

MAX 1000 Tutorial Expanding a Nios II System


Exercise Manual Part 2

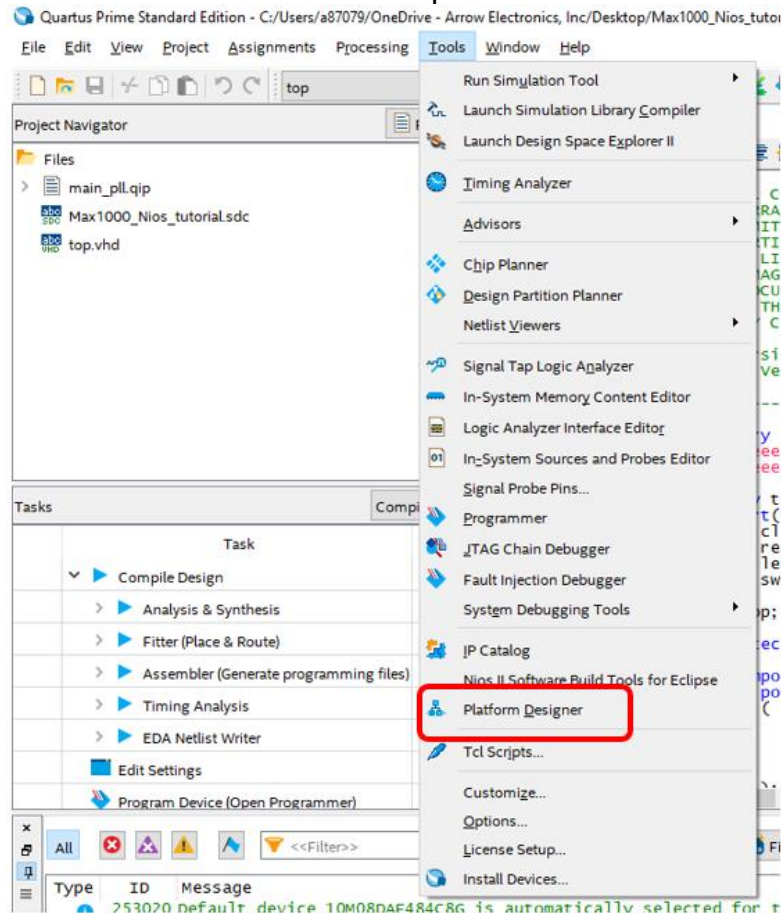
for MAX1000 Board

Quartus Prime 18.1 / Platform Designer

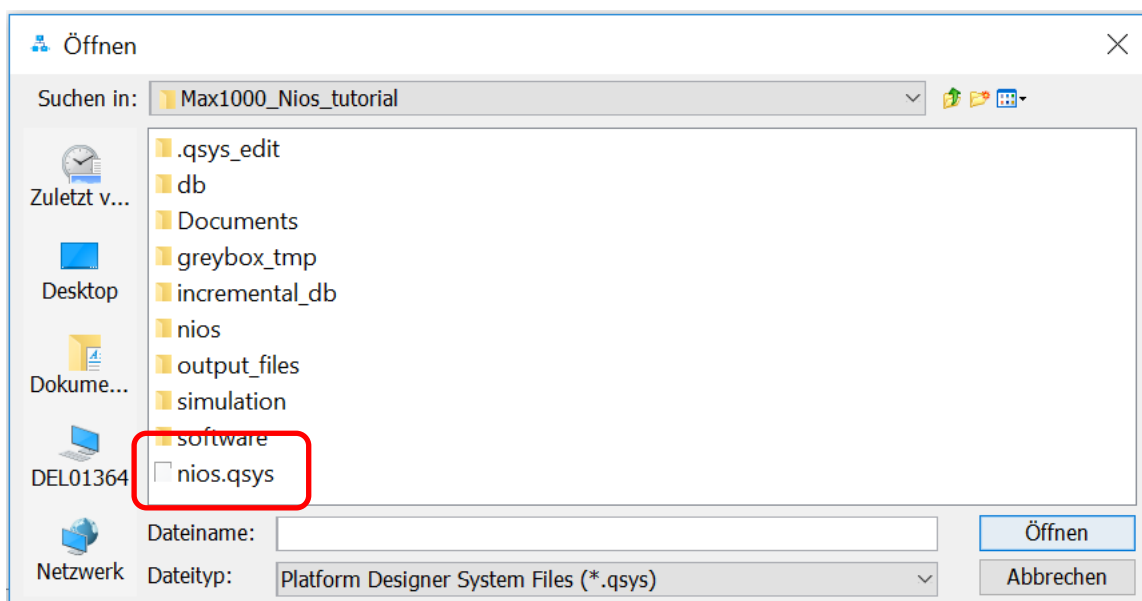
1. Now we are expanding the NIOSII system

a. Start Platform Designer from “Tools => Platform Designer” or by clicking the

icon  in the line below the top menu.



2. Open the NIOS.qsys from the first lab:

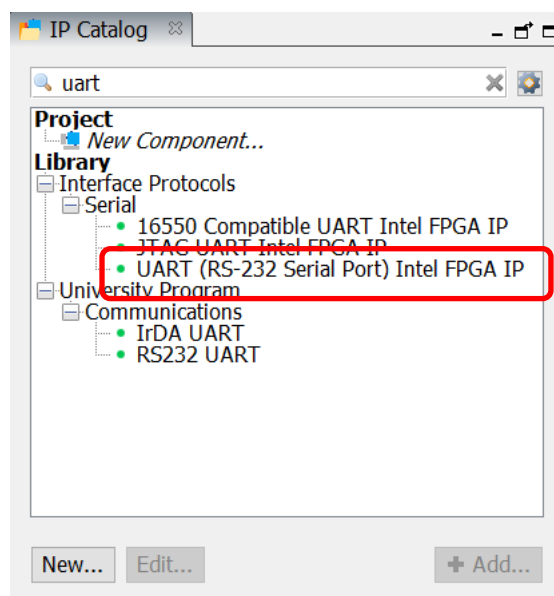


Now You see all components from the first lab:

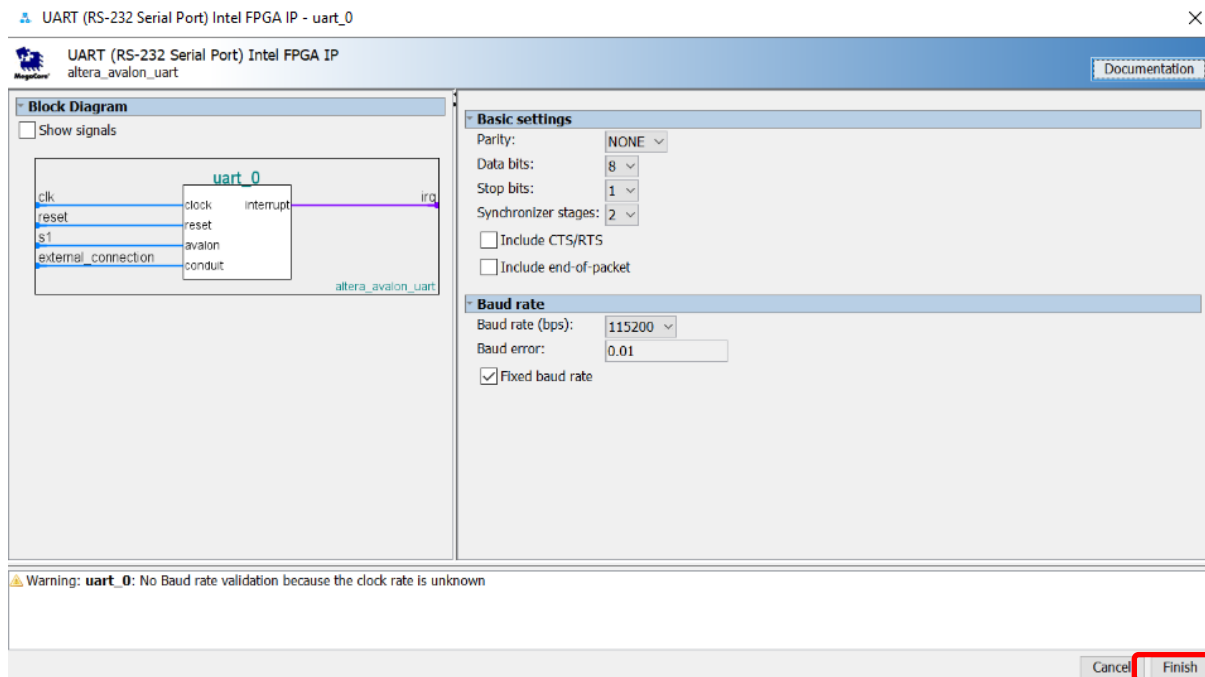
Use	Connections	Name	Description	Export	Clock	Base	End	I...	Tags
<input checked="" type="checkbox"/>		clk	Clock Source						
		clk_in	Clock Input	clk	exported				
		clk_in_reset	Reset Input	reset					
		clk	Clock Output	Double-click to	clk				
		clk_reset	Reset Output	Double-click to					
<input checked="" type="checkbox"/>		nios	Nios II Processor						
		clk	Clock Input	Double-click to	clk				
		reset	Reset Input	Double-click to	[clk]				
		data_master	Avalon Memory Mapped ...	Double-click to	[clk]				
		instruction_m...	Avalon Memory Mapped ...	Double-click to	[clk]				
		irq	Interrupt Receiver	Double-click to	[clk]			IRQ 0	IRQ 31
		debug_reset_r...	Reset Output	Double-click to	[clk]				
		debug_mem...	Avalon Memory Mapped ...	Double-click to	[clk]	# 0x0011 0800	0x0011_0fff		
		custom_instru...	Custom Instruction Master	Double-click to					
<input checked="" type="checkbox"/>		ram	On-Chip Memory (RAM o...						
		clk1	Clock Input	Double-click to	clk				
		s1	Avalon Memory Mapped ...	Double-click to	[clk1]	# 0x0010 8000	0x0010_ffff		
		reset1	Reset Input	Double-click to	[clk1]				
<input checked="" type="checkbox"/>		flash	On-Chip Flash Intel FPGA...						
		clk	Clock Input	Double-click to	clk				
		nreset	Reset Input	Double-click to	[clk]				
		data	Avalon Memory Mapped ...	Double-click to	[clk]	# 0x0008 0000	0x000c_dfff		
		csr	Avalon Memory Mapped ...	Double-click to	[clk]	# 0x0011 1030	0x0011_1037		
<input checked="" type="checkbox"/>		sys_id	System ID Peripheral Inte...						
		clk	Clock Input	Double-click to	clk				
		reset	Reset Input	Double-click to	[clk]				
		control_slave	Avalon Memory Mapped ...	Double-click to	[clk]	# 0x0011 1028	0x0011_102f		
<input checked="" type="checkbox"/>		led_pio	PIO (Parallel I/O) Intel F...						
		clk	Clock Input	Double-click to	clk				
		reset	Reset Input	Double-click to	[clk]				
		s1	Avalon Memory Mapped ...	Double-click to	[clk]	# 0x0011 1010	0x0011_101f		
		external_conn...	Conduit	Double-click to					
<input checked="" type="checkbox"/>		key_pio	PIO (Parallel I/O) Intel F...						
		clk	Clock Input	Double-click to	clk				
		reset	Reset Input	Double-click to	[clk]				
		s1	Avalon Memory Mapped ...	Double-click to	[clk]	# 0x0011 1000	0x0011_100f		
		external_conn...	Conduit	Double-click to					
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART Intel FPGA IP						
		clk	Clock Input	Double-click to	clk				
		reset	Reset Input	Double-click to	[clk]				
		avalon_jtag_sl...	Avalon Memory Mapped ...	Double-click to	[clk]	# 0x0011 1020	0x0011_1027		
		irq	Interrupt Sender	Double-click to	[clk]				

