# Embedded-Systems Hands-On

**Group 4**

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Sommersemester 2020
Sheet 1

## Task 1.1: Raspberry Pi

In the first task Raspbian OS should be installed on the Pi. The raspbian-ua-netinstaller was recommended for this process, but due to the problem that the pi could not detect the ethernet connection Raspbian OS Lite was installed with the provided package on the Raspbian Website.

## Task 1.2: OpenOCD

The OpenOCD Auto-Config file automatically detects whether a necessairy package is installed for specific subfeatures. This lead to the problem that if one did not have the necessairy prerequisites for a feature, the config file disabled the feature completely. This lead me to believe that OpenOCD was installed successfully even though the necessairy JTAG libs / configs and etc. were not installed. After getting an error message that the target configuration file stm32f0x.cfg could not be found and doing further research, the necessary prerequisites were installed and OpenOCD reinstalled. To find out the necessary Pin-IDs the following image was used:

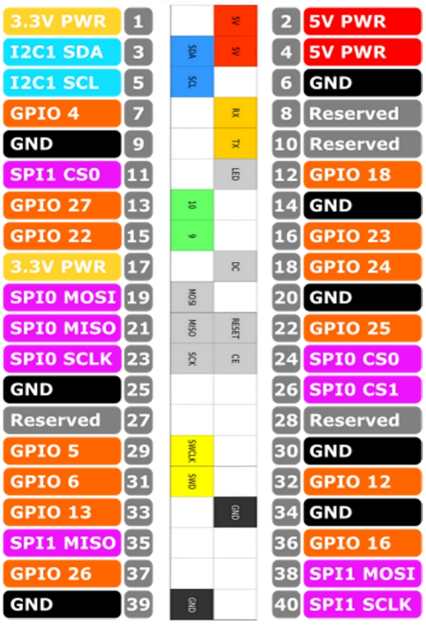
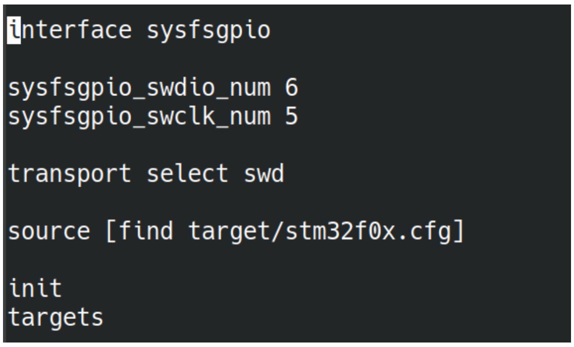


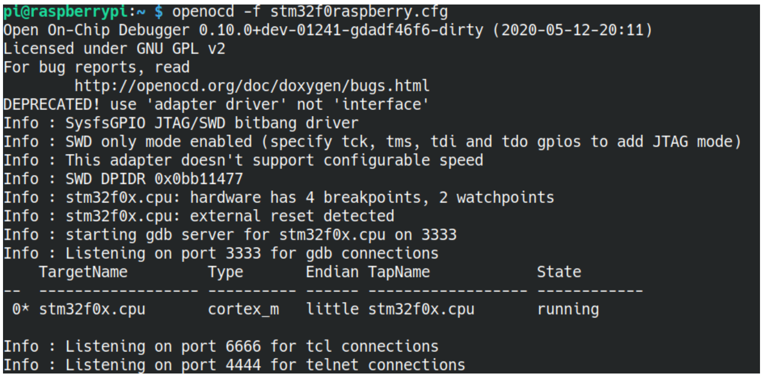
Figure 1: Pinlayout of ESHO-Board overlayed with Raspberry Pi Pin Header

SWDCLK is on GPIO-Pin 5 and SWDIO is on GPIO-Pin 6.



```
interface sysfsgpio

sysfsgpio_swdio_num 6
sysfsgpio_swclk_num 5

transport select swd

source [find target/stm32f0x.cfg]

init
targets
```

After launching OpenOCD with the given config file the following was output:

