User Manual

TPS-1

TPS-1 Low-cost Solution Kit - Getting Started with RX66T

YCONNECT-IT-TPS-1L

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Technology Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

R21UT0247ED0102, Rev. 1.02
12 September 2023

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics.

8. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti- crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

9. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.

10. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

11. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.

13. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

    (Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority- owned subsidiaries.

    (Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.

   The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.

   The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

   The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

Regional Information

Some information contained in this document may vary from country to country. Before using any Renesas Electronics product in your application, please contact the Renesas Electronics office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

Visit

http://www.renesas.com

to get in contact with your regional representatives and distributors.

Preface

Readers This manual is intended for users who want to understand the functions of the concerned microcontrollers.

Purpose This manual presents the hardware manual for the concerned microcontrollers.

Organisation This system specification describes the following sections:

- Pin function
- CPU function
- Internal peripheral function

Module instances These microcontrollers may contain several instances of a dedicated module. In general, the different instances of such modules are identified by the index “n”, where “n” counts from 0 to the number of instances minus one.

Legend Symbols and notation are used as follows:

• Weight in data notation:	Left is high order column, right is low order column
• Active low notation:	$\overline{xxx}$ (pin or signal name is over-scored) or /xxx (slash before signal name) or _xxx
• Memory map address:	High order at high stage and low order at low stage

Note Additional remark or tip

Caution Item deserving extra attention

Numeric notation

Binary:	xxxx or xxxB
Decimal:	xxxx
Hexadecimal	xxxxH or 0x xxxx

Numeric prefixes representing powers of 2 (address space, memory capacity):

K (kilo):	$2^{10} = 1024$
M (mega):	$2^{20} = 1024^2 = 1,048,576$
G (giga):	$2^{30} = 1024^3 = 1,073,741,824$

Register contents X, x = don’t care

Diagrams Block diagrams do not necessarily show the exact wiring in hardware but the functional structure. Timing diagrams are for functional explanation purposes only, without any relevance to the real hardware implementation.

How to Use This Manual

(1) Purpose and Target Readers

This manual is designed to provide the user with an understanding of the set up of the TPS-1 Solution Kit. It is intended for users evaluating the TPS-1. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual. The manual comprises a step-by-step description of the installation and initial usage of the tools, that are included in the TPS-1 solution kit package.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the TPS-1 product. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Electronics Web site.

Document Type	Description	Document Title	Document No.
Data Sheet	Hardware overview and electrical characteristics	TPS-1 Datasheet	R19DS0069EJ0107
User's manual for Hardware	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description. Note: Refer to the application notes for details on using peripheral functions.	TPS-1 User's manual for Hardware	R19UH0081ED0110
User's manual for Software	Description of CPU instruction set	TPS-1 Series User's manual for Software	not applicable
User's manual for development environment	Operation instructions for hard- and software tools	TPS-1 Low-cost Solution Kit User's manual Hardware	R21UT0239ED0103
	Description of tool installation and initial set up	TPS-1 Low-cost Solution Kit – Getting started	This document

(2) List of Abbreviations and Acronyms

Abbreviation	Full Form
CD	Compact Disc
CPU	Central Processing Unit
MAC	Media Access Control
PC	Personal Computer
TFTP	Trivial File Transfer Protocol
TPS-1	PROFINET I/O device chip
UART	Universal Asynchronous Receiver / Transmitter

All trademarks and registered trademarks are the property of their respective owners.

(3) List of related Documents

Document Title	Description
R19UH0081ED0110	User Manual TPS-1 device
TPS_Update_Manual_EN.pdf	Application Note for TPS-1 firmware update
PROFINET Configurator - Quick Start Guide.pdf	Quick start guide for the PROFINET Configurator tool
R21UT0239ED0103	User Manual(Hardware) TPS-1 Low-cost Solution Kit
R21UT0236ED0106	Getting started for TPS-1 Low-cost Solution Kit with RX630
R21UT0237ED0105	Getting started for TPS-1 Low-cost Solution Kit with RX231
R21UT0238ED0105	Getting started for TPS-1 Low-cost Solution Kit with Synergy S7G2
R21UT0243ED0104	Getting started for TPS-1 Low-cost Solution Kit in Parallel IO-Mode