3. Adding a RS232 UART

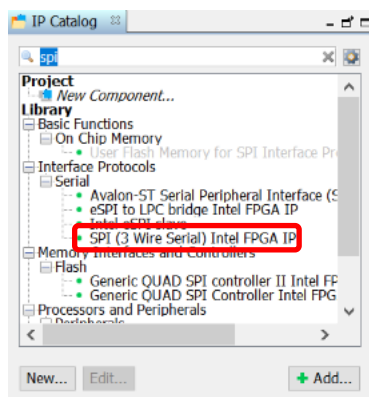
- a. In the IP Catalog search for “UART” and double click “UART (RS-232 Serial Port) Intel FPGA IP”



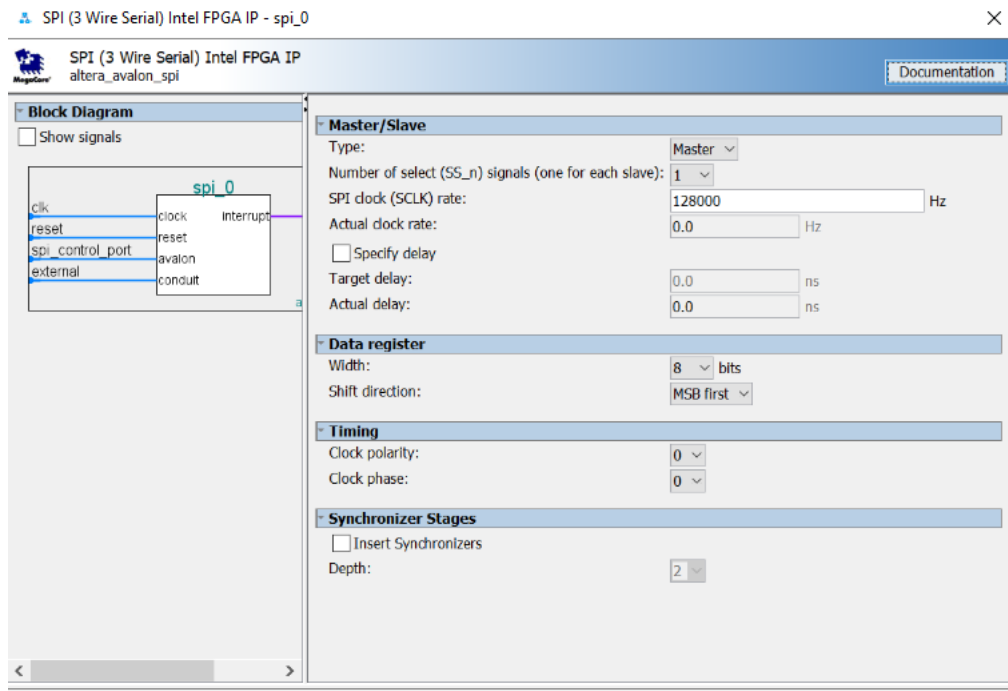
- b. There's no need to change anything. Here You can find all the settings for Your Terminal-program. Just click finish



4. Adding the SPI-Interface to connect the Accelerometer:
 - a. Search for “SPI” and double click “SPI (3 Wire Serial) Intel FPGA IP”



- b. There's no need to change anything. Just click finish

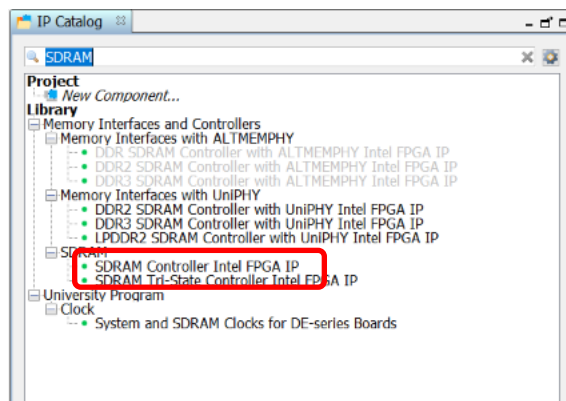


C.

Cancel Finish

5. Adding the SDRAM

- a. Search for “SDRAM” and double click “SDRAM Controller Intel FPGA IP”



- b. Set “Memory Profile” as shown:

SDRAM Controller Intel FPGA IP
altera_avalon_new_sdram_controller

Memory Profile **Timing**

Data Width
Bits: 16

Architecture
Chip select: 1
Banks: 4

Address Width
Row: 12
Column: 8

Generic Memory model (simulation only)
☐ Include a functional memory model in the system testbench

Memory Size = 8 MBytes
4194304 x 16
64 MBits

c.

d. And “Timing” to:

System: nios **Path:** sdram

SDRAM Controller Intel FPGA IP
altera_avalon_new_sdram_controller

Memory Profile **Timing**

CAS latency cycles::
☐ 1
☒ 2
☐ 3

Initialization refresh cycles: 2

Issue one refresh command every: 15.625 us

Delay after powerup, before initialization: 200.0 us

Duration of refresh command (t_rfc): 70.0 ns

Duration of precharge command (t_rp): 20.0 ns

ACTIVE to READ or WRITE delay (t_rcd): 20.0 ns

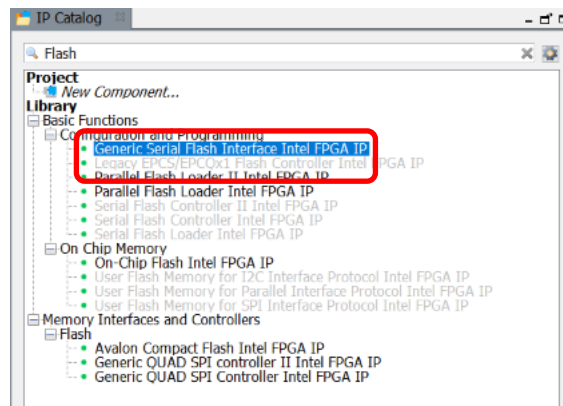
Access time (t_ac): 6.0 ns

Write recovery time (t_wr, no auto precharge): 14.0 ns

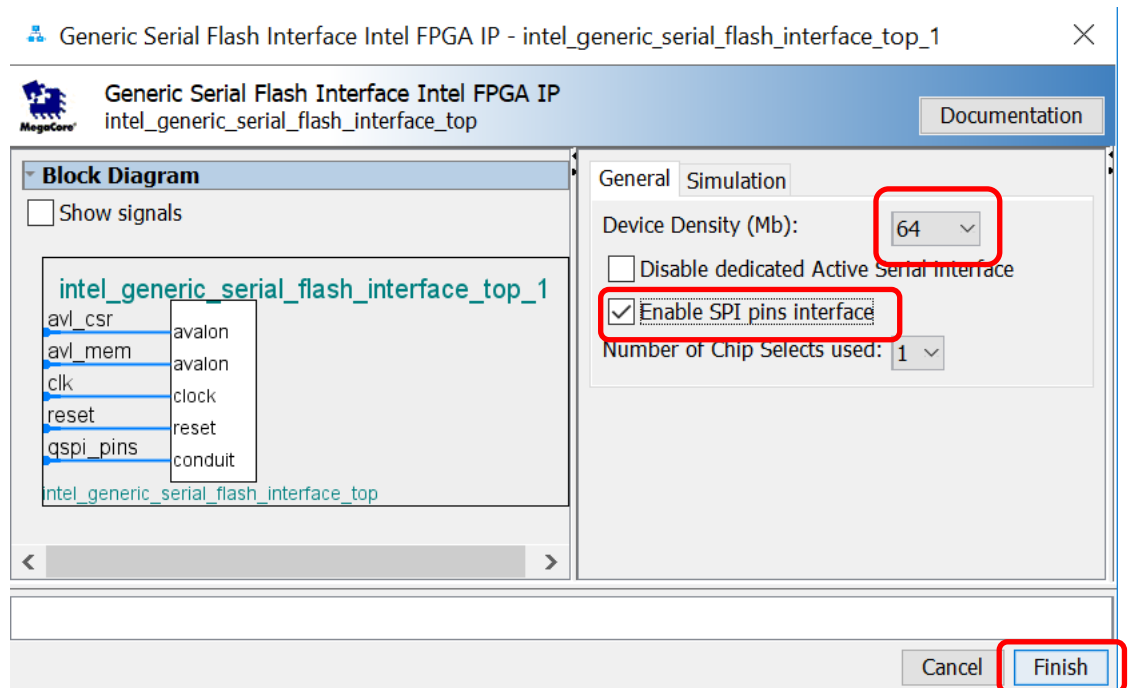
e.

6. Adding External SPI-Flash

- a. Search for “Flash” and double click “Generic Serial Flash interface Intel FPGA IP”



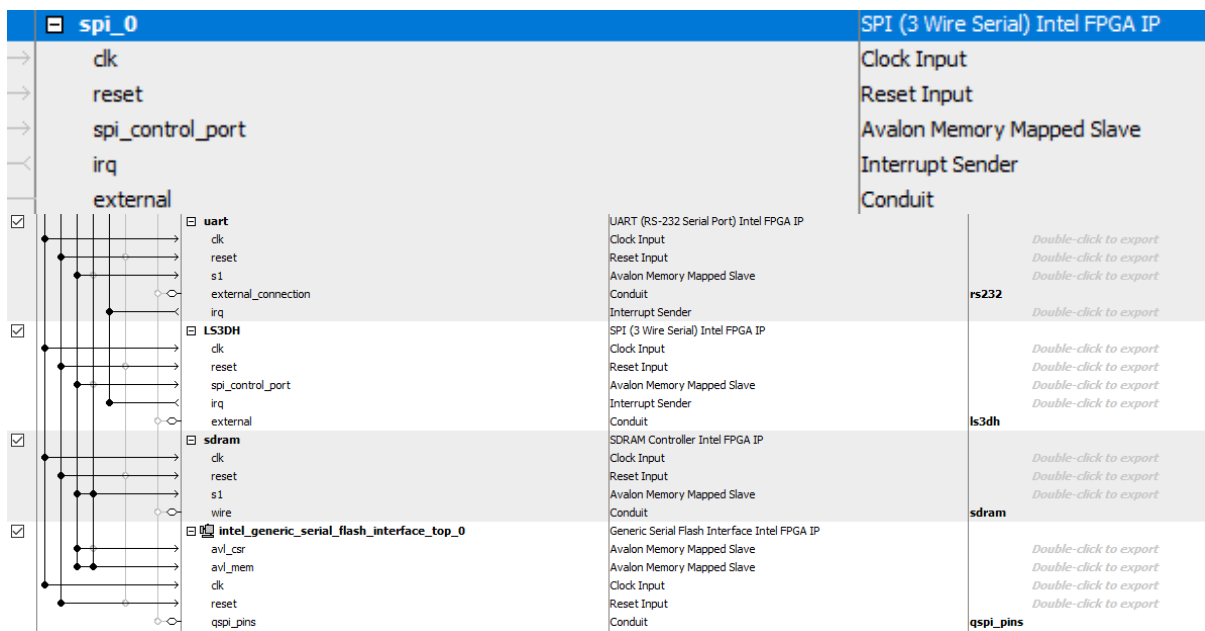
- b. Select 64 Mb Device Density and “Enable SPI pins Interface” and click “Finish”



7. Rename spi_0 to LS3DH by right click on the name and “rename”



8. Connect all buses

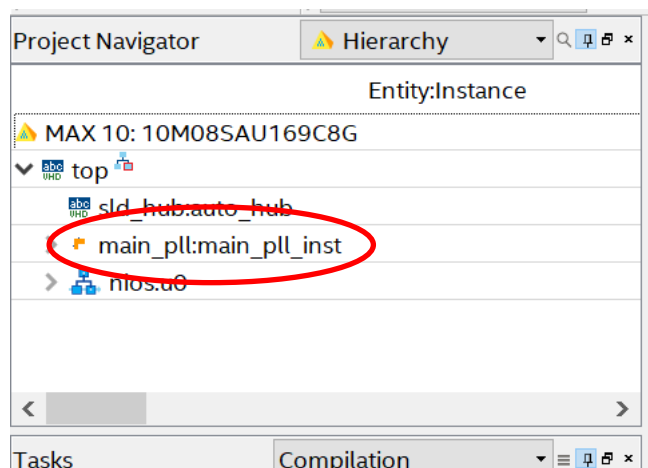


9. Click “Generate HDL”

10. After successful generation click “Finish”

11. Now we need a second clock for the SDRAM

12. In the Project Navigator of Quartus doubleclick “main_pll:main_pll_inst”



13. In the ALTPLL MegaWizard go to “Output Clocks -> clk c1

14. Set check mark at “Use this Clock”, Enter 50.000 MHz as input clk and “Clock phase shift” to -90 degree

MegaWizard Plug-In Manager [page 7 of 12] ? X

ALTPLL About Documentation

1 Parameter Settings 2 PLL Reconfiguration 3 Output Clocks 4 EDA 5 Summary

clk c0 > clk c1 > clk c2 > clk c3 > clk c4

c1 - Core/External Output Clock
Able to implement the requested PLL

☒ Use this clock

Clock Tap Settings

☒ Enter output clock frequency:
☐ Enter output clock parameters:
 Clock multiplication factor
 Clock division factor
 Clock phase shift
 Clock duty cycle (%)

Requested Settings

50.00000000 MHz
 1
 1
 -90.00 deg
 50.00

Actual Settings

50.000000
 25
 6
 -90.00
 50.00

Note: The displayed internal settings of the PLL is recommended for use by advanced users only

Per Clock Feasibility Indicators
 c0 c1 c2 c3 c4

main_pll

inclk0 frequency: 12.000 MHz
 Operation Mode: Normal

Clk	Ratio	Ph (deg)	DC (%)
c0	25/6	0.00	50.00
c1	25/6	-90.00	50.00

MAX 10

15.

