



FreeRTOS/lwIP on a NIOS II Softcore to run open62541

28. April 2020

The [Trenz Cyclone 10 LP RefKit](#) is a very interesting Development Board on the market with an Intel (former Altera) FPGA.

In our last few blog posts, we were showing how to port open62541 to a Xilinx MicroBlaze Softcore CPU. To follow up the open62541 topic we were able to get the stack up and running also on NIOS II Softcore CPU.

Since open62541 already supports FreeRTOS with lwIP (which is also what we were using on the MicroBlaze designs), the goal was to first get FreeRTOS and lwIP running on NIOS II.

Based on a very nice NIOS II port from EngineeringSpirt (<https://github.com/EngineeringSpirt/FreeLwIP-Nios-II>) we have updated the sources to the latest lwIP and FreeRTOS versions. In addition some modifications were made to get it working with Quartus 18.1.

Together with the Trenz Cyclone 10 LP RefKit NIOS II reference design and the FreeRTOS/lwIP boards support package we open the possibility to get open62541 running on an Intel FPGA device with a Softcore CPU.

The forked and modified FreeLwIP-Nios II project is available here:

<https://github.com/NetTimeLogic/FreeLwIP-Nios-II>

The example FPGA project and the application are available here:

https://github.com/NetTimeLogic/opcua/tree/master/Src/Quartus_Project

The open62541 implementation is available here (master):

<https://github.com/open62541/open62541/tree/master>

Overview

The open62541 implementation is basically the same as for the MicroBlaze and therefore no further details about this implementation steps are described.

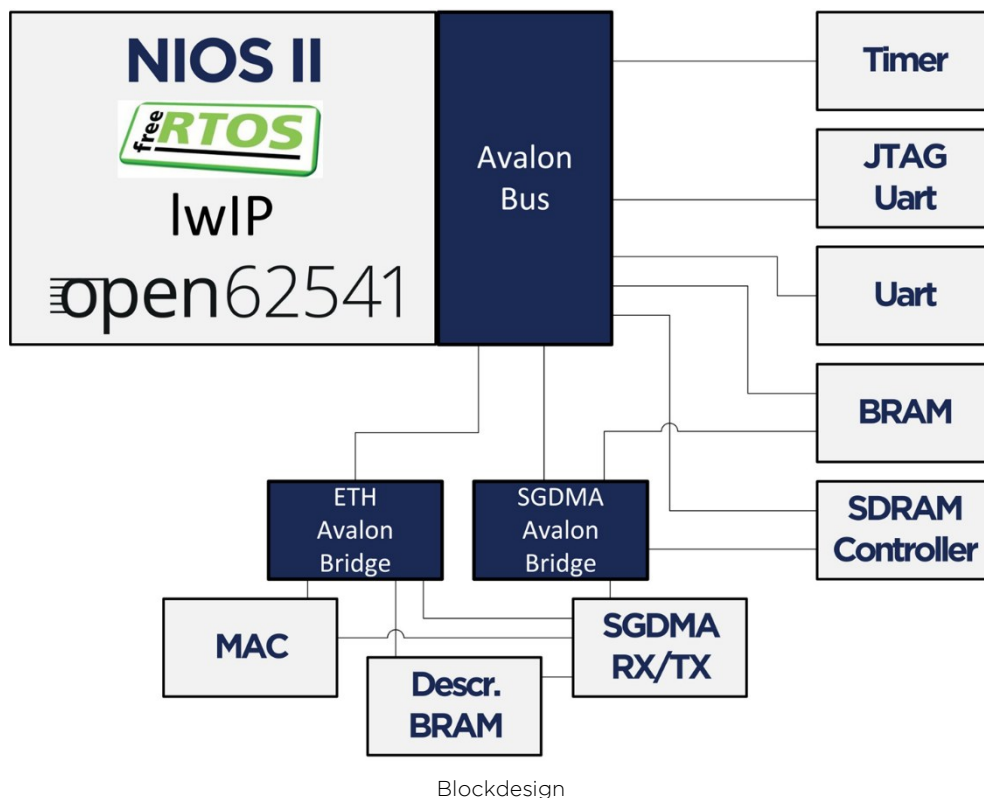
An adapted cmake file for the NIOS solution is available:

<https://github.com/NetTimeLogic/opcua/blob/master/Src/SdkNios.cmake>

The precompiled (AMALGAMATION) files of open62541 are part of the sources in the NIOS II project:

https://github.com/NetTimeLogic/opcua/tree/master/Src/Quartus_Project/software/FreeRTOS

On the NIOS II the latest FreeRTOS and lwIP version are building the base for an open62541 application. On the FPGA a QSYS system is build up with all required components. The application runs from the SDRAM, and via Scatter-Gatter DMA and a TSE MAC the Ethernet handling is done. The timer is required for FreeRTOS (system tick) and the UART interfaces can be used for the printouts. The System runs on 100MHz.



