PROFESSIONAL
INDUSTRIAL
COMMUNICATION

Integration of SoM

for STM32F4xx Application Controller

port GmbH

Regensburger Str. 7

D-06132 Halle/Saale

Disclaimer

This manual represents the current state of the product. Please check with port.de for the latest version as the document may have a newer version since errors may be corrected or changes for a newer version of the product may be incorporated. Port.de assumes no responsibility for errors in this document. Qualified feedback is appreciated at service@port.de.

This document is the Intellectual Property of port.de and is intended to be used with the described product only. It may be forwarded and/or copied in the original and unmodified format. All rights reserved.

The product enables to use technologies such as PROFINET, EtherNet/IP and/or EtherCAT and others. These technologies are promoted by trade organizations, such as PNO (profibus.org), ODVA (odva.org) or ETG (ethercat.org). These trade organizations as well maintain the specification and care about legal issues.
We strongly recommend to become a member of these organisations. Most technologies are making use of patented or otherwise copyrighted technologies, approaches or other intellectual property. The membership usually automatically entitles the member for use of most of the technology-inherent copyrighted or otherwise protected Intellectual Property of the corresponding trade organization and most 3rd parties. Otherwise the user will need to obtain licenses for many patented technologies separately.

Further we suggest to you to subscribe to the corresponding Conformance Test Tool of these trade organizations. For instance the ODVA only accepts conformance test applications from companies who have a valid membership and have a valid subscription to the recent Conformance Test Tool. We as port are members in all corresponding organizations and are holding a subscription to these tools - however you as a customer need to have an own membership and an own subscription to the tool.

All rights reserved

The programs, boards and documentations supplied by port GmbH are created with due diligence, checked carefully and tested on several applications.
Nevertheless, port GmbH cannot guarantee and nor assume liability that the program, the hardware board or the documentation are error-free or appropriate to serve a specific customer purpose. In particular performance characteristics and technical data given in this document may not be interpreted to be guaranteed product features in any legal sense.

For consequential damages, every legal responsibility or liability is excluded.
port has the right to modify the products described or their documentation at any time without prior warning, as long as these changes are made for reasons of reliability or technical improvement.
All rights of this documentation are with port. Unless expressively granted - the transfer of rights to third parties or duplication of this document in any form, whole or in part, is subject to written approval by port. Copies of this document may however be made exclusively for the use of the user and his engineers. The user is thereby responsible that third parties do not obtain access to these copies.
The soft- and hardware designations used are mostly registered and are subject to copyright.

Regensburger Straße 7
D-06132 Halle
Tel. +49 345 - 777 55 0
Fax. +49 345 - 777 55 20
E-Mail service@port.de
www.port.de
www.port-automation.com

Contents

Table of figures

Changelog

Version	Changes
1.0	Initial release

1 Introduction

This document describes integration delivered example projects for SoM using the STM32F4xx target with development environment STM32 Cube IDE.

Within the document, special recommendations are given marked by two signs:

ℹ Special information giving hints to avoid common pitfalls when using the software

⚠ Special information to prevent malfunction of the software or that require special attention of the user.

2 Preparations

2.1 Hardware Requirements

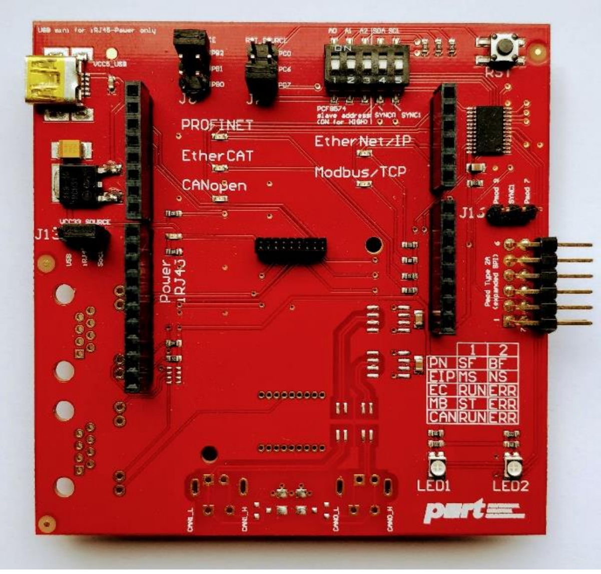


Figure 1 communication module with shield board

Make sure following settings are set:

J13: Connect "VCC33_Sockel" with "VCC33_RJ45"
J8: Connect "CS" with "PB2"
J7: Connect "RST" with "PD7"
All DIP-Switches to OFF

2.2 STM32F429I-Nucleo board

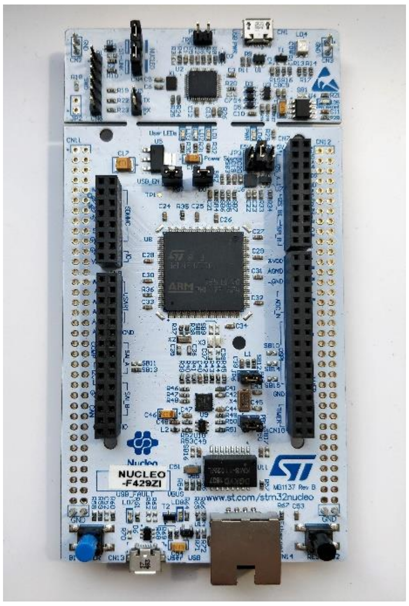


Figure 2 NUCLEO-F429ZI

2.3 Target System

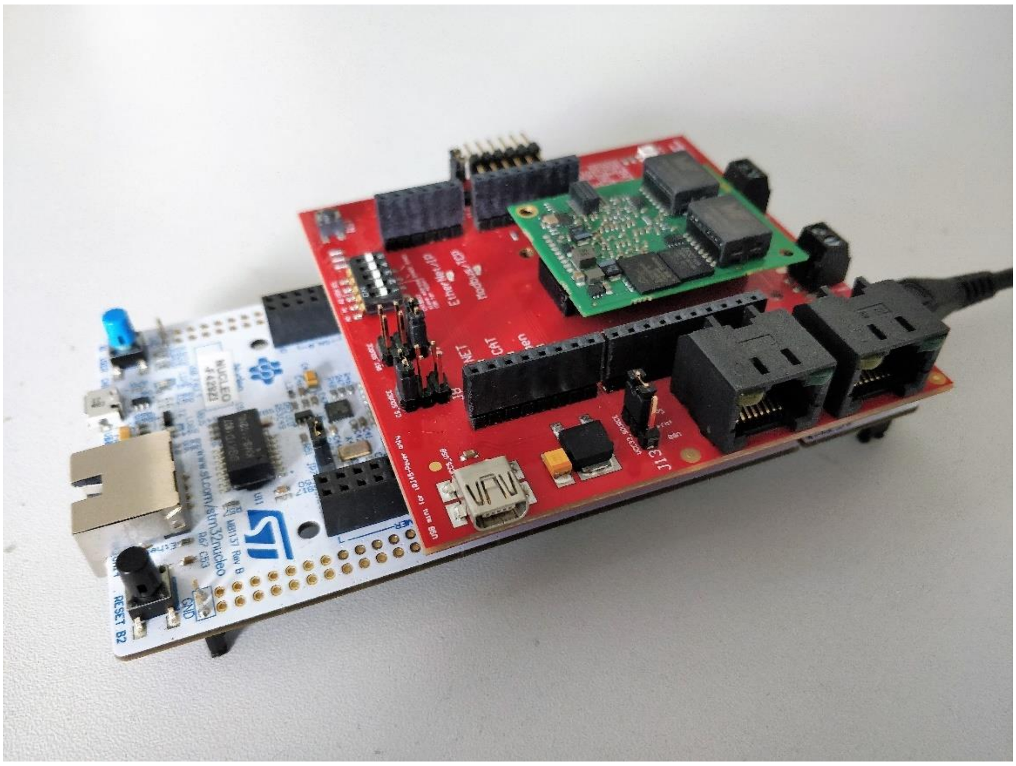


Figure 3 Target system

Connect both boards (see Figure 3 Target system). Voltage supply is done through the shown USB connector (USB Pwr) on the development board, which also supports flashing and debugging of the CPU.

2.4 Software Requirements

2.4.1 Development environment "STM32F4"

The development of software for the AC requires an installation of the STM32 Cube IDE. Version 1.0 was used. The toolchain can be obtained here:
https://www.st.com/en/development-tools/stm32cubeide.html

2.4.2 Delivery

- The delivery consists of 2 files:

File	Content
"2015013_20190611_2_20_0_port_STM32F429_Nucleo144_CCM_ci125.zip"	Goal Library and Headers
"iRJ45 Management Tool-201808221413-win32.win32.x86_64"	Management Tool

2.4.3 Management Tool

To use the PROFINET master functionality, WinPCap[1] needs to be installed.