16. Click "finish"

17. Now replace the old top.vhd with the provided new top.vhd.

18. To be able to assign the IO pins to the signal you need to run "Analysis & Elaboration"

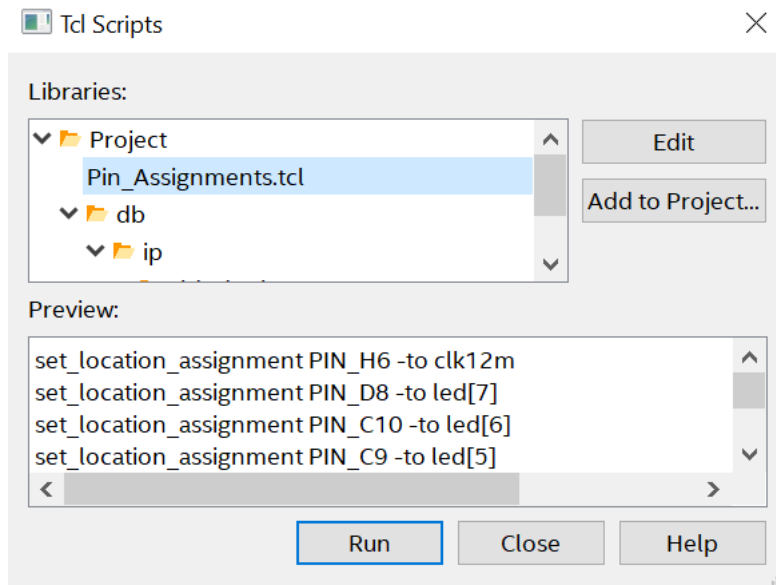
a. Click the icon for "Analysis & Elaboration" or use "Strg + k"



b. After the process is finished you can assign IOs. In this tutorial we will use a tcl script.

c. Copy Pin_Assignments.tcl from the provided USB-Stick to the project-directory

- d. In Quartus go to Menu: “Tools -> TCL scripts” select “Pin_Assignments.tcl” and click “Run”



19. Now you need to compile the design to get the needed programming files.

- a. Click the “Start Compilation” icon or use “Strg + I”



- b. This process will take a while. In the Tasks window you can follow the status

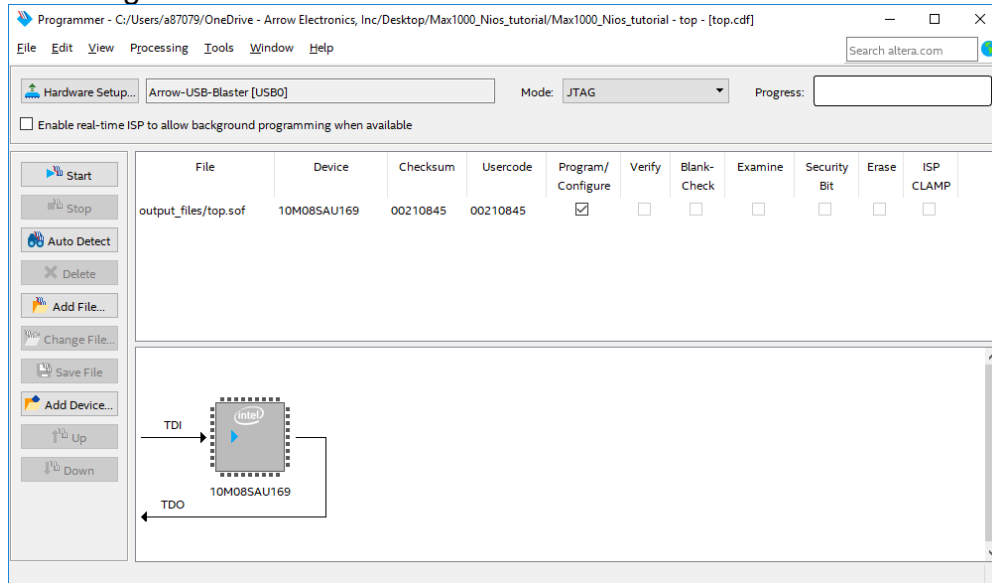
- c. After the compilation process has finished successfully the Compilation Report shows the result of the compilation process.

20. To be able to program the SRAM layer of the MAX10 you need to connect the MAX1000 Board via a Micro-USB cable to your computer.

- a. Open the Programmer tool by clicking the “Programmer” icon or use the “Tools” menu



b. The Programmer window should look like:



c. The top.sof file is used to program the SRAM layer of the MAX10 device

d. Click “Start” to start the program process

e. After the process has finished you can start the NIOSII Software tutorial.

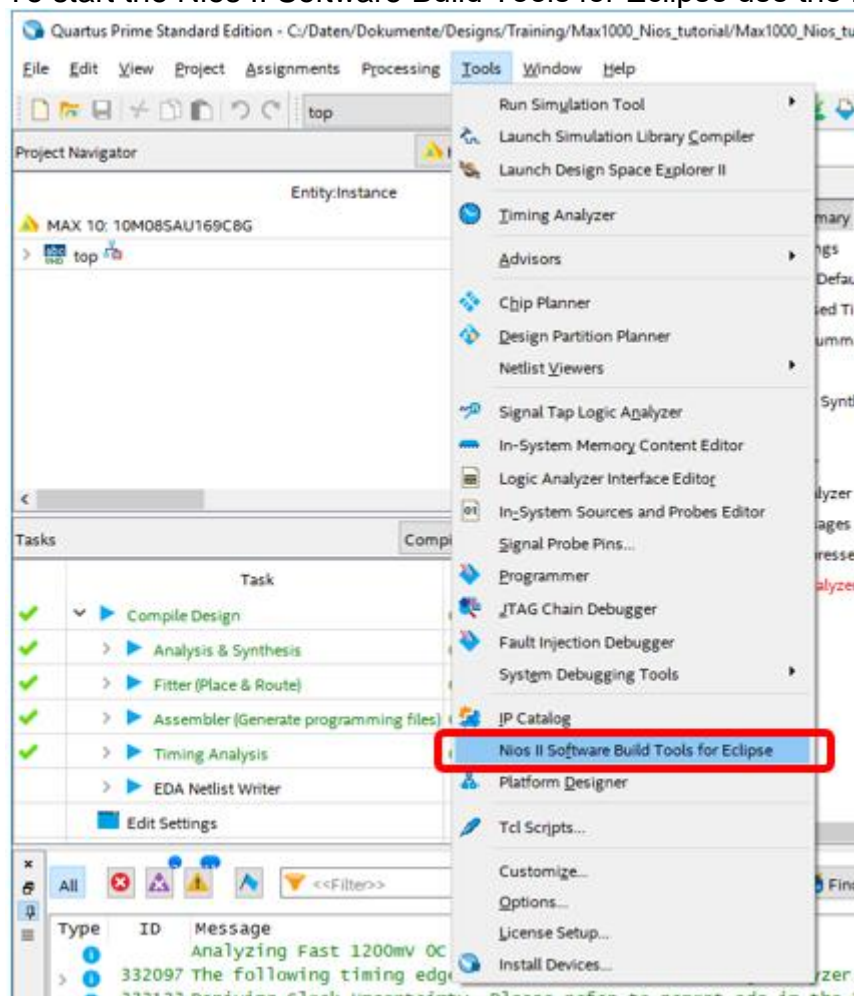
SW-Setup of the Nios II System

In this lab you will generate a C++ based NIOSII program, debug it on the MAX1000 board and generate the HEX file needed to program the UFM.

Hardware setup requirements:

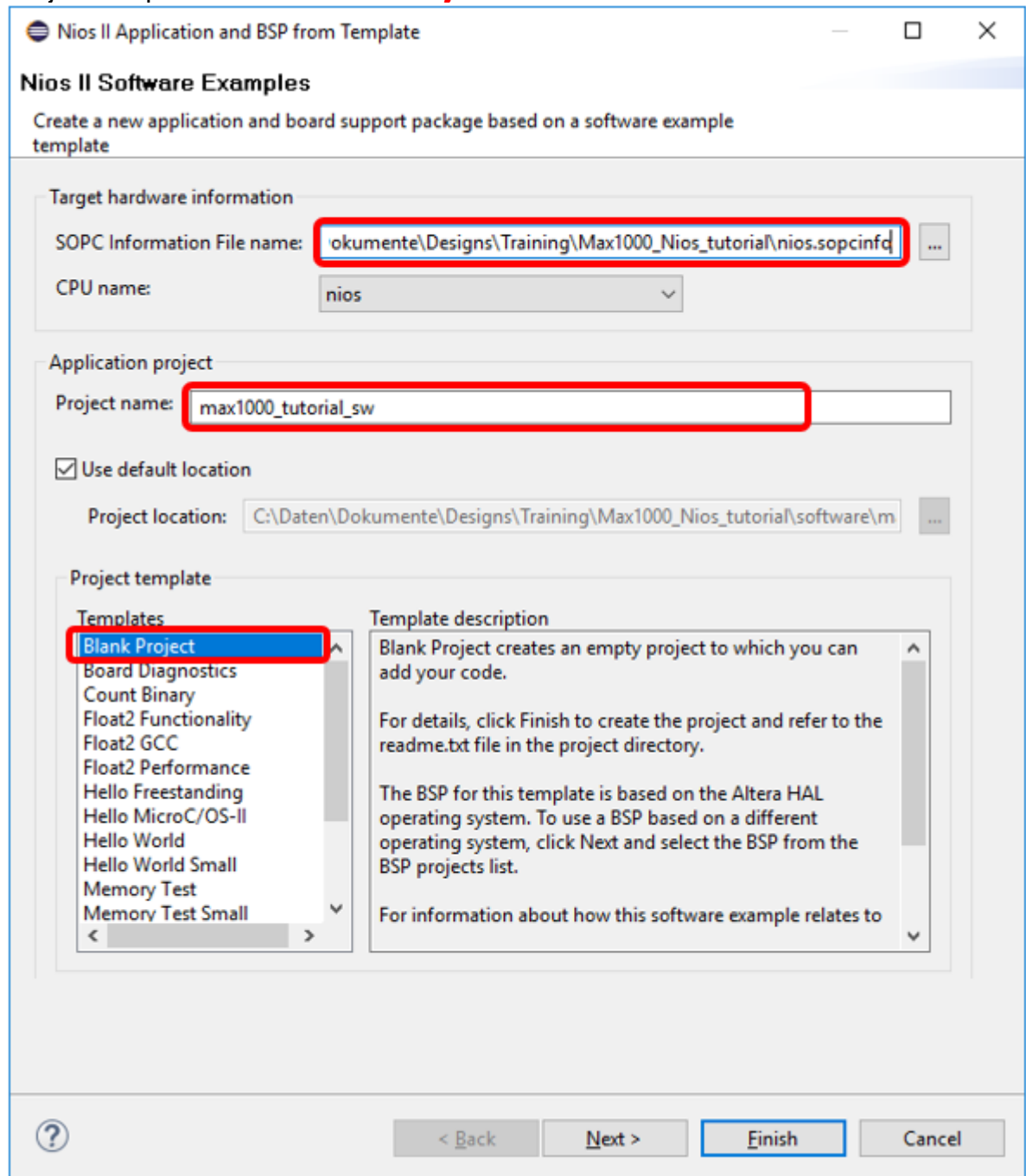
- Micro-USB cable connected between computer and MAX1000 board.
- SOF file programmed to the MAX10 device

