Embedded-Systems Hands-On

Gruppe 4

TECHNISCHE UNIVERSITÄT DARMSTADT

Sommersemester 2020
Sheet 2

Task 2.1: Serial Wire Debug

All codes are placed "**2_1**" subfolder. **pigpio** is required to compile the code. This can be installed with "**apt**" on the raspberry pi or by following these **instructions**. To compile the code the makefile can be executed. To run the code execute "**T2**" file with "**sudo**".

Task 2.2: GDB server

The objective of this task was to connect to the gdb server initiated by the OpenOCD with a GDB client running on a remote device. Problems arose when gdb running on the Raspberry Pi was able to connect to the server while the gdb on the laptop was not able to connect. The cause of this problem was that OpenOCD only listens to the loopback IP address 127.0.0.1 by default. The solution was simple. According to the documentation of OpenOCD the configuration file can be used to bind the initiated servers to an arbitary IP-Address. Launching OpenOCD with the modified configuration file allowed the gdb on the laptop to attach to the cortex-m0 gdb session.

```
interface sysfsgpio

sysfsgpio_swdio_num 6
sysfsgpio_swclk_num 5

transport select swd

source [find target/stm32f0x.cfg]

bindto 192.168.0.183

init
targets
```

Figure 1: Modified Configuration File to bind gdb server to a no loopback IP address

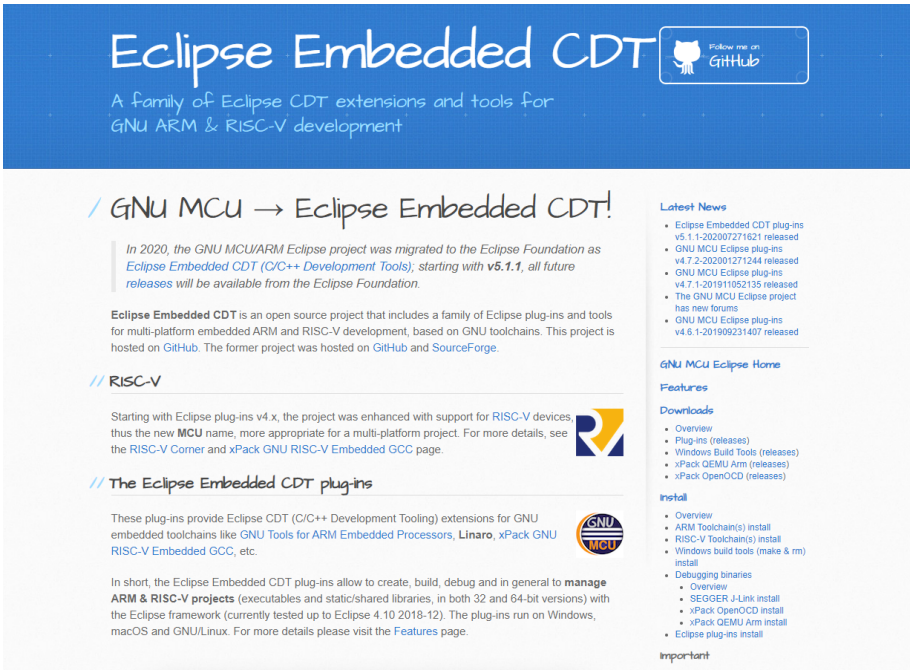
```
(gdb) info all-registers
r0             0x0                0
r1             0x1                1
r2             0x0                0
r3             0x2000132c         536875820
r4             0x8002341          134226753
r5             0x0                0
r6             0x55555555         1431655765
r7             0x55555555         1431655765
r8             0x55555555         1431655765
r9             0x55555555         1431655765
r10            0x55555555         1431655765
r11            0x55555555         1431655765
r12            0xe000e100         -536813312
sp             0x20001230         0x20001230
lr             0x80001c3          134218179
pc             0x8002340          0x8002340
xPSR           0x41000000         1090519040
msp            0x20000400         0x20000400
psp            0x20001230         0x20001230
primask        0x0                0
basepri        0x0                0
faultmask      0x0                0
control        0x2                2
```