The reference design is running on the Trenz Cyclone 10 LP RefKit. More details about this kit is available here:

<https://shop.trenz-electronic.de/de/TEI0009-02-055-8CA-Cyclone-10-LP-RefKit-10CL055-Development-Board-32-MByte-SDRAM-16-MByte-Flash?c=473>

Toolchain

To build the full project the following tools are required:

- Quartus Prime 18.1 (64bit)
- Nios II Embedded Design Suite 18.1
- [CMAKE](#) (Python 2.7.x or 3.x)
- [UA Expert](#)
- [Wireshark](#)

BSP

The BSP has to be installed via the alt-freertos-lwip-install.bat script. Call this script from a command line and follow the instructions:

<https://github.com/NetTimeLogic/FreeLwIP-Nios-II/blob/master/alt-freertos-lwip-install.bat>

For the TSE MAC HAL driver we are not using the one of the Quartus 18.1 version. The Quartus 18.1 version requires the MSGDMA (Modular Scatter Gatter DMA)

instead of the older SGDMA. Due to the existing environment which was based on the SGDMA, we have decided to not make this change right now.

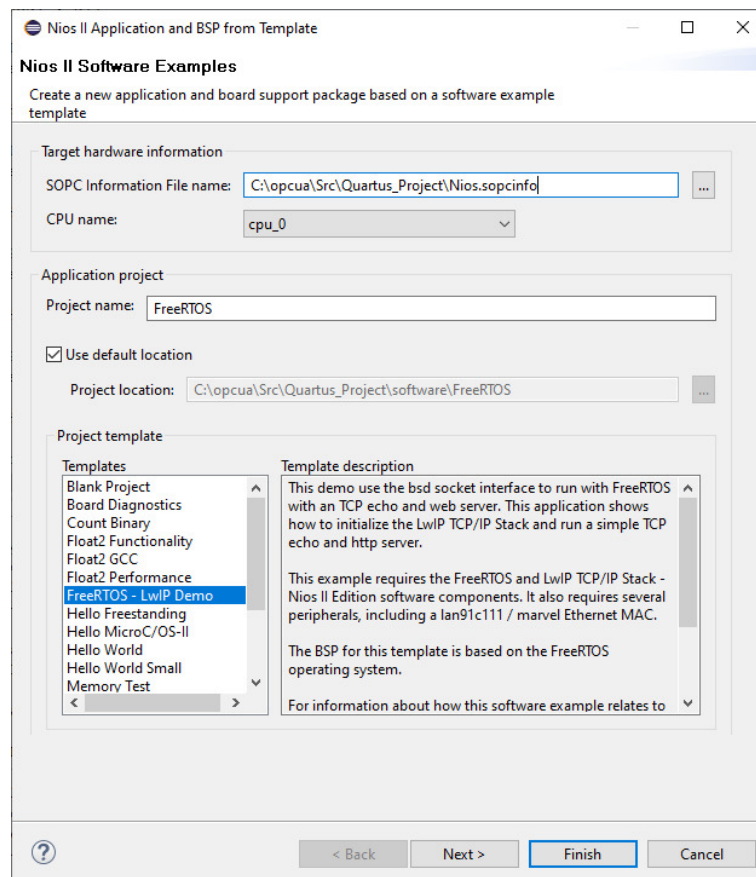
Beside that, the BSP was mainly tested for the required functionality with open62541 and it is still possible that not all lwIP features are fully working (e.g. IPv6).

Most parameters are editable directly via the BSP Editor. Depending on the needs there might be still some parameters which are not in the GUI.

Design preparation

Creating the design is quite easy since the script creates a full template.

1. Go to File->New->Nios II Application and BSP from Template
2. Select FreeRTOS - LwIP Demo with the Nios.sopcinfo



3. With the Finish Button the Demo Project will be created.

4. In the BSP the following settings must be changed:

Right Click on the BSP, NIOS II-> BSP Editor ...

- max_file_descriptors: 16
- (Optional) stdin, stdout, stderr: uart_0

5. Close the BSP Editor and regenerate the BSP
6. The demo application can be built; however, the demo was never tested as such on the hardware.
7. Remove the files and create your own application (e.g. main.c; iicNs.c/h and open62541.c/h)

Summary

Thanks to the well prepared FreeRTOS with LwIP integration in the Nios II EDS from EngineeringSpirit it was possible to get open62541 up and running on an Intel FPGA with a soft core inside.

We have updated the existing port to the Quartus 18.1 version and also used the latest lwIP and FreeRTOS versions. The update to the latest versions, a reference design for the Trenz Cyclone 10 LP RefKit and the demo PubSub example opens new and interesting opportunities to have OPC UA running on an Intel FPGA with a SoftCore.

Since the open62541 application is the same as for our performance test we have quickly rerun some of them on the NIOS II CPU.

A comparison between the two softcore variants can be found in the figure below even if they are a bit hard to directly compare (both @ 100MHz, but program code is running on DDR vs SDRAM etc.). Nevertheless it shows what performance could be achieved on such a soft core CPU.

	1 field (1 dynamic)		15 fields (2 dynamic)	
	<i>MicroBlaze @ 100MHz (DDR)</i>	<i>Nios @ 100MHz (SDRAM)</i>	<i>MicroBlaze @ 100MHz (DDR)</i>	<i>Nios @ 100MHz (SDRAM)</i>
<i>UA_PUBSUB_RT_NONE</i>	39	134	11	39
<i>UA_PUBSUB_RT_DIRECT_VALUE_ACCESS</i>	64	218	29	110
<i>UA_PUBSUB_RT_FIXED_SIZE</i>	90	510	64	185