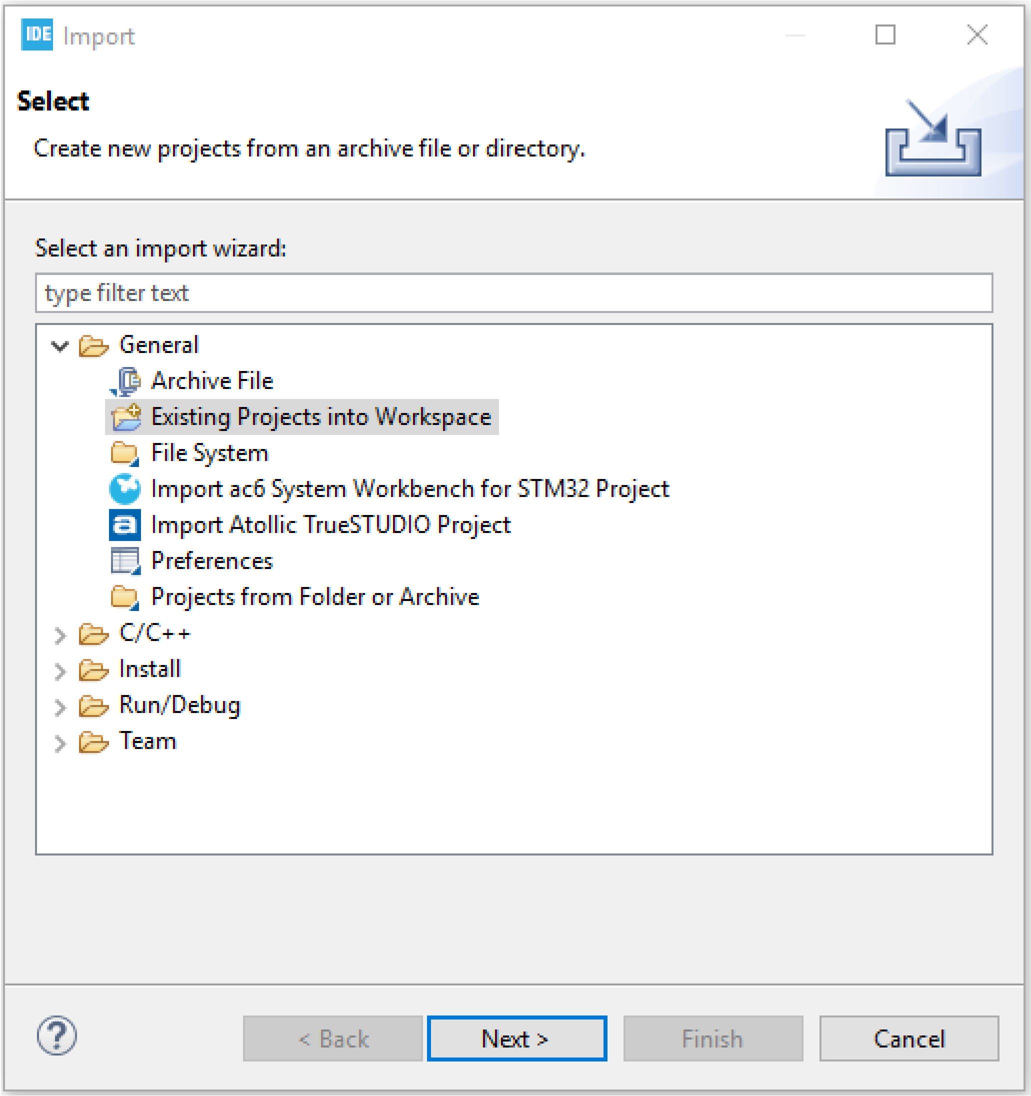
Please unpack the delivery " iRJ45 Management Tool-*-win32.win32.x86_64.zip" to a local folder. The resulting folder contains the executable “mantool”, which can be started.

2.4.4 Prepare goal and project

Unpack the goal headers and library (2015013*.zip) to a local folder. This folder contains the goal library and the associated headers for synergy required to build an application for the SoM. Beside that example projects for e2studio are contained.

[1] https://www.winpcap.org/

3 Import project



Import

Select

Create new projects from an archive file or directory.

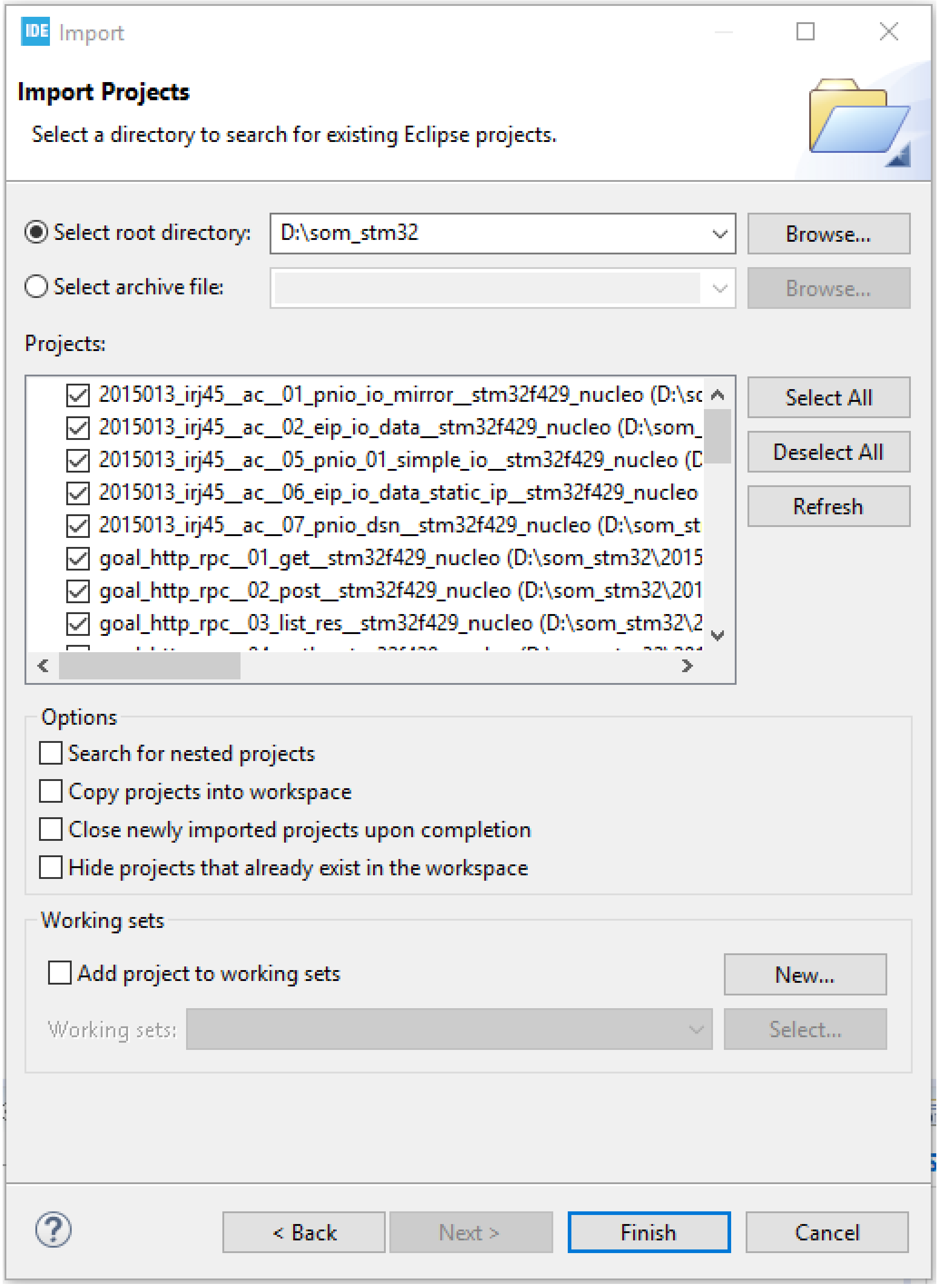
Select an import wizard:

type filter text

General
Archive File
Existing Projects into Workspace
File System
Import ac6 System Workbench for STM32 Project
Import Atollic TrueSTUDIO Project
Preferences
Projects from Folder or Archive
C/C++
Install
Run/Debug
Team
< Back
Next >
Finish
Cancel

Figure 4 IDE Import dialog

Use the import dialog of STM Cube IDE to import the project located in the unpacked project delivery into the IDE. Chose "Existing projects into workspace" when prompted for import type.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: D:\som_stm32 Browse...

○ Select archive file: Browse...