Table of Contents

Chapter 1 Introduction 10
1.1 Software Installation 10
Chapter 2 TPS-1 Set Up 11
2.1 Hardware Connections 11
2.1.1 Connections between PC and TPS-1 board 11
2.1.2 Connections between TPS-1 and RX66T 12
2.2 General Settings 13
2.3 Configuration Download to TPS-1 15
Chapter 3 RX66T Host Processor Connection 20
3.1 Loading and Building Sample Project for RX66T 20
3.2 Changing Host Communication Mode 27
Chapter 4 PROFINET Configuration 29
Chapter 5 Firmware Image Replacement on TPS-1 36
5.1 Generate Image File 36
5.2 TPS-1 Firmware Update 37
5.2.1 TPS Firmware Updater preferences setting 37
5.2.2 Transfer the Updater Image 37
5.2.3 Transfer the Firmware Image 38
5.2.4 Finalize the Firmware Update 38
5.2.5 Netnames+ Tool 40
Chapter 6 Erasing the Flash Memory 41
Revision History 44

Chapter 1 Introduction

This manual describes the installation and initial set up of the TPS-1 Low-cost Solution Kit. All required steps that need to be taken to run the demonstration program that comes with the TPS-1 Solution Kit will be described.

This manual does not replace the manuals that come together with the various software components. These manuals are still required to study when functions beyond the demonstration program are needed.

Some preparations are required, before the setup of the TPS-1 Low-cost Solution Kit can be tackled; the preparations are

- the installation of various software programs
- the creation of a fixed directory structure

Note: The instructions and screen shots in this manual are based on tool versions that were up to date at the time when the manual was prepared. Newer versions may exhibit slightly different behaviour. Please consider this before downloading and installing newer versions of the TPS-1 development tool kit.

1.1 Software Installation

As a first step the following tools need to be installed

- from the Phoenix Contact Software CD
  - TPS Configurator
  - TPS FW Updater
  - PROFINET Configurator
  - PROFINET Smart Control
- from the Renesas Electronics CD that comes with the TPS-1 Low-cost Solution Kit
  - the RX66T sample application
- from the Renesas Electronics website
  - e^{2}studio
  - GCC for Renesas RX toolchain integration

The installation of these tools is basically self-explaining, for details with respect to the installation please refer to the related software manuals.

Note: Please be aware of starting the software tools with “Run as administrator” to run them correctly. TPS Configurator, TPS FW Updater and PROFINET Smart Control need access through the firewall in order to communicate to the TPS-1 via a network adapter. Therefore, make sure that access through the firewall is enabled for these programs.

Chapter 2 TPS-1 Set Up

This chapter describes the steps that are required to set up the TPS-1 and its serial flash content properly. The TPS-1 Solution Kit is delivered with a default image of the PROFINET stack in the serial flash for the TPS-1. This default image has no device specific settings like MAC-addresses or host mode configuration.

Note that each board has individual MAC-addresses assigned (printed on a sticker on the board) and the TPS-1 set up process (not only) stores these addresses in serial Flash.

2.1 Hardware Connections

This chapter describes the required hardware connections between the TPS-1 board and the PC as well as the TPS-1 and the RX66T board.

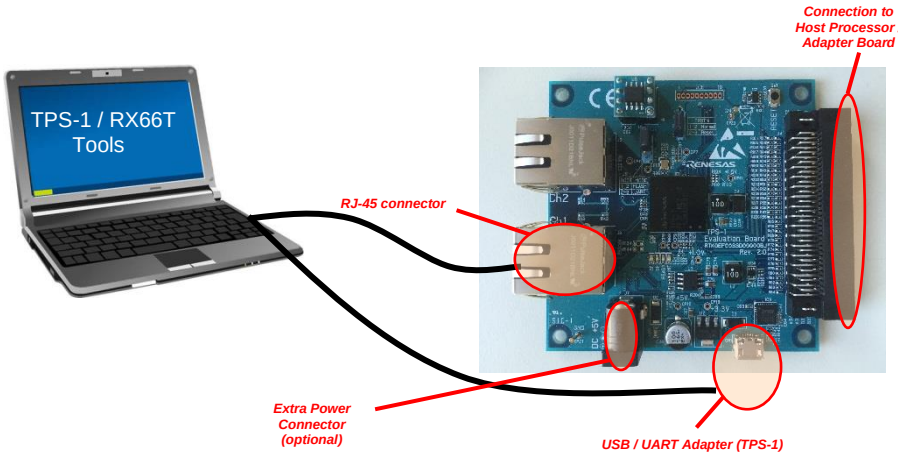
2.1.1 Connections between PC and TPS-1 board