1. To start the Nios II Software Build Tools for Eclipse use the Menu “Tools”



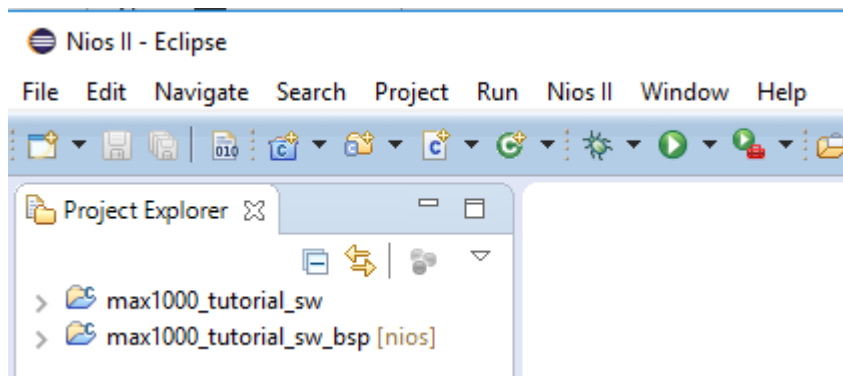
2. You need to specify a workspace
 - a. Browse to the **MAX1000_Nios_tutorial_2** directory and select the directory **“software”**
 - b. Click “OK”

- c. Nios II – Eclipse starts with an empty window
3. Now you generate a new Nios II Application
 - a. File -> New -> Nios II Application and BSP from Template
 - b. Select the SOPC Information File name: “**nios.sopcinfo**”
 - c. Project name: “**max1000_tutorial_2_sw**”
 - d. Project template: Select “**Blank Project**”

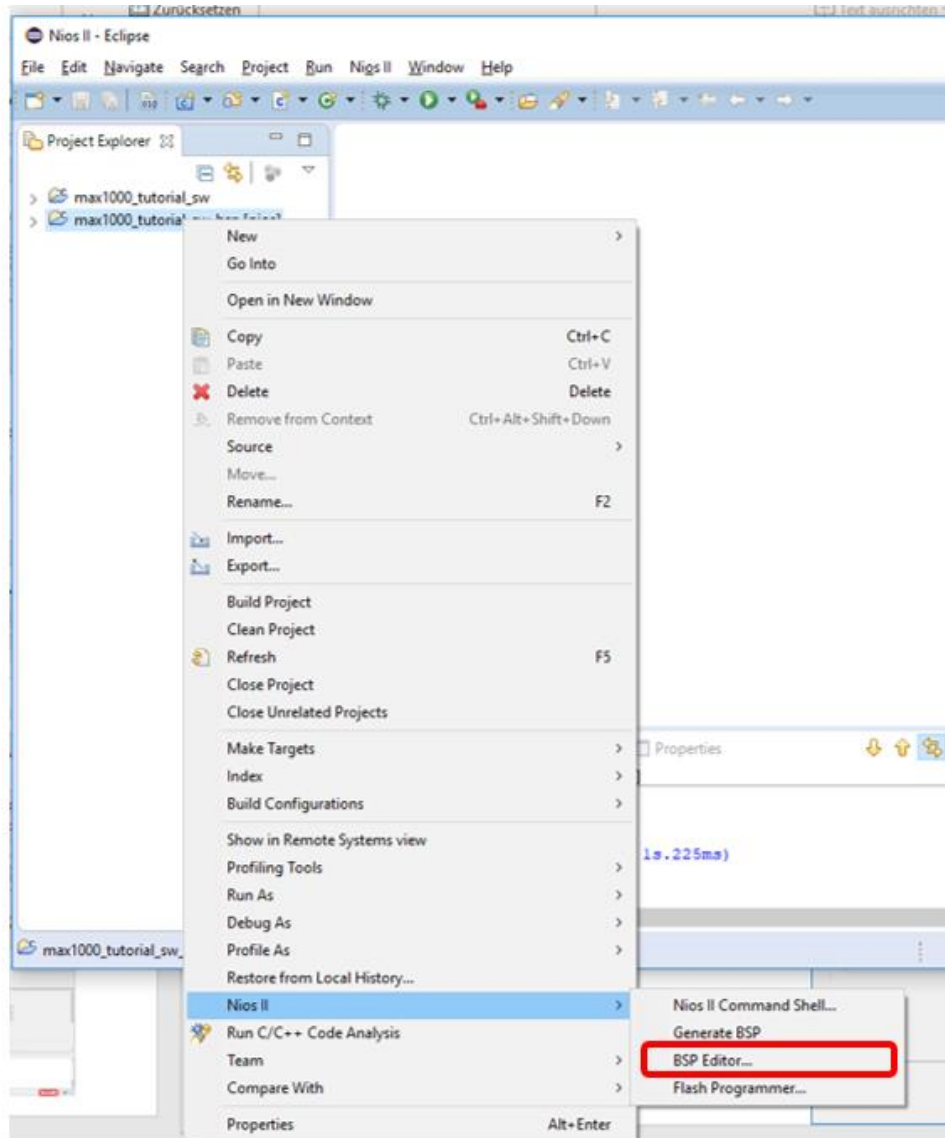


- e. Click “Finish”

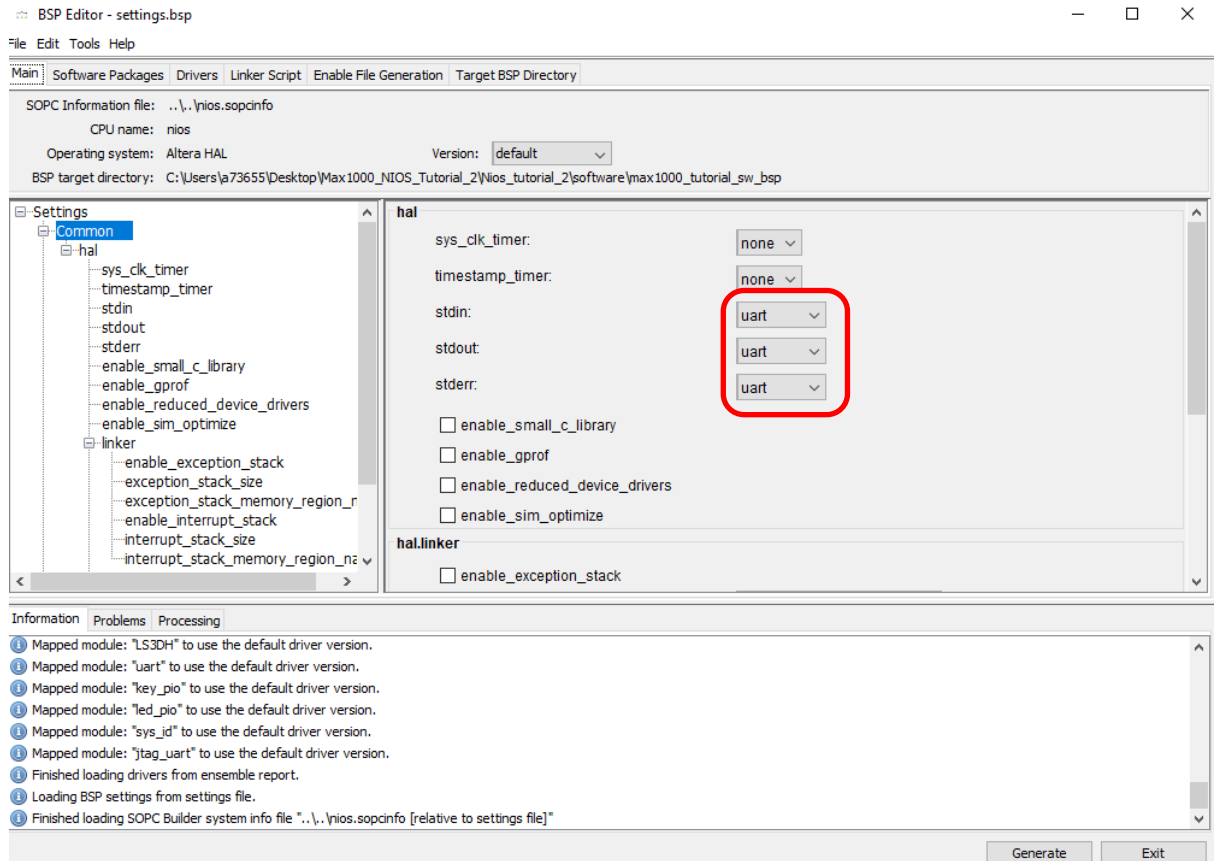
- f. In the “Project Explorer” window you now can find two folders



4. Next you need to generate the BSP for the Nios II system
- Right click the “max1000_tutorial_sw_bsp [nios]” folder
 - Select Nios II -> BSP Editor...