Projects:

- ☑ 2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo (D:\sc
- ☑ 2015013_irj45__ac__02_eip_io_data__stm32f429_nucleo (D:\som_
- ☑ 2015013_irj45__ac__05_pnio_01_simple_io__stm32f429_nucleo (D
- ☑ 2015013_irj45__ac__06_eip_io_data_static_ip__stm32f429_nucleo
- ☑ 2015013_irj45__ac__07_pnio_dsn__stm32f429_nucleo (D:\som_st
- ☑ goal_http_rpc__01_get__stm32f429_nucleo (D:\som_stm32\2015
- ☑ goal_http_rpc__02_post__stm32f429_nucleo (D:\som_stm32\201
- ☑ goal_http_rpc__03_list_res__stm32f429_nucleo (D:\som_stm32\2

Options

☐ Search for nested projects

☐ Copy projects into workspace

☐ Close newly imported projects upon completion

☐ Hide projects that already exist in the workspace

Working sets

☐ Add project to working sets

Working sets:

Figure 5 Import project dialog

Select the root directory of the unpacked delivery project for import and “Finish” import (see Figure 5 Import project dialog). There are several types of projects, which can be imported all or selectively.

After importing the projects can be built. Use the project context menu, menu item “Build Project” to do so. As a result, a binary should be generated and the “Console” log should show the following text:

```
…
Finished building target:
2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo.elf

arm-none-eabi-size   2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo.elf
arm-none-eabi-objdump -h -S
2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo.elf  >
"2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo.list"
arm-none-eabi-objcopy  -O binary
2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo.elf
"2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo.bin"
   text    data        bss        dec          hex      filename
 190440     888      136000     327328        4fea0
  2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo.elf
Finished building: default.size.stdout

Finished building: 2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo.bin

Finished building: 2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo.list


13:00:37 Build Finished. 0 errors, 358 warnings. (took 16s.925ms)
```

If so, compilation was successful. The resulting binary now can be started:

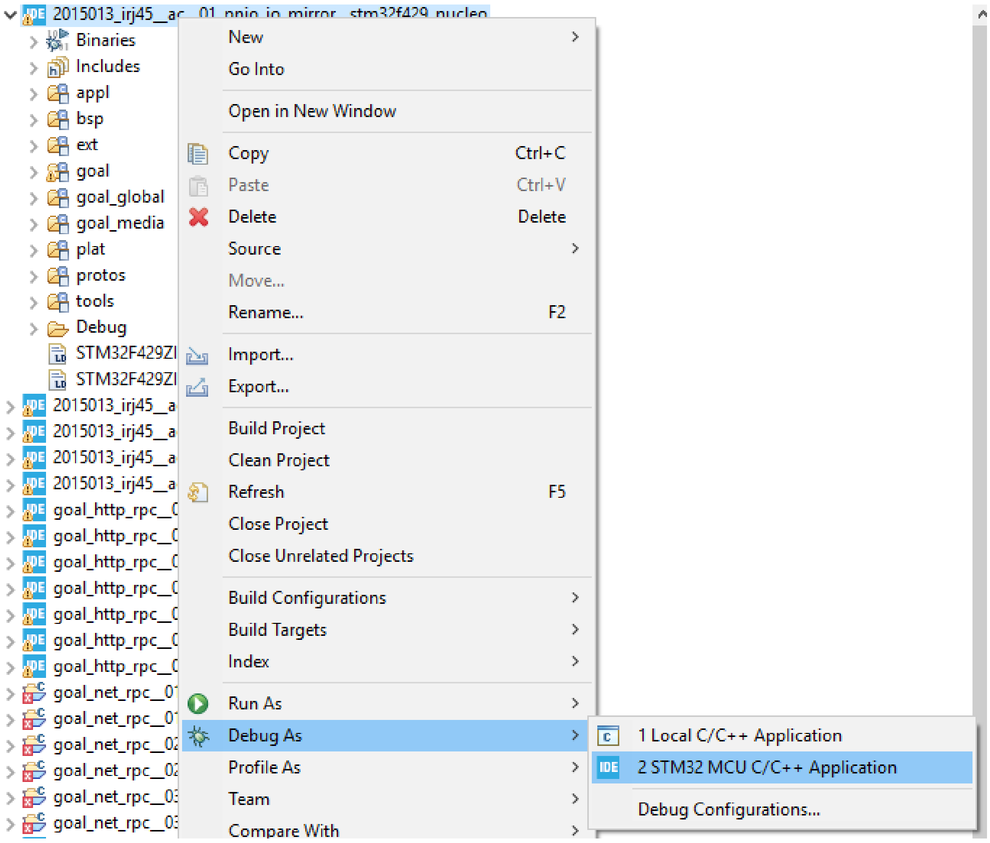
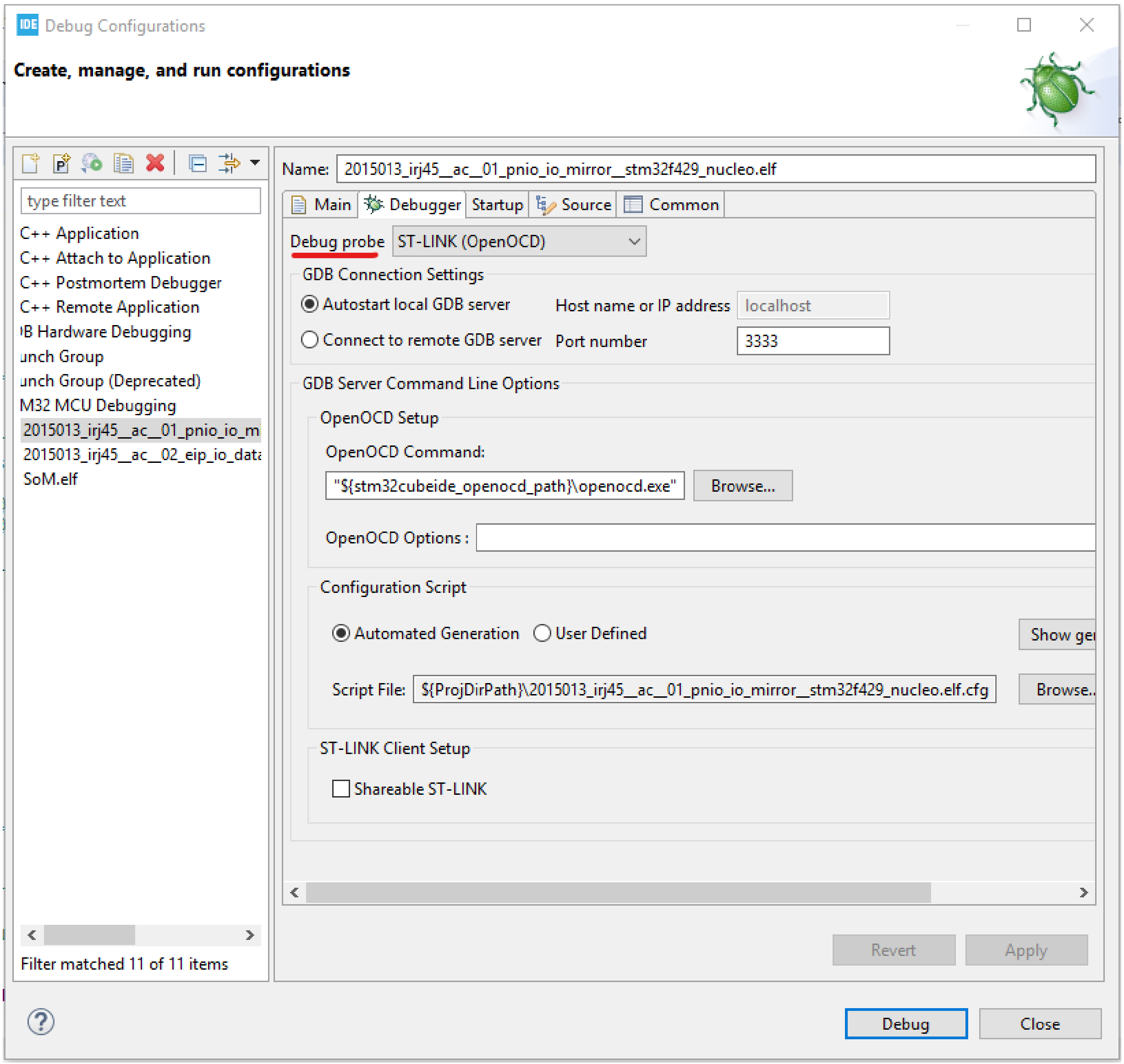


Figure 6 Initiate debug session

⚠ Debug configuration must be edited regarding debug probe for proper function.

Please configure the debug probe as shown (Figure 7 Configure debug session) to ST-LINK (OpenOCD). When prompted allow firewall exception to be configured.



Debug Configurations

Create, manage, and run configurations

type filter text

- C++ Application
- C++ Attach to Application
- C++ Postmortem Debugger
- C++ Remote Application
- 'B Hardware Debugging
- unch Group
- unch Group (Deprecated)
- M32 MCU Debugging
- 2015013_irj45__ac__01_pnio_io_m
- 2015013_irj45__ac__02_eip_io_data
- SoM.elf

Filter matched 11 of 11 items

Name: 2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo.elf

Debug probe: ST-LINK (OpenOCD)

GDB Connection Settings

- ● Autostart local GDB server — Host name or IP address: localhost
- ○ Connect to remote GDB server — Port number: 3333

GDB Server Command Line Options

OpenOCD Setup

OpenOCD Command: "${stm32cubeide_openocd_path}\openocd.exe"

OpenOCD Options :

Configuration Script

● Automated Generation ○ User Defined

Script File: ${ProjDirPath}\2015013_irj45__ac__01_pnio_io_mirror__stm32f429_nucleo.elf.cfg

ST-LINK Client Setup

□ Shareable ST-LINK

Figure 7 Configure debug session

After initiating the debug session, the “Debug perspective” will be shown, where the application can be started by “Resuming” execution.