The TPS-1 Solution Kit must be connected to your PC with two cables that are all delivered as part of the kit. These are

- a USB cable for power supply of the board and for serial communication with the TPS-1
- an Ethernet cable for communication with one of the TPS-1 Ethernet ports (connect to connector J3 (Ch1) or J5 (Ch2) on the board)

Figure 2-1 visualizes the required connections.

Figure 2-1: PC Connections for TPS-1 Solution Kit



The external power connector J7 can alternatively be used when hardware is connected to J4 that exceeds the USB power capabilities.

Note: Please be aware that the Windows 7 OS disconnects an Ethernet connection when in state of inactivity. It may therefore be helpful to use an Ethernet switch between PC and TPS-1 Low-cost Solution Kit).

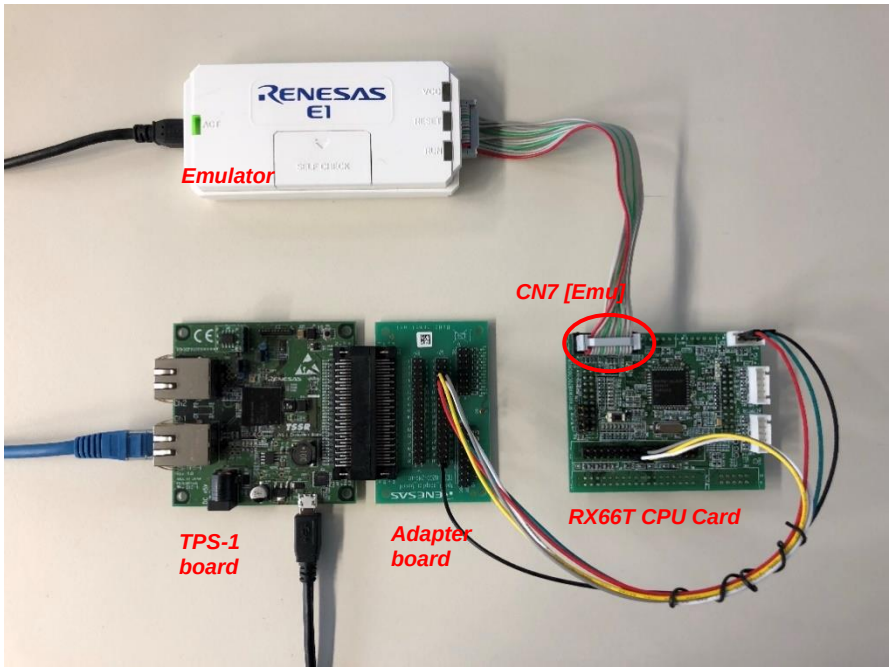
2.1.2 Connections between TPS-1 and RX66T

The TPS-1 Low-cost Solution Kit uses an adapter board to make the serial host interface of the TPS-1 easily accessible. So, plug the TPS-1 board and the adapter board together as shown in Figure 2-2.

In this manual we will use a RX66T CPU card as a platform for the RX66T. The RX66T CPU card (order code RTK0EMX870C00000BJ) is an optional CPU board that's used to mount on 24V Motor Control Evaluation System.

In order to execute and debug code on the RX66T, attach a Renesas E1 debugger to the connector named CN7(Emulator) on the top-left side of the board and connect the E1 debugger via USB to your PC (see Figure 2-2). The debugger can be configured in such a way, that it supplies the RX66T board with power (done in the RX66T software project).