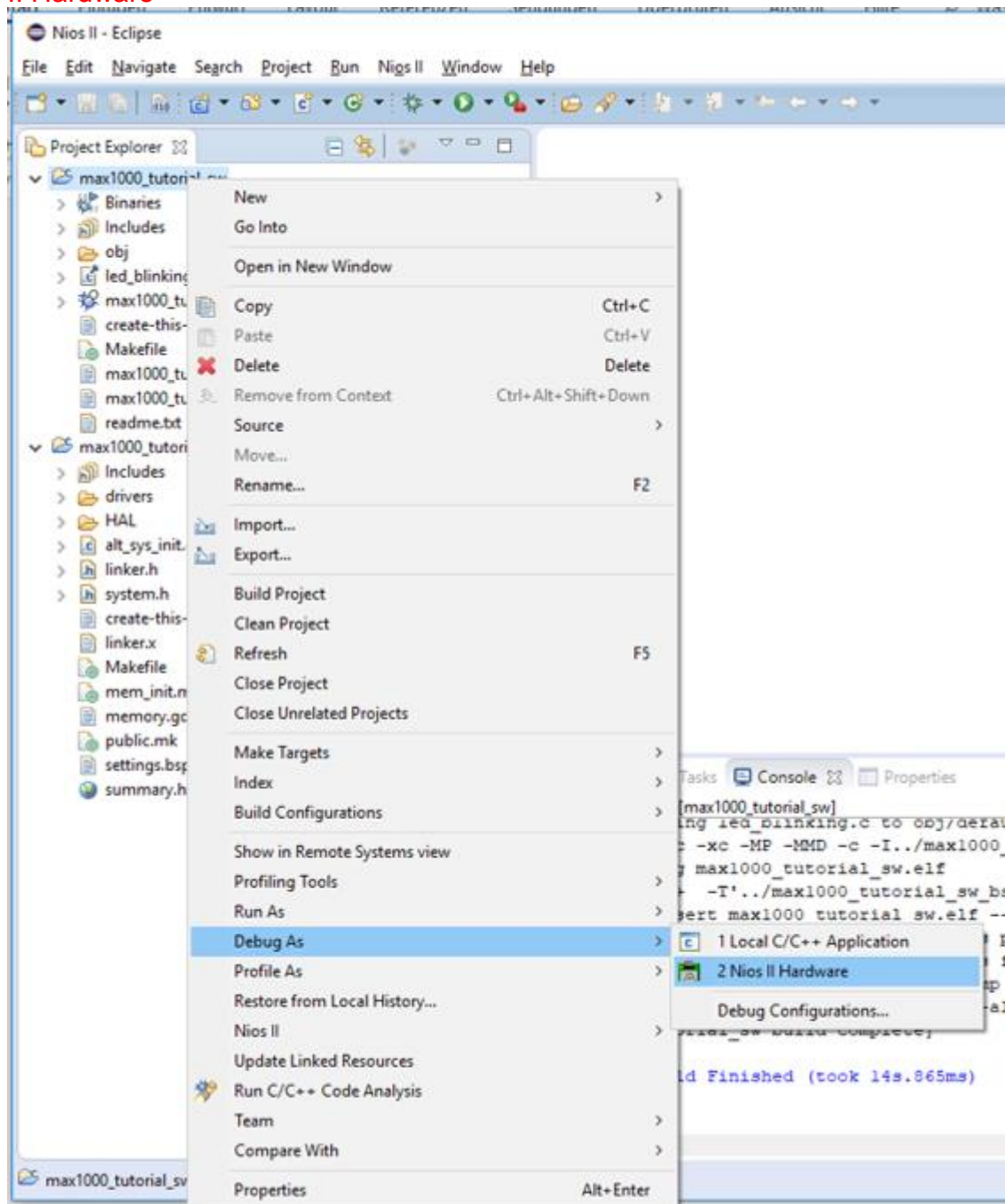


- c. You can choose, whether You want to use the integrated console, or use an external terminal program like teraterm or similar.
- d. To choose the external terminal-program just select UART for stdin, stdout and stderr.

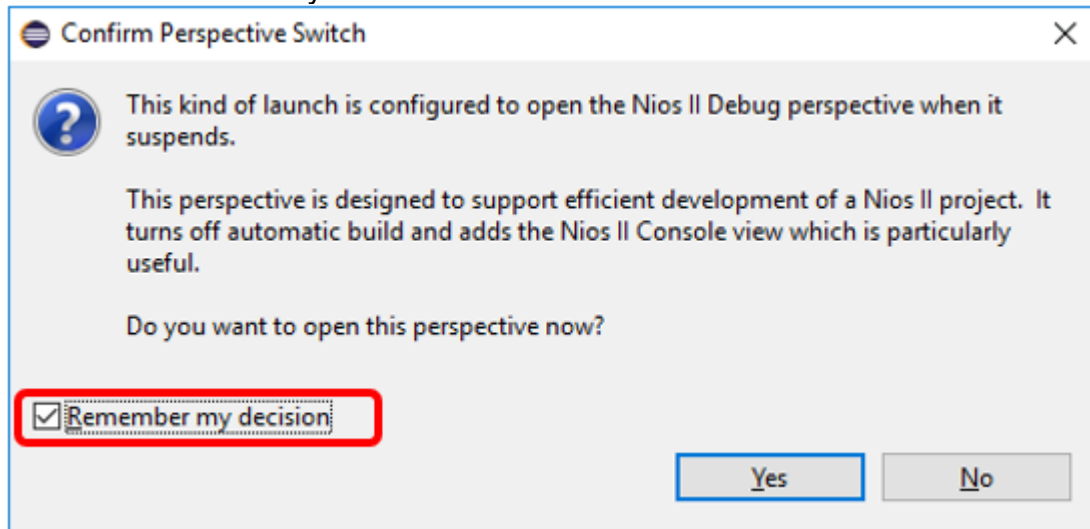


- e.
 - f. Click “Generate” and “Exit”
5. On the provided USB-Stick is a C file **memtest.c** and **operation.c**. These files must be copied into the **Max1000_NIOS_Tutorial_2\software** folder.
 - a. Use a file browser to navigate to the **Max1000_Nios_tutorial_2\software** and copy the **hello_world.c** and **operations.c** files.
 - b. In the Nios II – Eclipse window right click the **max1000_tutorial_2_sw** folder and select “Paste”
 6. To build the project right click the **max1000_tutorial_2_sw** folder and select “Build Project”
 7. Now you can debug the project