Figure 2: GDB output after info all-registers command

Task 2.3: Debugging IDE

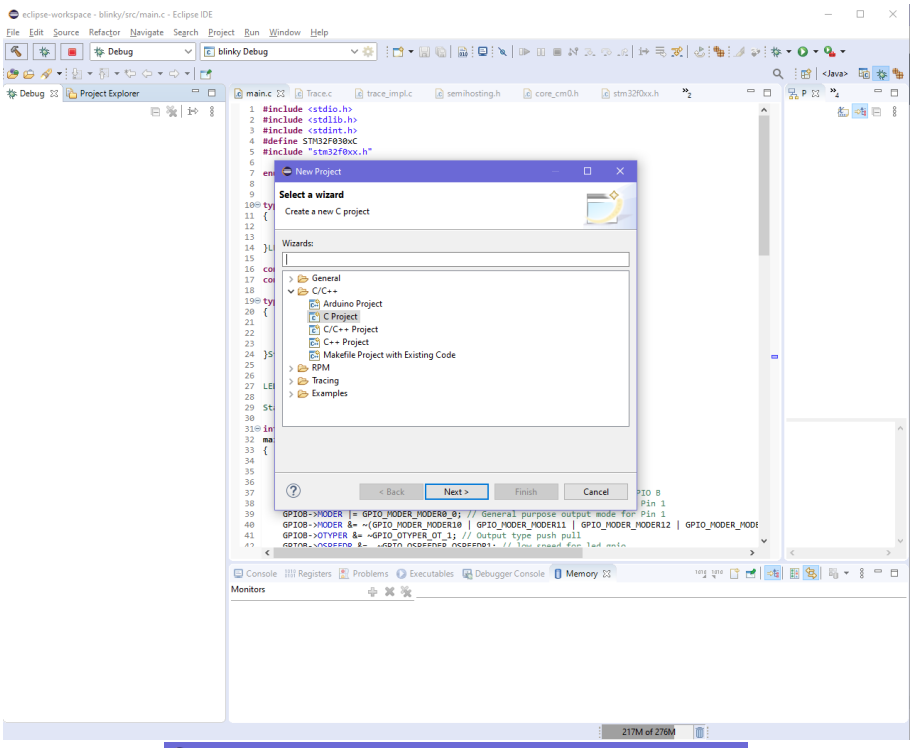
For the task we chose to use the MCU Eclipse.

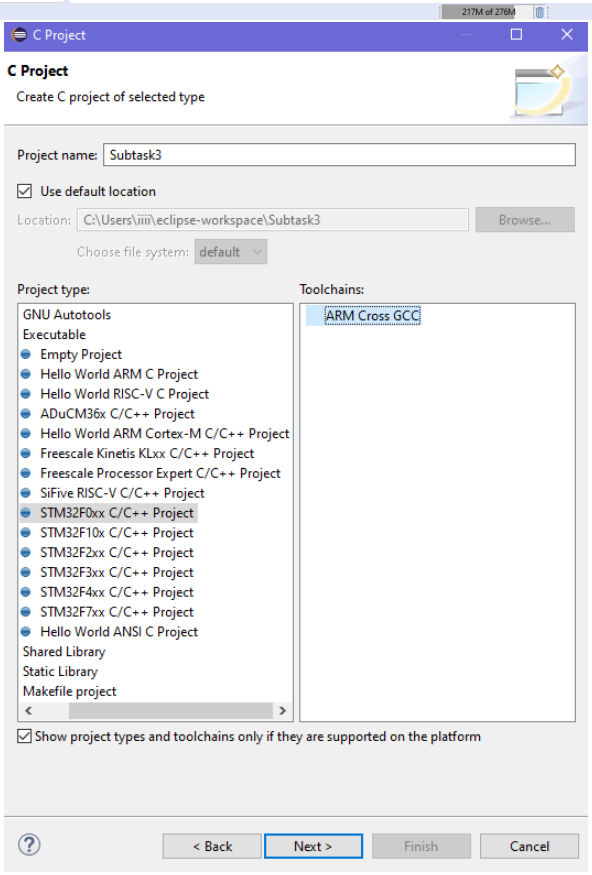
1.



We followed the instructions on the official **MCU Eclipse** website to install Eclipse MCU.