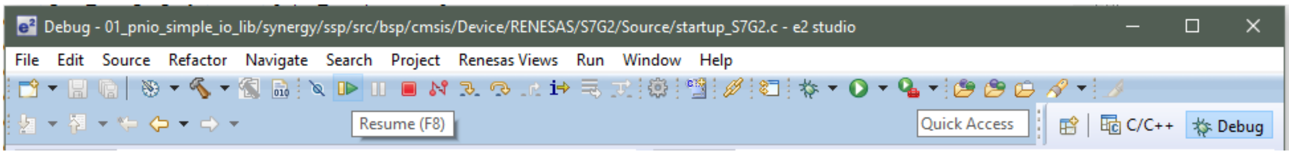
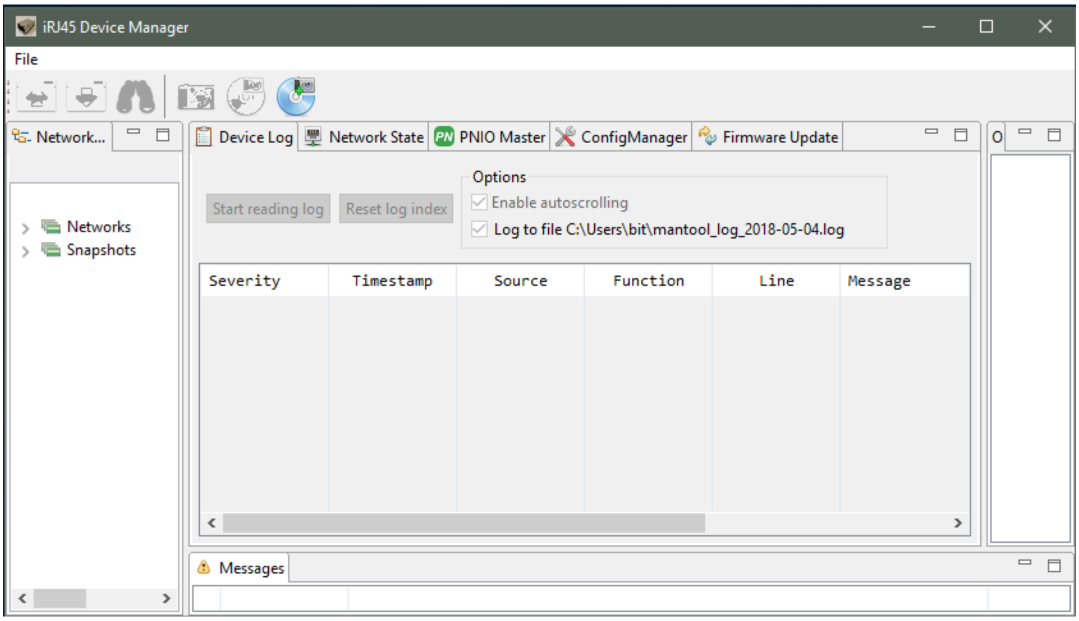

Debug - 01_pnio_simple_io_lib/synergy/ssp/src/bsp/cmsis/Device/RENESAS/S7G2/Source/startup_S7G2.c - e2 studio
File
Edit
Source
Refactor
Navigate
Search
Project
Renesas Views
Run
Window
Help
Resume (F8)
Quick Access
C/C++
Debug

Figure 8 IDE Toolbox

⚠ If the communication module was previously started up using another application, a manual reboot will be required of the module (reset button on the Arduino shield).

4 Management Tool

The Management Tool allows development related configuration and management of the SoM application. This management is based on a UDP broadcast communication. Thus, it works independently from IP settings of the management PC and the SoM.



Options
☑ Enable autoscrolling
☑ Log to file C:\Users\bit\mantool_log_2018-05-04.log

Figure 9 Management Tool main window

This tool is organized in panels. The “Network Navigator” shows a list of available networks. The panel “Messages” shows information regarding actions. The panel “Outline” shows additional information depending on the selected function panel.

Following function panels are available:

Panel	Function
Device Log	shows log messages of the running application from both communication controller (CC) and application controller (AC).
Network state	Shows link state of the available network interfaces of the SoM
PNIO Master	Provides simple PNIO master functionality.
Config Manager	Provides access to the config manager variables of the SoM.
Firmware Update	Allows update of the firmware in the SoM.

4.1 Device Detection

At first a communication needs to establish with the SoM. Thus, connect the SoM to the network. Between the management PC and the SoM a network connection must be possible.

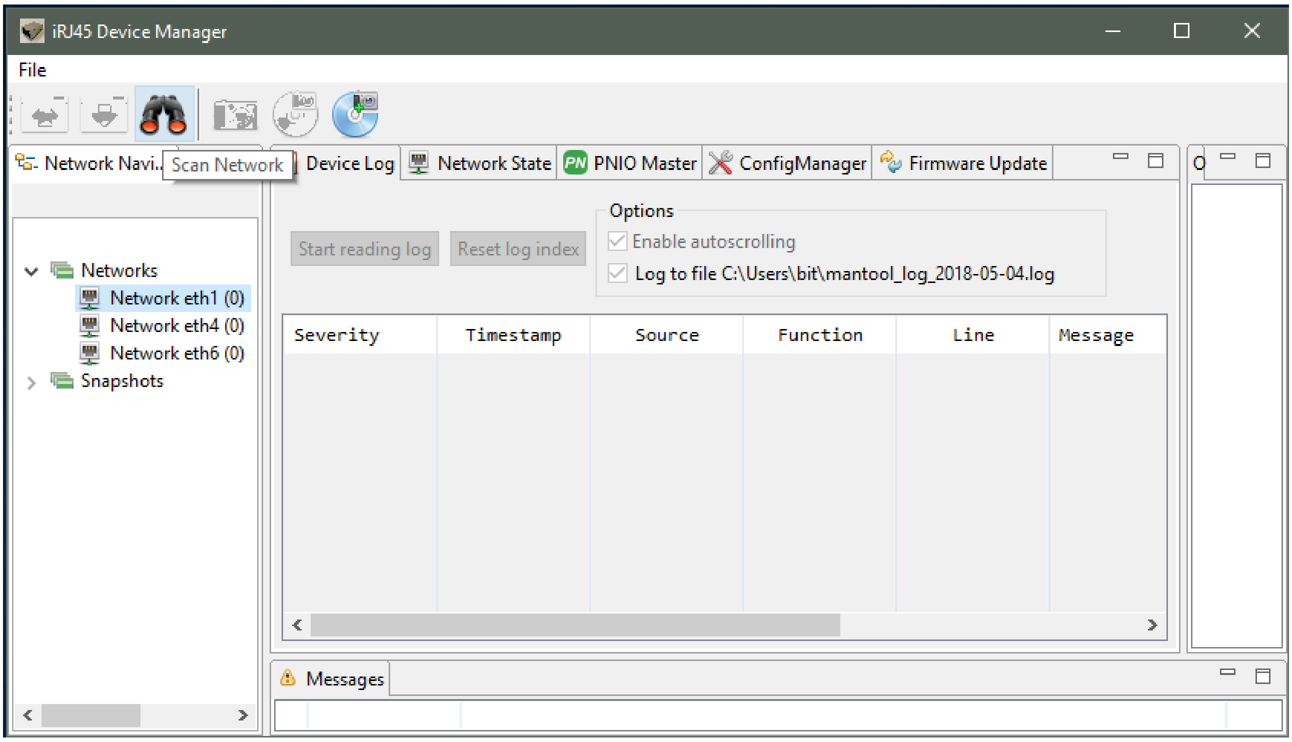
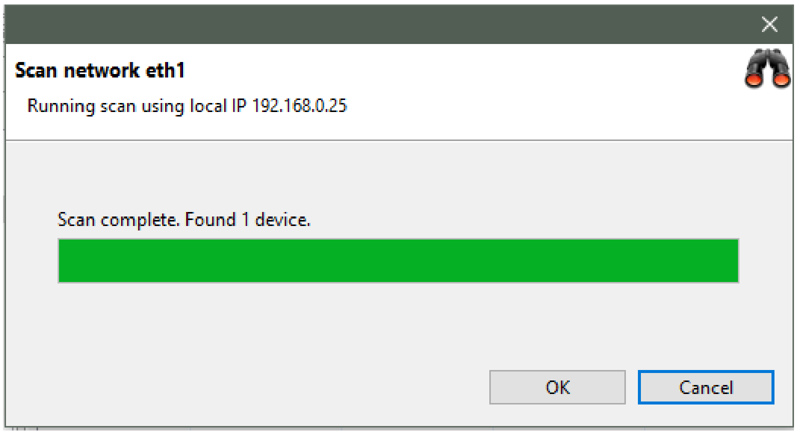


Figure 10 Managment Tool Network scan

To communicate with the SoM, at first open the “Networks” list in the “Network Navigator”. Choose the network interface where the SoM is reachable. The select the “Scan Network” button in the toolbar.