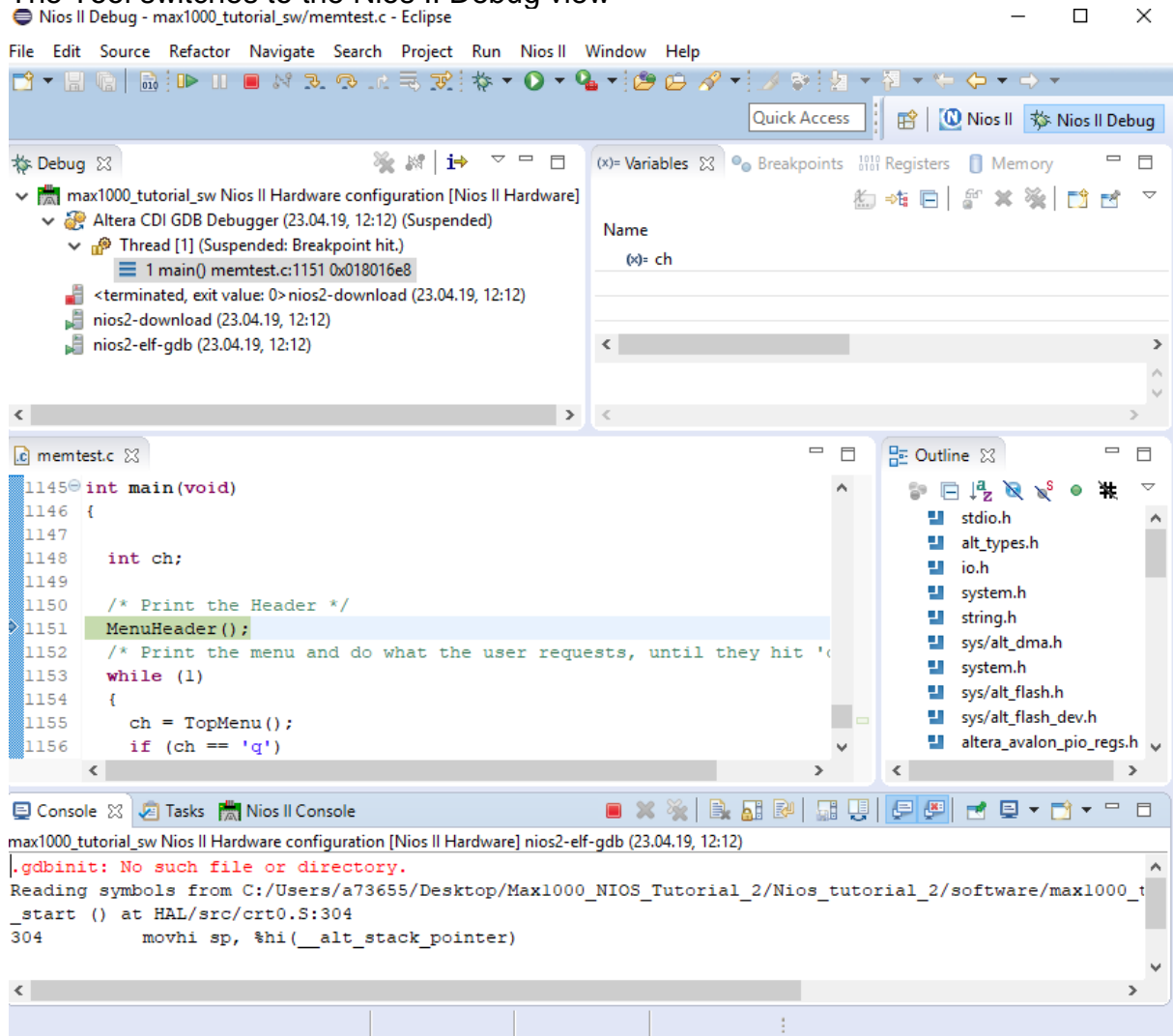
- a. Right click the `max1000_tutorial_2_sw` folder and select “Debug As” -> “2 Nios II Hardware



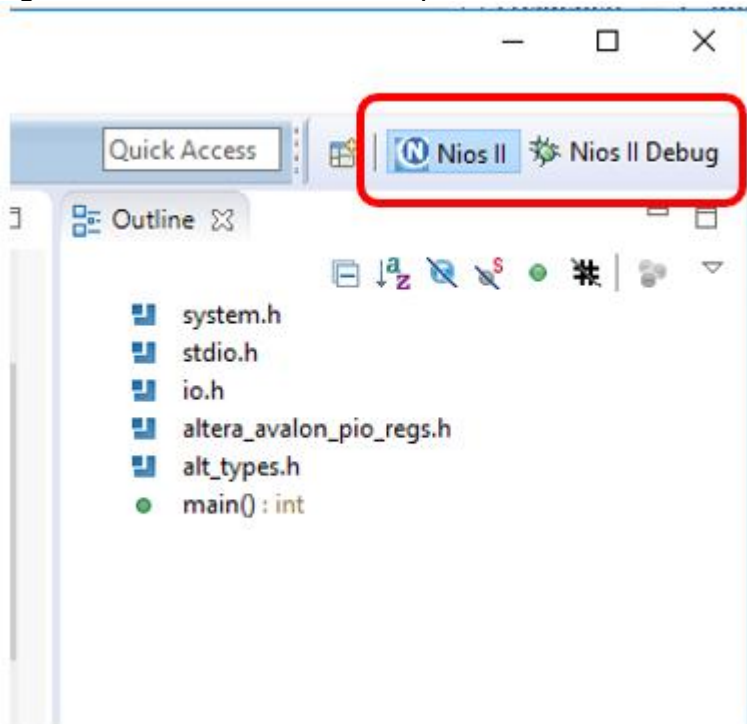
- b. Select “Remember my decision” and click “Yes”



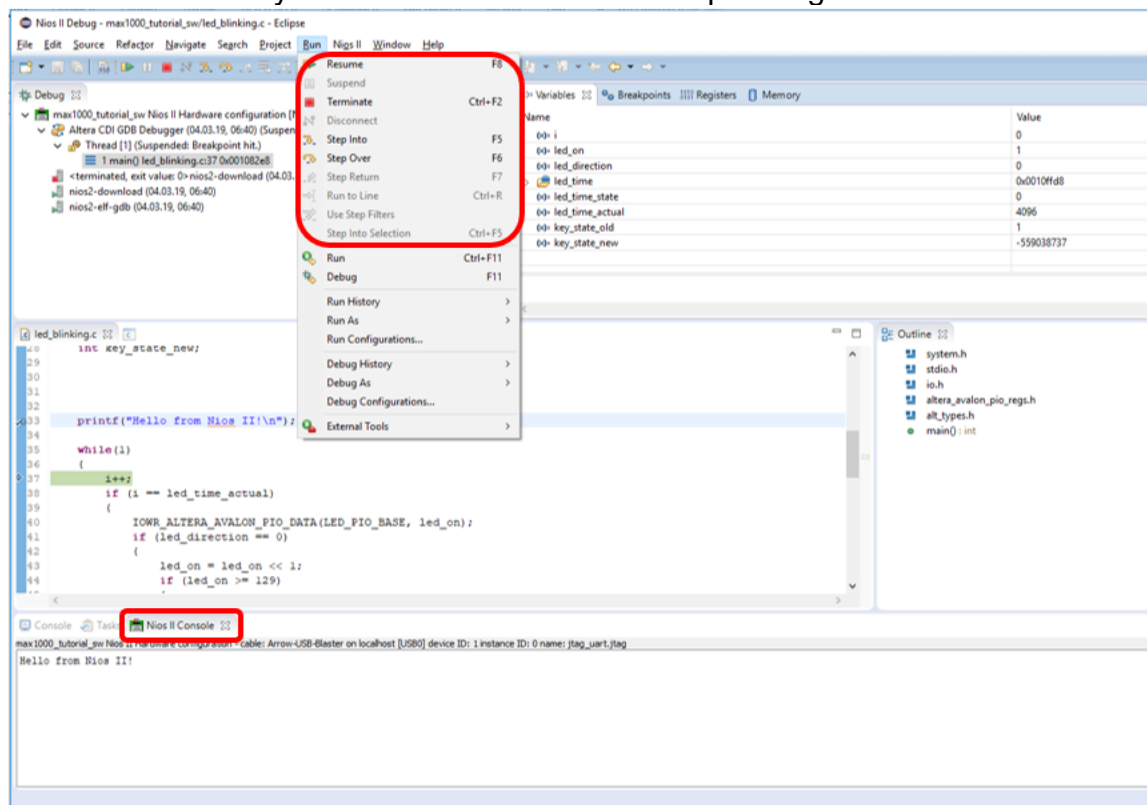
- c. The Tool switches to the Nios II Debug view



- d. Switching between the two views can be made with the two buttons on the top right corner of the Nios II Eclipse window



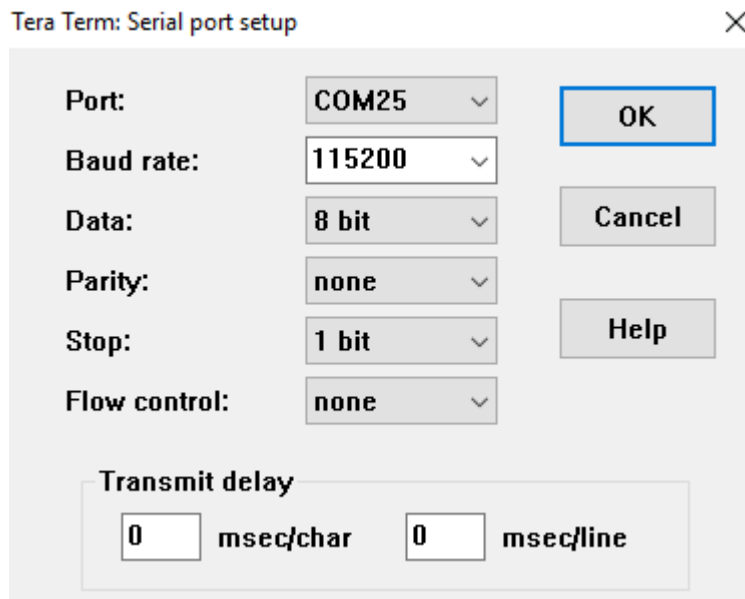
- e. After loading the ELF File to the hardware, the program stops at the first executable line of code.
- f. In the menu “Run” you can find the controls to step through the code:



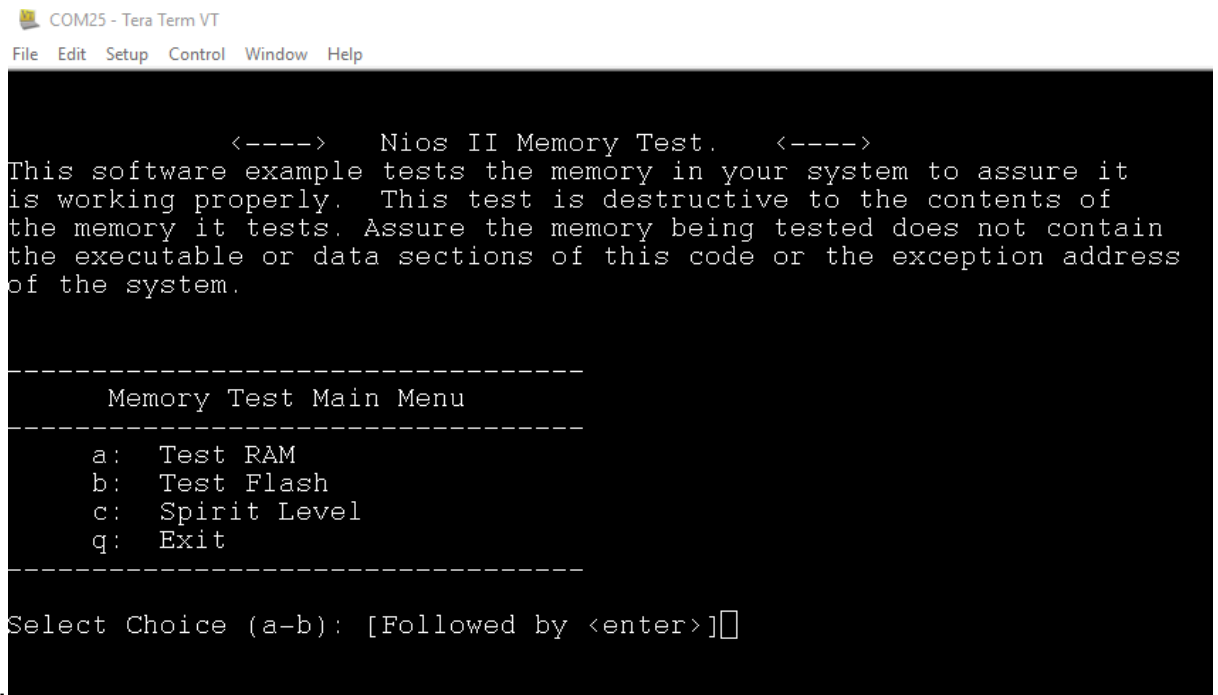
With “Step Over” or F6 the code is executed line by line

With “Step Into” or F5 you will also see the subroutines of every line
With F8 the whole code is executed until the next breakpoint (if there is any)
By double clicking on a line number in the code window you can set and delete breakpoints

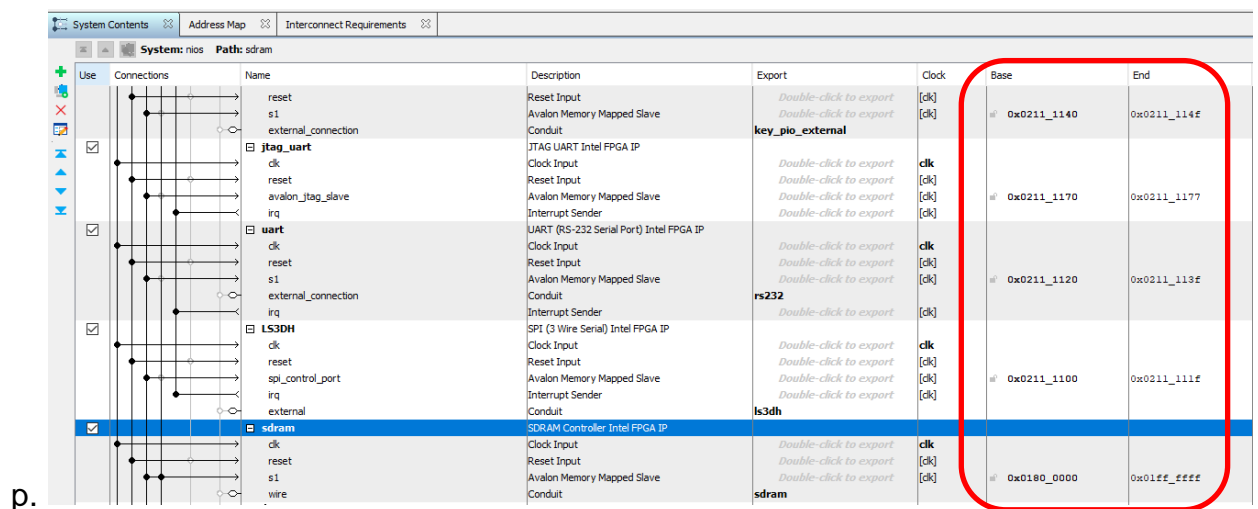
- g. In the “Nios II Console” you will find the outputs of printf commands, if You have selected JTAG-UART for stdin, stdout and stderr.
- h. Otherwise You have to open the terminal program and setup the serial connection via the virtual-com-port (in this example COM25).



- i.
- j. Press F8 in Eclipse to let the program run.
- k. **Important: If you need to reprogram the MAX10 you need first to terminate the “Debug” or “Run” session**
- l. In the terminal window (or the eclipse console) now You have a selection choice to test the onboard SDRAM, SPI-Flash and Accelerometer:



- m.
- n. To test Parts of the SDRAM enter a and “Return”
- o. Now enter Start- and End-Address for the test. The Base address of the SDRAM You can find in the Platform-Designer:



- q. Due to the program, that resides in the SDRAM, you should set the Start-Address higher, than the end of the program. This program needs about 130KB. In this example start address is set to 0x01830000. Otherwise the test will destroy the program, and the debug session is interrupted.

```
-----
Memory Test Main Menu
-----
a:  Test RAM
b:  Test Flash
c:  Spirit Level
q:  Exit
-----

Select Choice (a-b): [Followed by <enter>]a
Base address to start memory test: (i.e. 0x800000)
>0x01830000
End Address:
r. >0x018f0000
```

- s. Click Enter to start the test and view the results:

```
COM25 - Tera Term VT
File Edit Setup Control Window Help

-----
Memory Test Main Menu
-----
a:  Test RAM
b:  Test Flash
c:  Spirit Level
q:  Exit
-----

Select Choice (a-b): [Followed by <enter>]a
Base address to start memory test: (i.e. 0x800000)
>0x01830000
End Address:
>0x018f0000

Testing RAM from 0x1830000 to 0x18F0000
-Data bus test passed
-Address bus test passed
-Byte and half-word access test passed
-Testing each bit in memory device. . . passed
Memory at 0x1830000 Okay

Press enter to continue...
```

- t.

- u. Pressing enter brings you back to the menu:

```
COM25 - Tera Term VT
File Edit Setup Control Window Help
End Address:
>0x018f0000

Testing RAM from 0x1830000 to 0x18F0000
-Data bus test passed
-Address bus test passed
-Byte and half-word access test passed
-Testing each bit in memory device. . . passed
Memory at 0x1830000 Okay

Press enter to continue...

-----
Memory Test Main Menu
-----
a: Test RAM
b: Test Flash
c: Spirit Level
q: Exit
-----

v. Select Choice (a-b): [Followed by <enter>]
```

- w. For testing the W74M... SPI Onboard Flash type b followed by enter.
- x. There will be done some read, write, protect, unprotect, sector-erase and chip-erase tests.

```
Onboard flash W74M.. device will be tested.
>Flash Device found: Winbond flash W74M64FS
Device ID: 1740ef
Reading data at address 0...
Memory content at address 0: ffffffff
Address 0 not containing data...
Writing flash-ID to address 0...
Reading data at address 0...
Memory content at address 0: 1740ef
All sectors in this configuration device is not protected
Now performing sector protection...
Write register for sector protect in progress.....
All sectors in this configuration device is now successfully protected
Trying to erase sector 0...
Reading data at address 0...
Memory content at address 0: 1740ef
Now perform sector unprotect...
Sector unprotect successfully! :)
Trying to erase sector 0...
sector erase in progress.....
Reading data at address 0...
Memory content at address 0: ffffffff
Address 0 not containing data...
Writing data to address 0...
Current memory in address 0: 100
Current memory in address 1: 101
Current memory in address 2: 102
Current memory in address 3: 103
Current memory in address 4: 104
Current memory in address 5: 105
Current memory in address 6: 106
Current memory in address 7: 107
Current memory in address 8: 108
Current memory in address 9: 109
Current memory in address 10: 10a
Current memory in address 11: 10b
Current memory in address 12: 10c
Current memory in address 13: 10d
Current memory in address 14: 10e
Current memory in address 15: 10f
Read data match with data written. Write memory successful.
Trying to erase chip
.....
Reading data at address 0...
Memory content at address 0: ffffffff
Chip erase successful.
Press enter to continue...
```

y. ☐

- z. At last, there is a spirit level application to test the accelerometer. Type c followed by enter to start this. The onboard LEDs show the level of the board. By tilting the board, you can see the LEDs lighting accordingly. You can finish the test, by tilting the board more than 50° in one direction (USB port down).

Congratulation, you have successfully finished the second Part of MAX1000 tutorial.