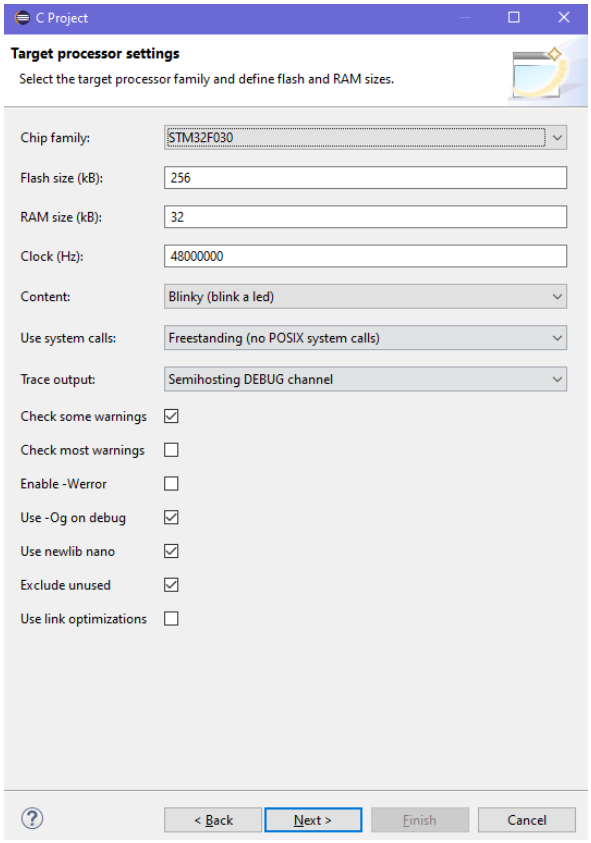
2.





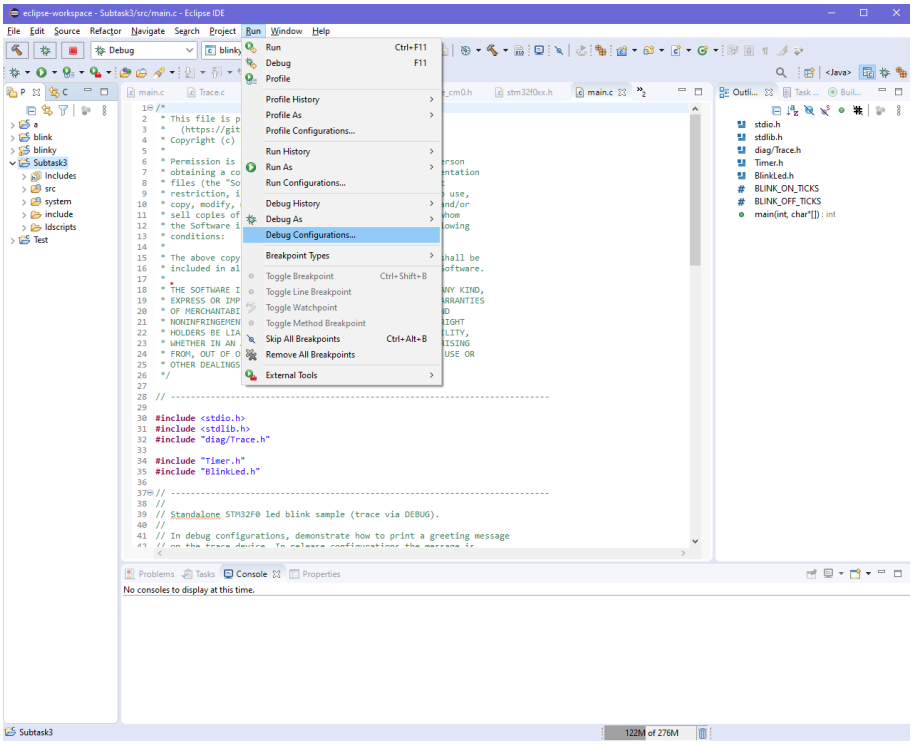
After installing and opening Eclipse, we created a new STM32F0xx C/C++ project.

3.