The following dialog appears and 1 found device will be reported:

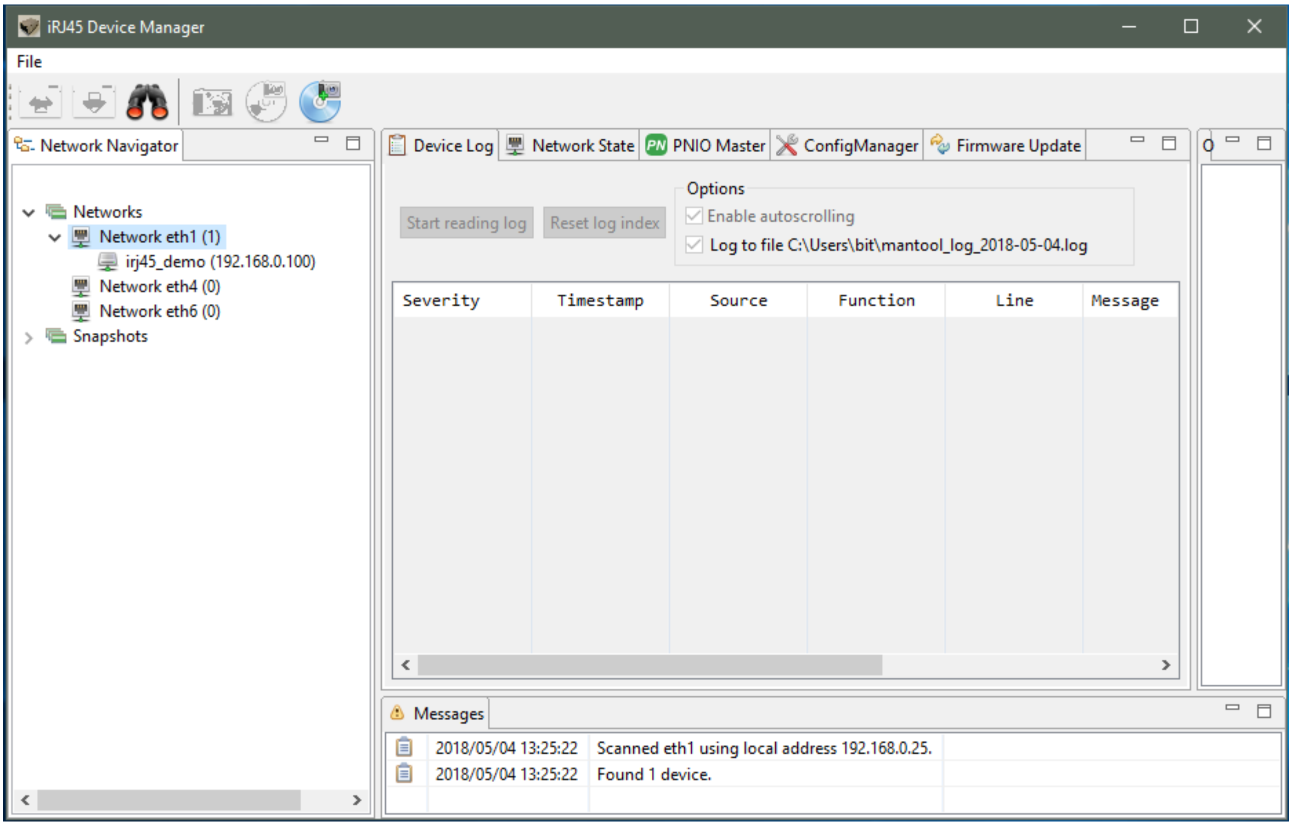

Scan network eth1

Running scan using local IP 192.168.0.25

Scan complete. Found 1 device.

Figure 11 Scan Network dialog

As a result, a new SoM will be shown in the „Network Navigator“ within the scanned network.



☑ Enable autoscrolling
☑ Log to file C:\Users\bit\mantool_log_2018-05-04.log

```
2018/05/04 13:25:22   Scanned eth1 using local address 192.168.0.25.
2018/05/04 13:25:22   Found 1 device.
```

Figure 12 Management Tool with found SoM

Please select the newly found SoM for further steps.

4.2 Logging

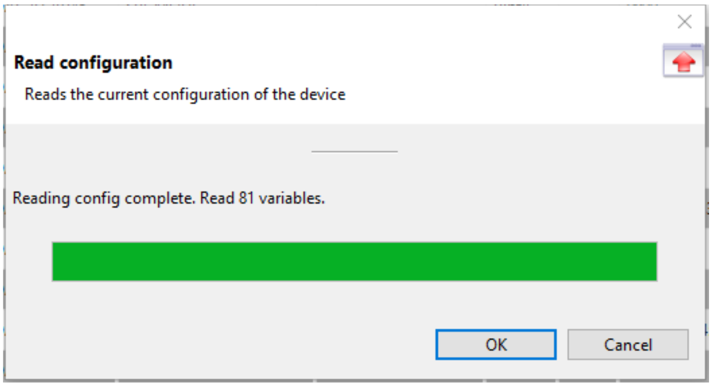
With the selected SoM and within the “Device Log” function panel, it now is possible to read the logging buffer using “Start reading log”. For the demo application, it shows both the log messages from the communication controller (SoM) and the application controller (STM32F4xx). Those can be distinguished by the “Source” column, which either shows “CC” ore “AC”. A successful started application reports a successful initialization of PROFINET:

```
[I|goal_miMctcLoop:499] running appl_setup
[I|goal_pnioNewAc:369] PROFINET Application Core successfully started
[I|appl_setup:226] Initializing device structure
[I|appl_setup:275] PROFINET ready
[I|appl_setup:281] Configuring DD
[I|appl_setup:309] DD ready
```

4.3 Config Manager / IP Configuration

This panel provides access to the config manager variables of the SoM (volatile and nonvolatile stored configuration variables).

To read a list of all variables, select the “Read configuration” button.



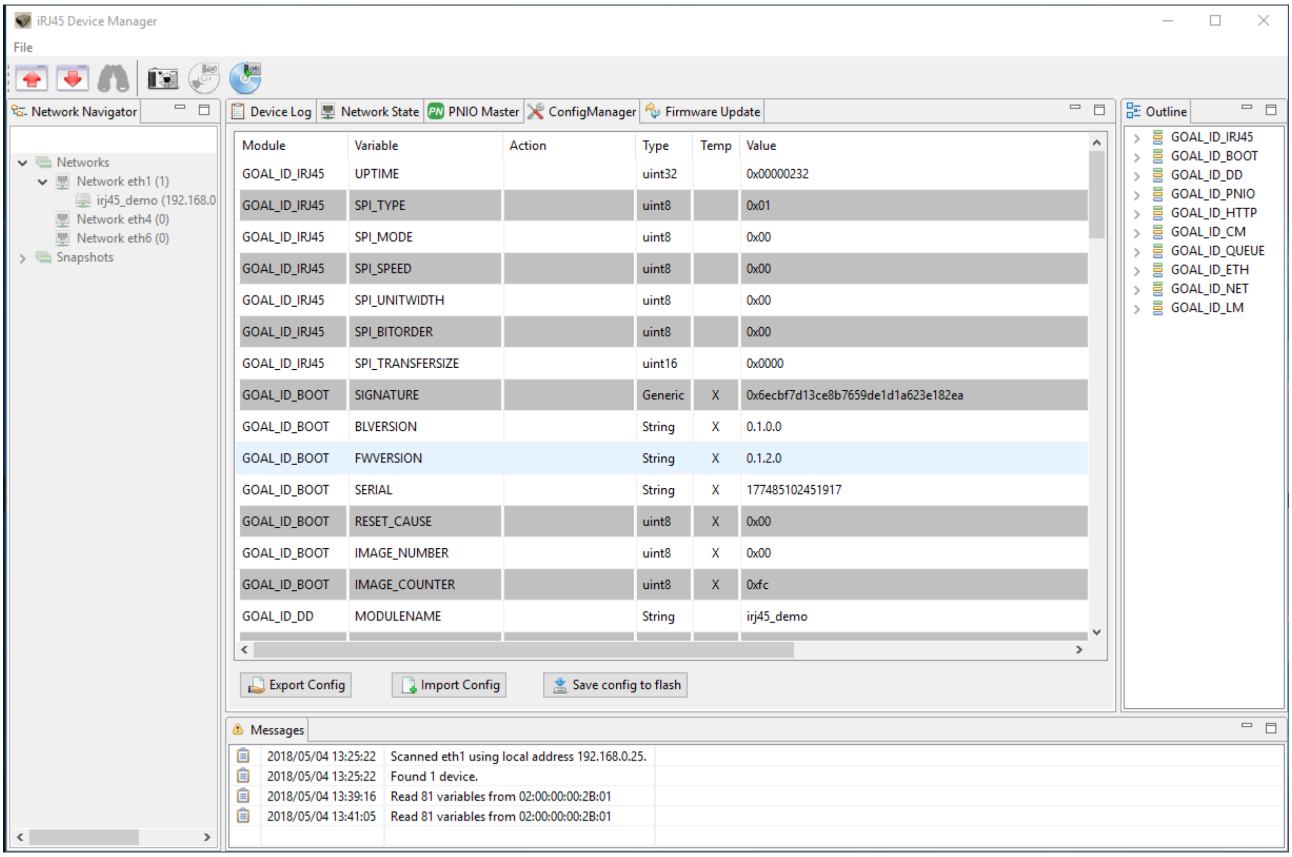
Read configuration

Reads the current configuration of the device

Reading config complete. Read 81 variables.