Figure 2-2: TPS-1 and RX66T board set up



Finally, a number of connections between the adapter board of the TPS-1 Low-cost Solution kit and the RX66T CPU card need to be made. To achieve this, we recommend to soldering a 36-pin connector to CNC on the RX66T board. Then the cables delivered with the TPS-1 Low-cost solution kit can be used to establish the serial connection between TPS-1 and RX66T. Connect the pins as described in Table 2-1.

Table 2-1 Serial interface connections between TPS-1 and RX66T

RX66T CPU card		TPS-1 adapter board	
Signal	**Conn. - Pin**	**Signal**	**Conn. - Pin**
GND	CN2 - 4	GND	B - 34
TDX5-SCI	CN2 - 2	HOST_SRXD_IN/GPIO40	B - 7
RDX5-SCI	CN2 - 3	HOST_STXD_OUT/GPIO42	B - 9
SCK5 (PD2_WN_3_GTIOC2B)	CND - 17	HOST_SCLK_IN/GPIO41	B - 8
P90	CNC - 17	HOST_RESET_IN/GPIO38	B - 5
P91	CNC - 18	HOST_SFRM_IN/GPIO39	B - 6

Note: The pin numbers given in Table 2-1 for the TPS-1 adapter board correspond to the print in the board's silkscreen.

2.2 General Settings

As a preparation for working with the TPS-1 a fixed directory structure must be established on your PC. You need a working directory that will be referred to as

[Work Directory]
In the working directory however, several subfolders must be prepared, that will be referred to as

- ***[FW_Updater_Dir]*** Holds files for download using the Firmware Updater program. Copy the following files from the <TPS Stack> folder (located on Phoenix Contact Software CD) to here. The other files that will be placed here are described later.
  - <174_1234_01upd.dat>
  - <TPS_Default_Download_Image_ETH.img>

- ***[TPS Stack]*** Holds target image files that was created from <TPS_Stack.bin> and other files. Copy the following files from folder <TPS Stack> (located on Phoenix Contact Software CD) to here.
  - <hdr.txt>
  - <make_Target_Image.bat>
  - <TPS_Image_Maker.exe>
  - <TPS_Stack_Release.bin>
  - <TPS_Stack_Debug.bin>

- ***[TPS Starter]*** Holds starter file for download over serial interface. Copy the following files from folder <TPS Starter> (located on Phoenix Contact Software CD) to here.
  - <TPS_Starter.s>
  - <TPS_Erase_Flash.s>
  - <TPS_DefaultImageLoader.s>

- ***[TPS Updater ETH]*** Holds updater image created from <TPS_Updater.bin> file. Copy the following files from folder <TPS Updater ETH> (located on Phoenix Contact Software CD) to here.
  - <udphdr.txt>
  - <make_Target_Image.bat>
  - <TPS_Image_Maker.exe>
  - <TPS_Updater.bin>

- ***[TPS Configurator]*** Holds example configurations for TPS-1. Copy the folder <Example Configuration> from </TPS Configurator> directory (located on Phoenix Contact Software CD) to this directory.

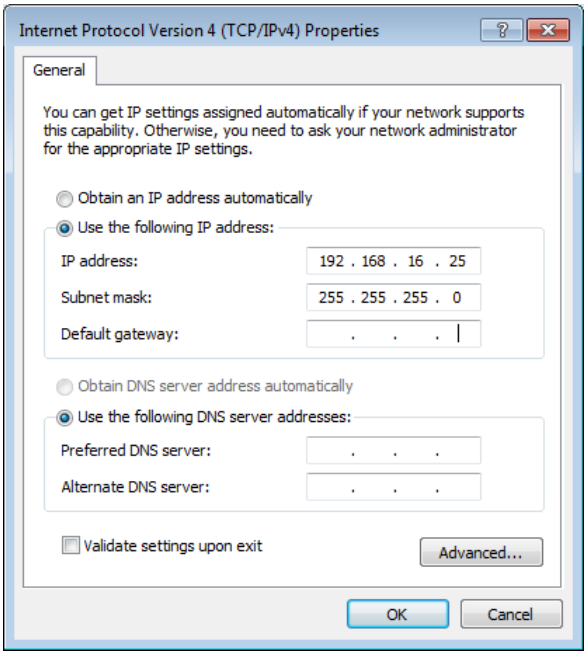
- ***[RX66T_Software]*** Holds example of Host CPU software and TPS-1 API driver. Install the programs from the folder <Sample Software> (located on Renesas CD) to here. Other files that will be placed here are described later.