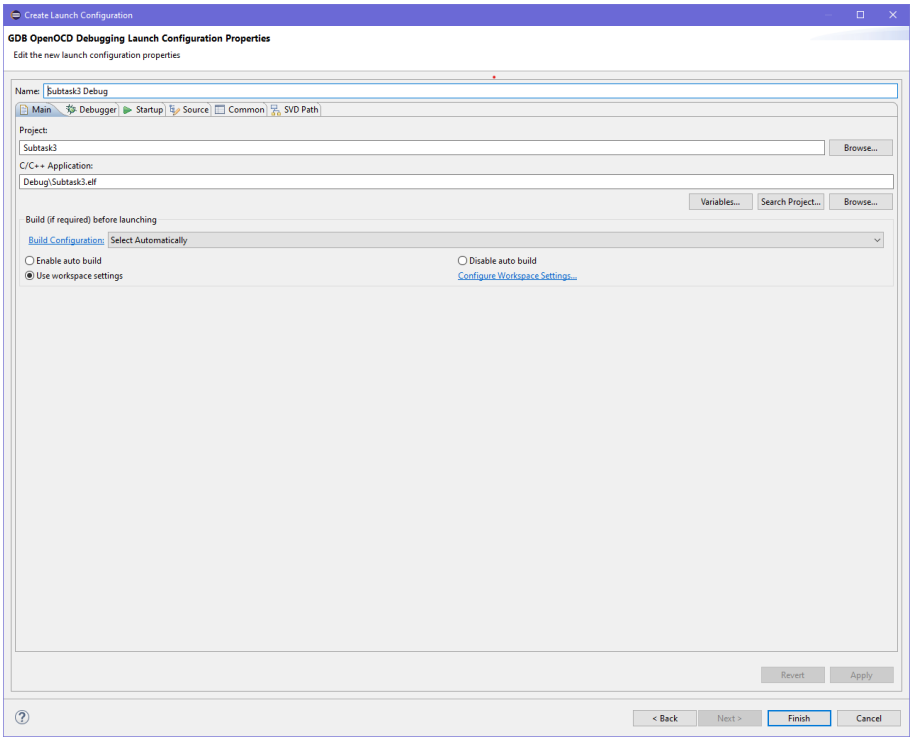


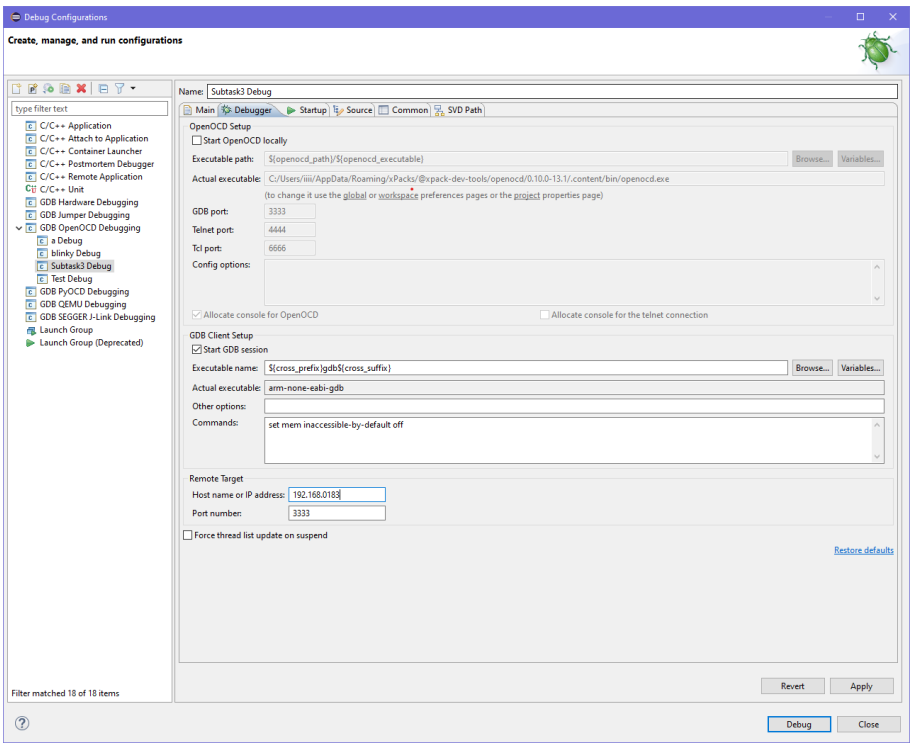
We then had to set the specification according to our Cortex-M0(STM32F030CC) which can be extracted from the datasheets.

4.