Figure 13 Management Tool Read configuration

As a result, all variables with value are shown.



Module	Variable	Action	Type	Temp	Value
GOAL_ID_IRJ45	UPTIME		uint32		0x00000232
GOAL_ID_IRJ45	SPI_TYPE		uint8		0x01
GOAL_ID_IRJ45	SPI_MODE		uint8		0x00
GOAL_ID_IRJ45	SPI_SPEED		uint8		0x00
GOAL_ID_IRJ45	SPI_UNITWIDTH		uint8		0x00
GOAL_ID_IRJ45	SPI_BITORDER		uint8		0x00
GOAL_ID_IRJ45	SPI_TRANSFERSIZE		uint16		0x0000
GOAL_ID_BOOT	SIGNATURE		Generic	X	0x6ecbf7d13ce8b7659de1d1a623e182ea
GOAL_ID_BOOT	BLVERSION		String	X	0.1.0.0
GOAL_ID_BOOT	FWVERSION		String	X	0.1.2.0
GOAL_ID_BOOT	SERIAL		String	X	177485102451917
GOAL_ID_BOOT	RESET_CAUSE		uint8	X	0x00
GOAL_ID_BOOT	IMAGE_NUMBER		uint8	X	0x00
GOAL_ID_BOOT	IMAGE_COUNTER		uint8	X	0xfc
GOAL_ID_DD	MODULENAME		String		irj45_demo

```
2018/05/04 13:25:22   Scanned eth1 using local address 192.168.0.25.
2018/05/04 13:25:22   Found 1 device.
2018/05/04 13:39:16   Read 81 variables from 02:00:00:00:2B:01
2018/05/04 13:41:05   Read 81 variables from 02:00:00:00:2B:01
```

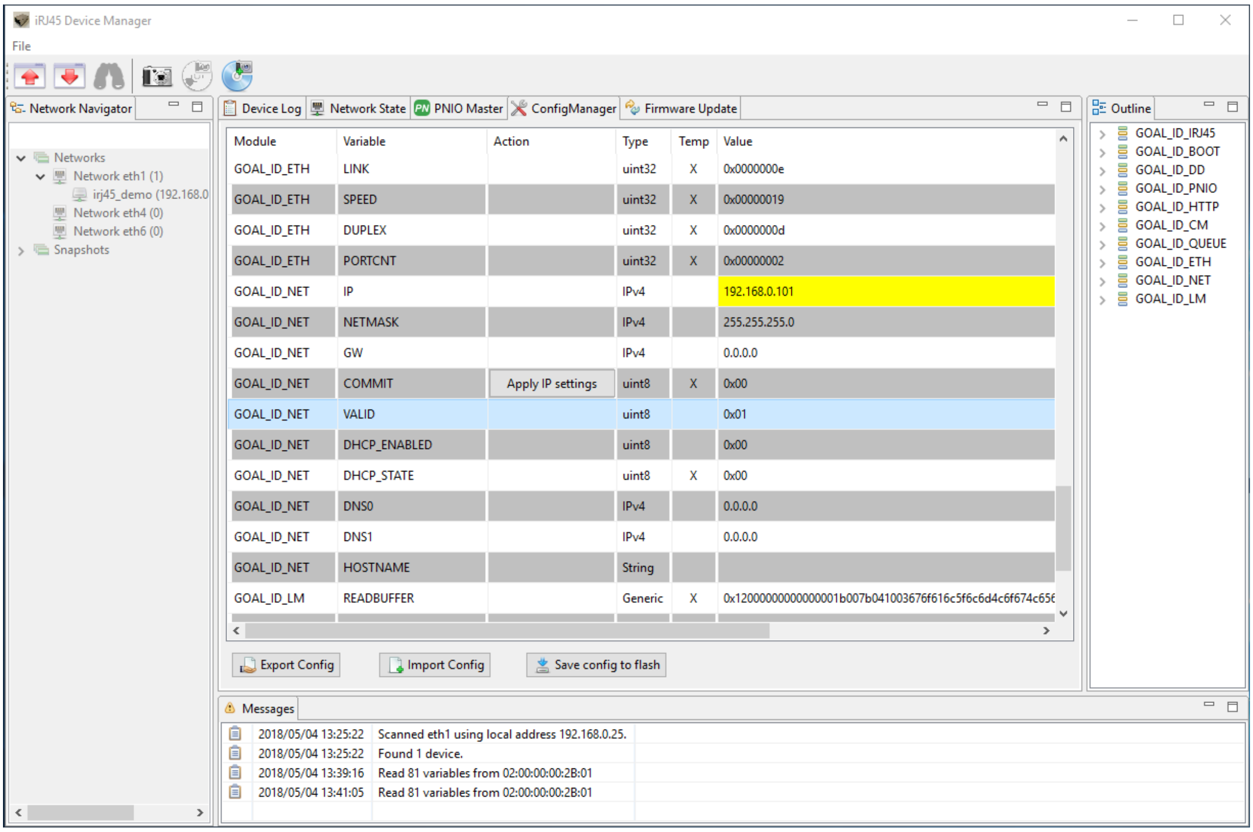
Figure 14 Management Tool Configuration Manager

To communicate with the SoM, the IP address of the SoM must be within the same IP network as the IP address of the Management PC IP address. Thus, chose a valid IP address and configure the SoM accordingly.

To configure an IP address, navigate to the variables of the "Module" GOAL_ID_NET. There it is possible to configure IP, NETMASK and GW. Modify required values. Make sure the variable

“VALID” is set to 1.

The Management Tool will show locally modified variables with a yellow highlight.



Module	Variable	Action	Type	Temp	Value
GOAL_ID_ETH	LINK		uint32	X	0x0000000e
GOAL_ID_ETH	SPEED		uint32	X	0x00000019
GOAL_ID_ETH	DUPLEX		uint32	X	0x0000000d
GOAL_ID_ETH	PORTCNT		uint32	X	0x00000002
GOAL_ID_NET	IP		IPv4		192.168.0.101
GOAL_ID_NET	NETMASK		IPv4		255.255.255.0
GOAL_ID_NET	GW		IPv4		0.0.0.0
GOAL_ID_NET	COMMIT	Apply IP settings	uint8	X	0x00
GOAL_ID_NET	VALID		uint8		0x01
GOAL_ID_NET	DHCP_ENABLED		uint8		0x00
GOAL_ID_NET	DHCP_STATE		uint8	X	0x00
GOAL_ID_NET	DNS0		IPv4		0.0.0.0
GOAL_ID_NET	DNS1		IPv4		0.0.0.0
GOAL_ID_NET	HOSTNAME		String		
GOAL_ID_LM	READBUFFER		Generic	X	0x1200000000000001b007b041003676f616c5f6c6d4c6f674c656

```
2018/05/04 13:25:22   Scanned eth1 using local address 192.168.0.25.
2018/05/04 13:25:22   Found 1 device.
2018/05/04 13:39:16   Read 81 variables from 02:00:00:00:2B:01
2018/05/04 13:41:05   Read 81 variables from 02:00:00:00:2B:01
```

Figure 15 Management Tool with modified variables

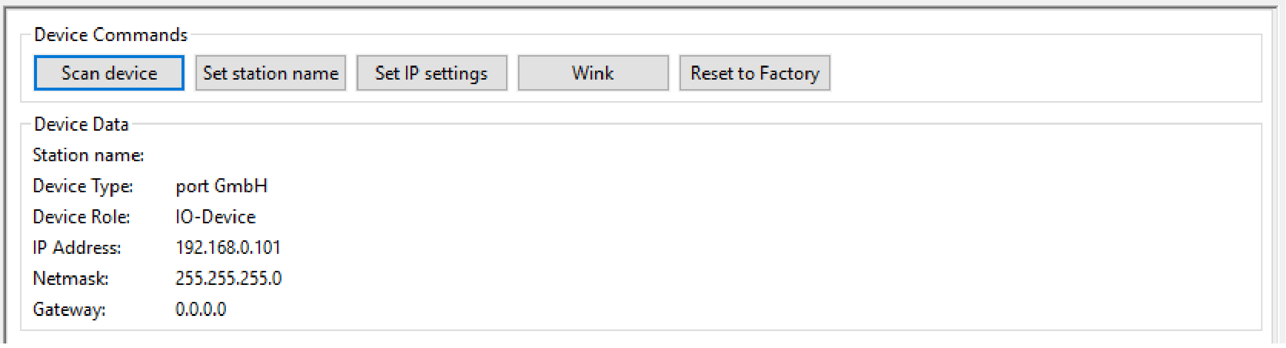
Those locally modified variables are downloaded to the SoM using the “Write configuration” button in the toolbar. When prompted if changed values shall be written, answer “Yes”. Afterwards the locally modified values are transferred to the SoM, where there are only modified in RAM. make changes permanently, answer the following prompt with “Yes”. Modified IP settings are applicated after restart of the system (power cycle the STM32F4xx/SoM).