- ***[PROFINET Configurator]*** Holds example files for PROFINET IO controller and network. Copy complete folder <Example Project> from folder <PROFINET Configurator> (located on Phoenix Contact Software CD) to here.

Additionally, the network adapter of your PC that you use for communication with the TPS-1 must be configured to a manually set, fixed IPv4 address of 192.168.16. XX. e.g., 192.168.16.25. Figure 2-3 illustrates the setting.

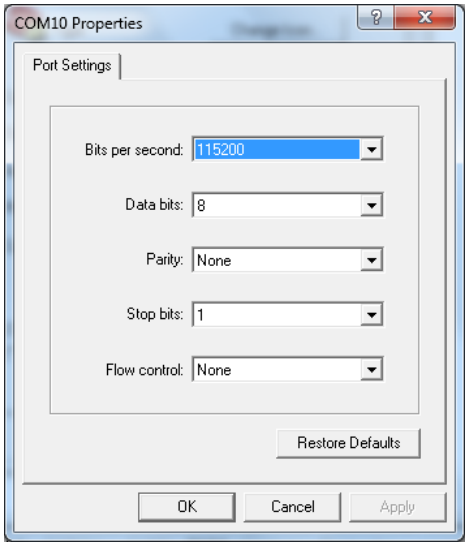
Note: Please note that the usage of this IP-address is not mandatory. However, we will use it as an example through the rest of this document.

Figure 2-3: IP address setting



Next, the serial port of the PC has to be configured to the proper communication parameters for talking with the TPS-1. The UART of the TPS-1 works with 115.2 kbps, 8 bit, no parity, 1 stop bit and no flow control. Throughout this manual we will use Tera Term as an example for a terminal emulation, but this is not mandatory, also the COM port number may change. The screenshot showing the required settings is given in Figure 2-4.

Figure 2-4: Serial port settings on PC



2.3 Configuration Download to TPS-1

After setting the communication parameters, everything is ready for making first contact with the TPS-1. To do this, push the reset button on the TPS-1 Low-cost Solution Kit board. The TPS-1 will then check the content of the serial Flash.

When the serial Flash is empty

If the TPS-1 serial Flash is erased and empty, the TPS-1 will notify via the UART that the Flash is empty. The error code 00000012 in Figure 2-5 indicates an empty Flash.

At that point, you may send the default image of the PROFINET stack to the serial Flash. By pressing “s” start the transfer process the program (Motorola S-Record) TPS_DefaultImageLoader.s.