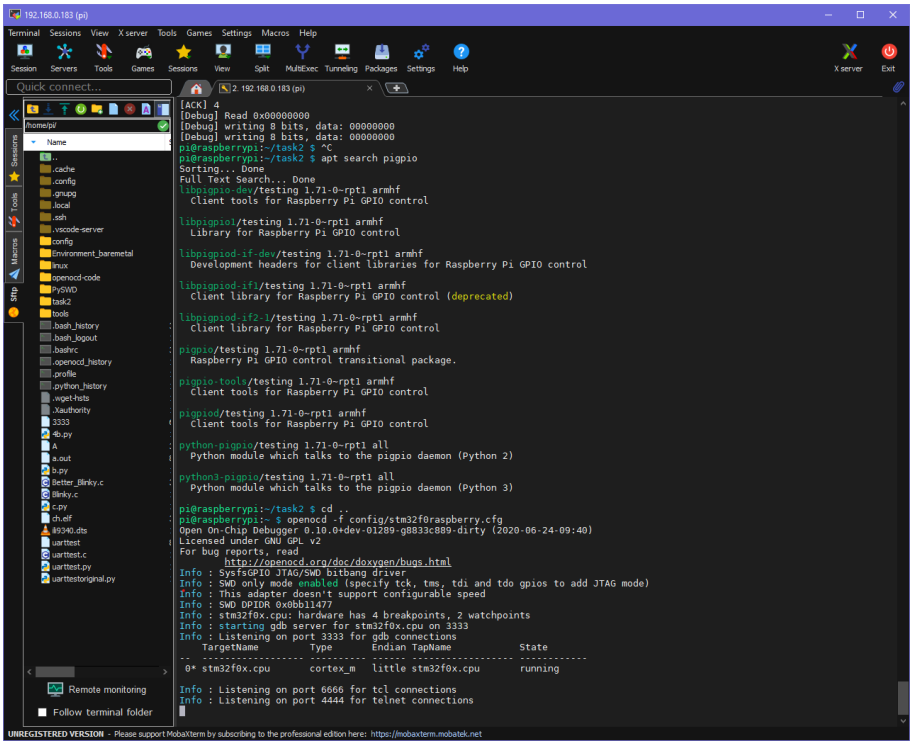


We then had to setup our new debug configurations like the following.