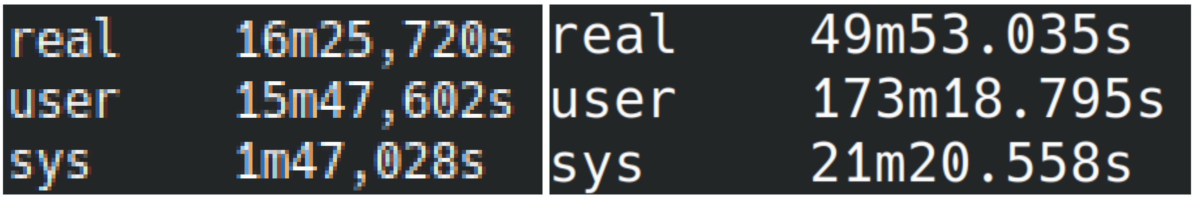


```
pi@raspberrypi:~ $ openocd -f stm32f0raspberry.cfg
Open On-Chip Debugger 0.10.0+dev-01241-gdadf46f6-dirty (2020-05-12-20:11)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
DEPRECATED! use 'adapter driver' not 'interface'
Info : SysfsGPIO JTAG/SWD bitbang driver
Info : SWD only mode enabled (specify tck, tms, tdi and tdo gpios to add JTAG mode)
Info : This adapter doesn't support configurable speed
Info : SWD DPIDR 0x0bb11477
Info : stm32f0x.cpu: hardware has 4 breakpoints, 2 watchpoints
Info : stm32f0x.cpu: external reset detected
Info : starting gdb server for stm32f0x.cpu on 3333
Info : Listening on port 3333 for gdb connections
    TargetName         Type       Endian TapName            State
--  ------------------ ---------- ------ ------------------ ------------
 0* stm32f0x.cpu       cortex_m   little stm32f0x.cpu       running

Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
```

Figure 2: Output after running OpenOCD with given Configuration File

OpenOCD has successfully identified our Cortex-M0 via GDB.

### Task 1.3: Compile a Linux Kernel



```
real    16m25,720s
user    15m47,602s
sys     1m47,028s
```

```
real    49m53.035s
user    173m18.795s
sys     21m20.558s
```

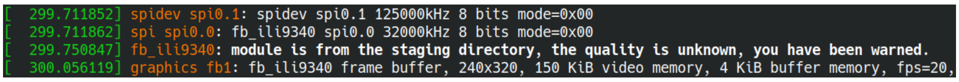
Figure 3: Compile time for x86_64(left) and ARM Cortex-A72(Pi 4)

Compile time for x86_64 Intel Core I5-8250u running Kubuntu 20.04: 16 minutes and 25 seconds(985s)

Compile time for ARM Cortex-A72 running Raspbian Lite: 49 minutes and 53 seconds(2993s)

The Pi took approximately **three** times longer to compile the kernel than the x86_64. Config files can be found in .config_CROSS_COMPILE and .config_PI respectively.

### Task 1.4: Loading Linux Modules



```
[  299.711852] spidev spi0.1: spidev spi0.1 125000kHz 8 bits mode=0x00
[  299.711862] spi spi0.0: fb_ili9340 spi0.0 32000kHz 8 bits mode=0x00
[  299.750847] fb_ili9340: module is from the staging directory, the quality is unknown, you have been warned.
[  300.056119] graphics fb1: fb_ili9340 frame buffer, 240x320, 150 KiB video memory, 4 KiB buffer memory, fps=20,
```

Figure 4: Dmesg output after loading tft_device driver

As depicted in figure 1 the display's SPI ports(MOSI, MISO, SCLK, CE) are connected to the SPI0 ports of the Raspberry Pi(physical Pins 19, 21, 23, 24). The RESET Pin is connected to GPIO 25, the DC Pin to GPIO 24 and LED Pin to GPIO 18 respectively. Given this and with regards to the display, the display driver has to be configured as follows: '*sudo modprobe tft_device name=adafruit22a gpios=led:18,dc:24,reset:25*'. The adafruit22a driver was chosen because the adafruit22a display uses the same ILI9340 display and thus the display on the extension board can be driven with the same driver. Unfortunately my display was broken and we had to verify the given parameters with a different board. The FPS archieved by the display is 20 shown in figure 4.

### Task 1.5: Compile FBTFT into the Kernel

The config file can be found in .config2 and the dts file can be found in ili93401.dts.