Figure 2-5: TPS-1 start-up messages with empty serial Flash

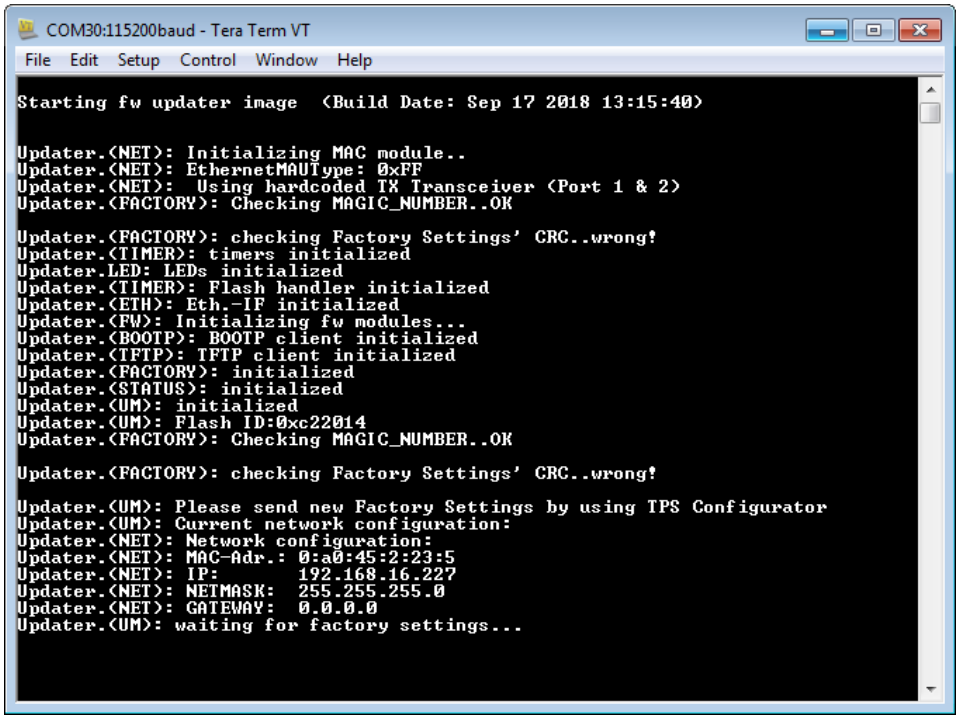
After starting the program with “g” the TPS-1 is waiting for a transfer of the default image of the PROFINET stack via the Ethernet-Interface. At that time, please keep the TPS Firmware Updater open with the preference settings described in Chapter 5.2.1.

When the default image of the PROFINET stack is successfully transferred into the flash, TPS-1 will notify “waiting for the factory settings….”

When the serial Flash contains a default image

On the other hand, if the serial Flash already has the default PROFINET stack image, the TPS-1 will notify via the UART that it has not yet received factory settings respectively the hardware configuration settings. The related output is shown in Figure 2-6.

Figure 2-6: TPS-1 start-up messages from default image



The factory settings like MAC and IP-Addresses or interface configuration, that the TPS-1 is waiting for, are prepared using the TPS-1 Configurator. After starting the TPS-1 Configurator program, several tabs can be selected:

- General settings like interface configuration to host CPU or PROFINET device identification.
- Detailed settings for the selected CPU interface type (tabs are enabled depending on the configuration that was selected in 'General Settings')
- Ethernet settings like MAC- and IP-Addresses

Note: Make sure to start the TPS Configurator program as administrator.

After starting the TPS-Configurator, use the <File> menu (1) to open the <host_interface_serial.xml> file (2) as illustrated in Figure 2-7 and Figure 2-8.