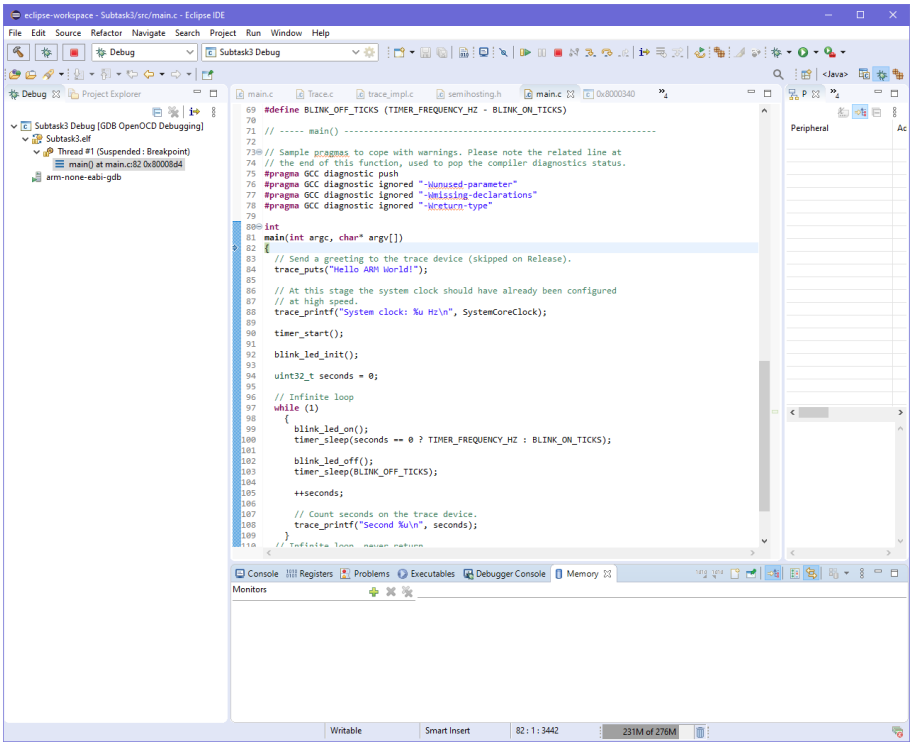

Debug Configurations
Create, manage, and run configurations
type filter text
C/C++ Application
C/C++ Attach to Application
C/C++ Container Launcher
C/C++ Postmortem Debugger
C/C++ Remote Application
C/C++ Unit
GDB Hardware Debugging
GDB Jumper Debugging
GDB OpenOCD Debugging
a Debug
blinky Debug
Subtask3 Debug
Test Debug
GDB PyOCD Debugging
GDB QEMU Debugging
GDB SEGGER J-Link Debugging
Launch Group
Launch Group (Deprecated)
Filter matched 18 of 18 items
Name: Subtask3 Debug
Main
Debugger
Startup
Source
Common
SVD Path
OpenOCD Setup
Start OpenOCD locally
Executable path:
${openocd_path}/${openocd_executable}
Browse...
Variables...
Actual executable:
C:/Users/iiii/AppData/Roaming/xPacks/@xpack-dev-tools/openocd/0.10.0-13.1/.content/bin/openocd.exe
(to change it use the global or workspace preferences pages or the project properties page)
GDB port:
3333
Telnet port:
4444
Tcl port:
6666
Config options:
Allocate console for OpenOCD
Allocate console for the telnet connection
GDB Client Setup
Start GDB session
Executable name:
${cross_prefix}gdb${cross_suffix}
Browse...
Variables...
Actual executable:
arm-none-eabi-gdb
Other options:
Commands:
set mem inaccessible-by-default off
Remote Target
Host name or IP address:
192.168.0183
Port number:
3333
Force thread list update on suspend
Restore defaults
Revert
Apply
Debug
Close

5.



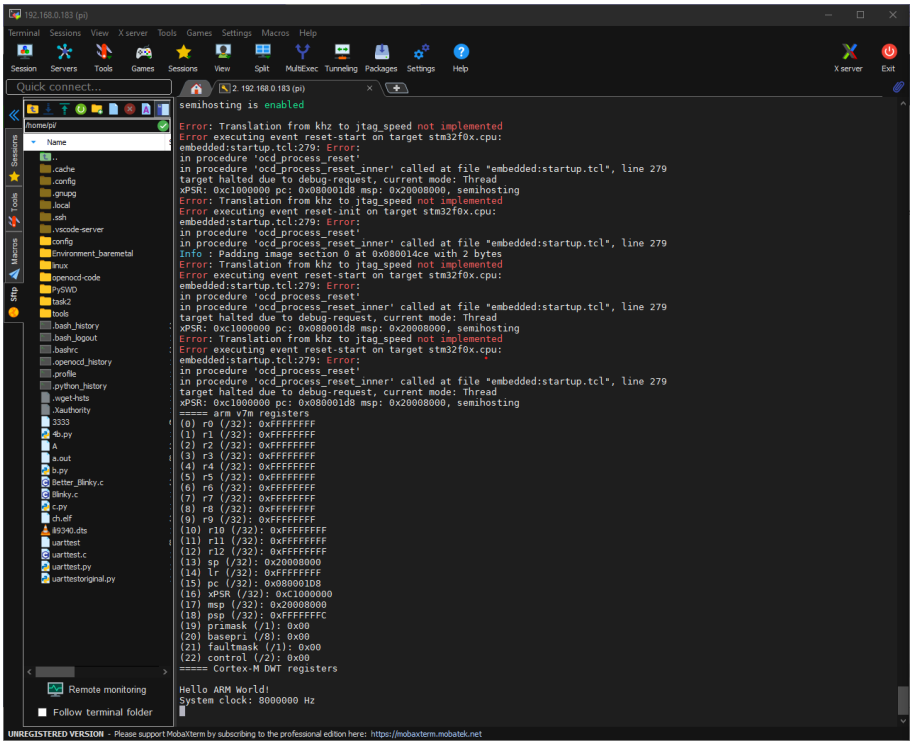
We then launched OpenOCD on the Raspberry Pi.

6.



Once we launched the debug configuration we were shown this debug perspective.

7.



OpenOCD output while stepping through the Blinky example project.