5 Examples

5.1 01_pnio_simple_io

Please start the example “01_pnio_simple_io” according to the previous description.

To establish a PROFINET communication, at first the SoM must be selected in the “Network Navigator”. Then select the PNIO Master function panel. At first use “Scan device” to detect the PROFINET device.

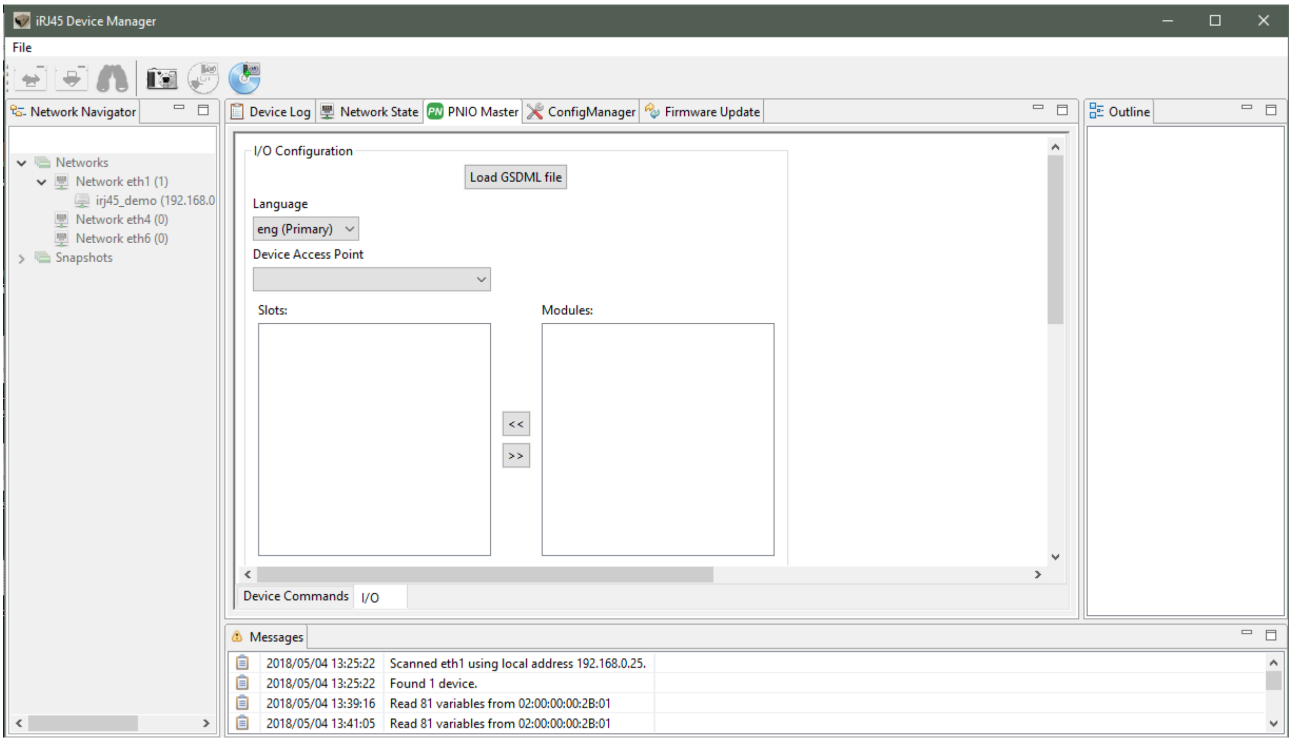


Station name:
Device Type: port GmbH
Device Role: IO-Device
IP Address: 192.168.0.101
Netmask: 255.255.255.0
Gateway: 0.0.0.0

Figure 16 Management PROFINET master

Use the “Wink” command to identify the connected SoM, which will be shown with a flashing “LED1” on the Arduino shield.

To establish a cyclic PROFINET communication use the I/O panel of the PNIO Master.



```
2018/05/04 13:25:22   Scanned eth1 using local address 192.168.0.25.
2018/05/04 13:25:22   Found 1 device.
2018/05/04 13:39:16   Read 81 variables from 02:00:00:00:2B:01
2018/05/04 13:41:05   Read 81 variables from 02:00:00:00:2B:01
```

Figure 17 Management Tool PNIO Master I/O

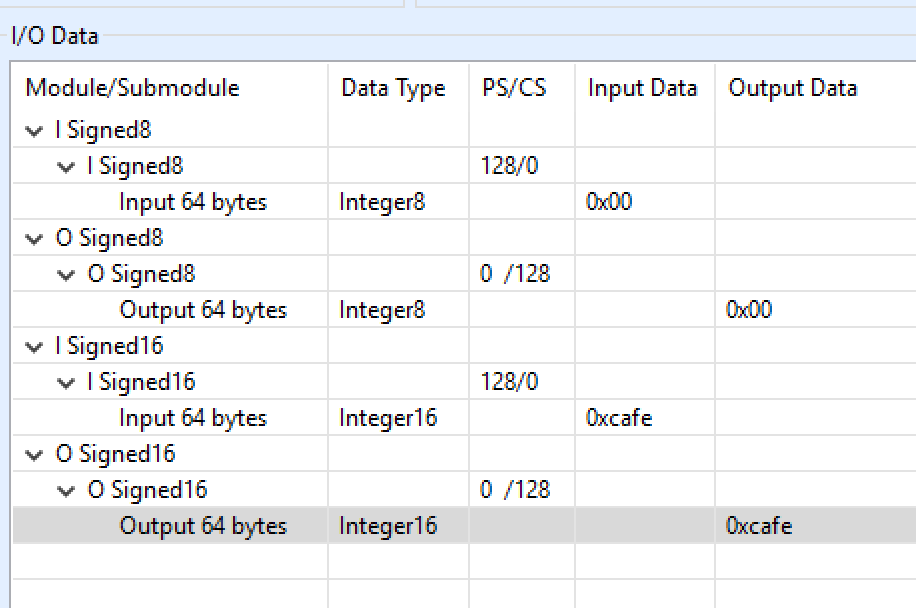
To continue, load the GSDML file provided with the distribution, located in “goal\appl\2015013_irj45\ac\gsdml\GSDML-V2.32-portGmbH-irj45-20180810.xml".

In the selector “Device Access Point” select “2-port Device”.

Afterwards press the “Connect” button. This button initiated a cyclic PROFINET communication.

The example application on the application controller will mirror the output data to the input data. I/O data can be manipulated and monitored in the I/O Data table. Beside that if a connection is established, the “LED1” Led on the Arduino shield will be enabled.

Process data can be monitored and manipulated using the “IO Data” panel.



