STM32F103.

Ole Ernst for code review, linux Makefile and linux download-extract-patch-script, much better macro implementation and new protocol. [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>

Discussion

Express opinions, suggestions for improvement, criticism and similar at <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/> .

<https://www.mikrocontroller.net/topic/347290> seems to be fallen asleep.

Construction manual

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

Much fun with IRMP on STM32!