I/O Data

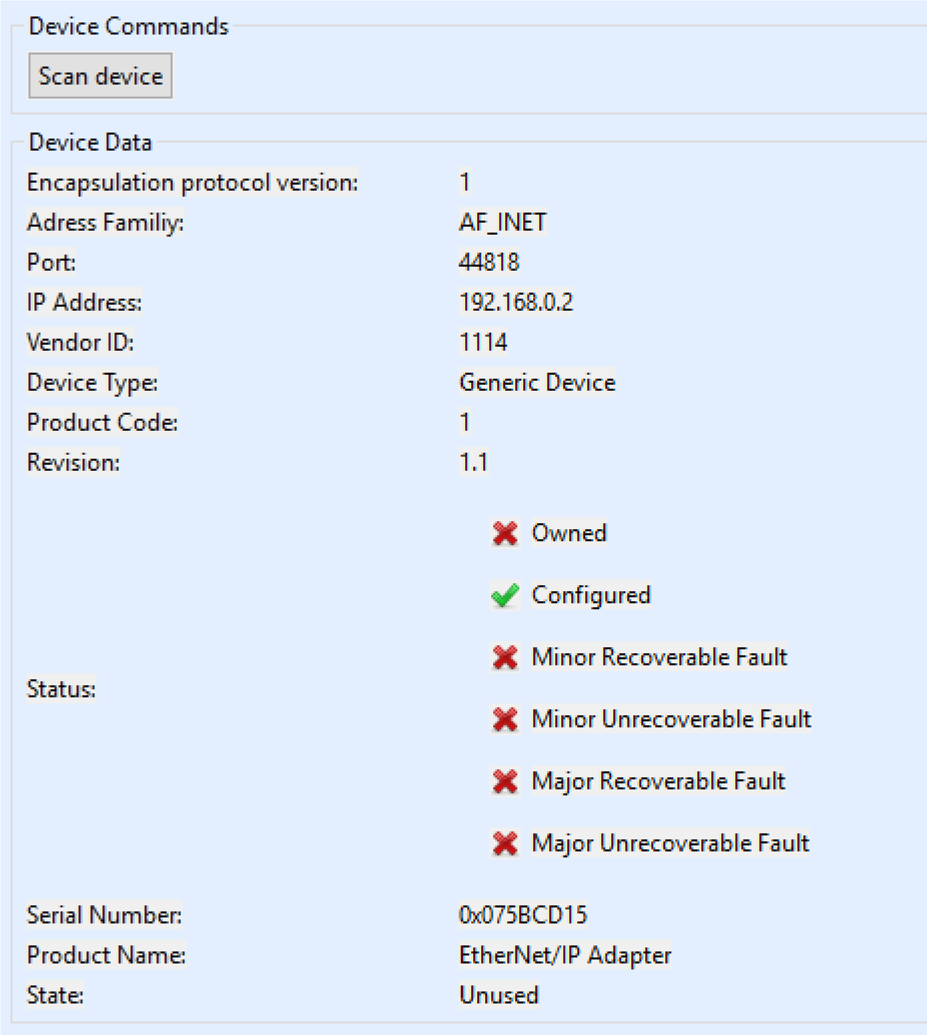
Module/Submodule	Data Type	PS/CS	Input Data	Output Data
I Signed8				
I Signed8		128/0		
Input 64 bytes	Integer8		0x00	
O Signed8				
O Signed8		0 /128		
Output 64 bytes	Integer8			0x00
I Signed16				
I Signed16		128/0		
Input 64 bytes	Integer16		0xcafe	
O Signed16				
O Signed16		0 /128		
Output 64 bytes	Integer16			0xcafe

Figure 18 PNIO IO Data panel

5.2 02_eip_io_data

Please start the example “02_eip_io_data” according to the previous description.

To establish an Ethernet/IP communication, at first the SoM must be selected in the “Network Navigator”. Then select the “EtherNet/IP Master” function panel. At first use “Scan device” to detect the EtherNet/IP device.



Device Commands

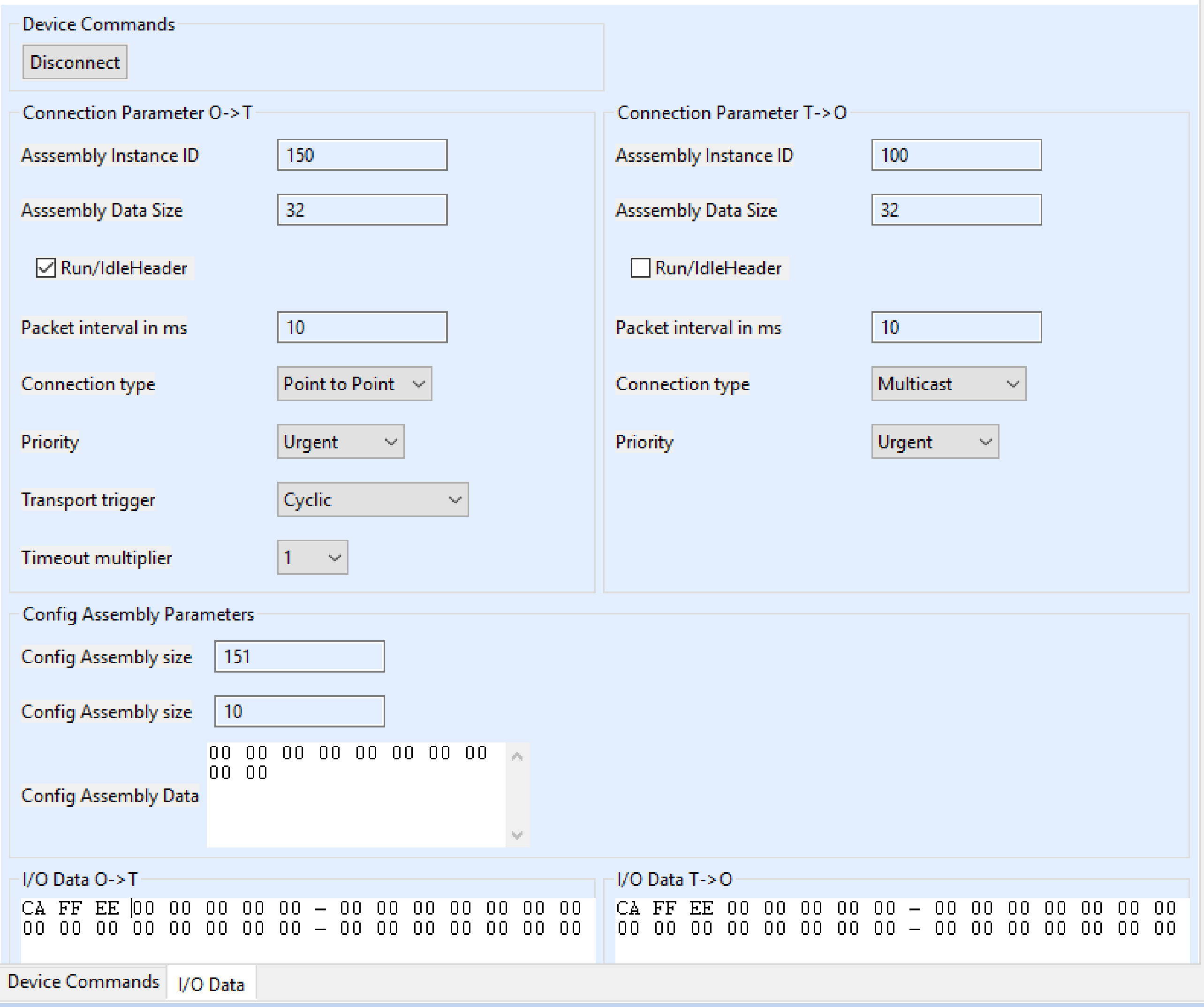
Device Data

Encapsulation protocol version:	1
Adress Familiy:	AF_INET
Port:	44818
IP Address:	192.168.0.2
Vendor ID:	1114
Device Type:	Generic Device
Product Code:	1
Revision:	1.1
Status:	✗ Owned ✓ Configured ✗ Minor Recoverable Fault ✗ Minor Unrecoverable Fault ✗ Major Recoverable Fault ✗ Major Unrecoverable Fault
Serial Number:	0x075BCD15
Product Name:	EtherNet/IP Adapter
State:	Unused

Figure 19 Management EtherNet/IP master

To establish an EtherNet/IP communication with the device, IP settings must be set according to the previous description. You can verify the current settings using the Management Tool.

To establish a cyclic EtherNet/IP communication use the I/O panel of the Master.



Connection Parameter O->T

Assembly Instance ID: 150
Asssembly Data Size: 32
☑ Run/IdleHeader
Packet interval in ms: 10
Connection type: Point to Point
Priority: Urgent
Transport trigger: Cyclic
Timeout multiplier: 1

Connection Parameter T->O

Asssembly Instance ID: 100
Asssembly Data Size: 32
□ Run/IdleHeader
Packet interval in ms: 10
Connection type: Multicast
Priority: Urgent

Config Assembly Parameters

Config Assembly size: 151
Config Assembly size: 10
Config Assembly Data:

```
00 00 00 00 00 00 00 00
00 00
```

I/O Data O->T

```
CA FF EE 00 00 00 00 00 - 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00
```

I/O Data T->O

```
CA FF EE 00 00 00 00 00 - 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00
```

Figure 20 Management Tool EtherNet/IP Master I/O

Default settings are compatible with the example. Press the “Connect” button. This button initiated a cyclic EtherNet/IP communication.

The example application on the application controller will mirror the output data to the input data. I/O data can be manipulated and monitored in the I/O Data tables. Beside that if a connection is established, the “LED1” and “LED2” Leds on the Arduino shield will both be green.

5.3 01_udp_receive

Please start the example “01_udp_receive” according to the previous description.

This example demonstrated networking from application controller. It provides a server, listening on IP address 192.168.0.25 and port 1234 and 1235. It will mirror any data received on those ports using UDP.

Please note that this example overwrites any taken IP settings for demonstration purpose.

5.4 01_http_get

Please start the example “01_http_get” according to the previous description.

Once started this example will provide simple web server functionality. It will deliver a simple web site showing the version number of the SoM.

Please consider the correct IP settings. Those can always be checked and manipulated using the management tool.