Figure 2-7: Loading a configuration file

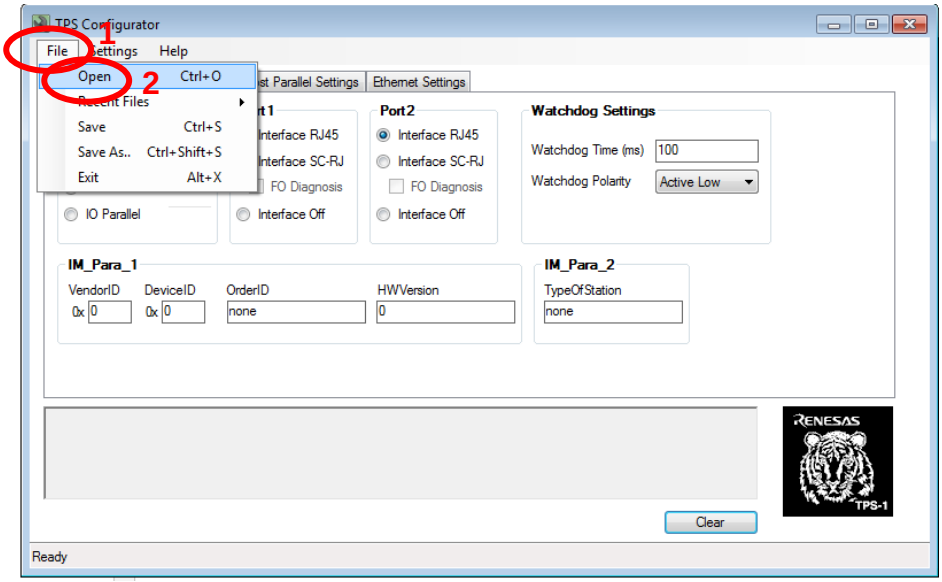
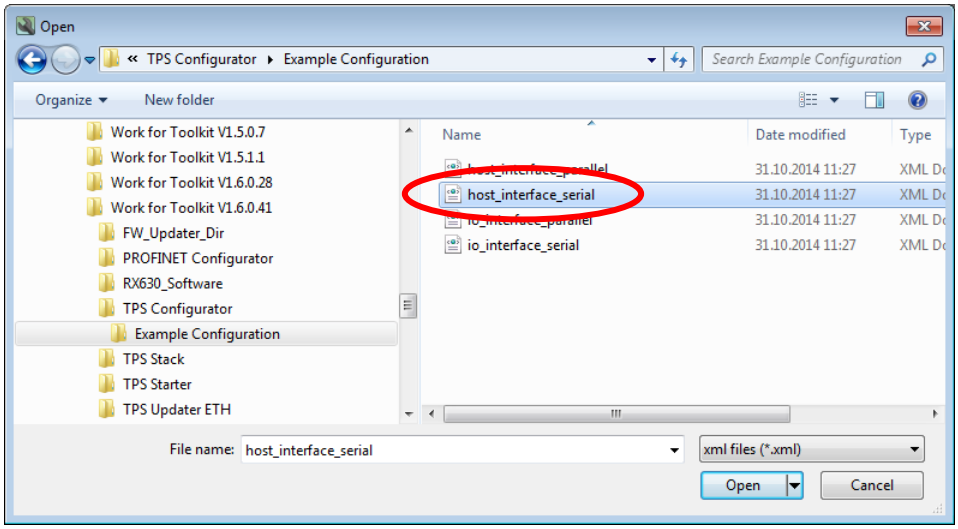


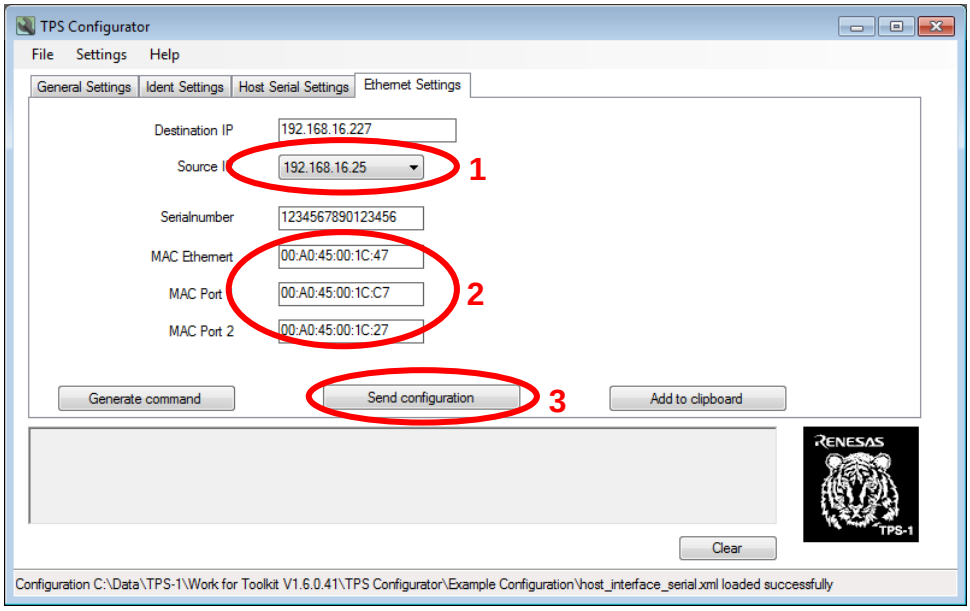
Figure 2-8: Selecting the <host_interface_serial.xml> file



All settings from the <host_interface_serial.xml> file can be used without modifications and you can switch to the <Ethernet settings> tab of the TPS Configurator. Then select the IP address (1) of your network adapter from the drop-down menu and the MAC-addresses (2) from the sticker on your TPS-1 board (different from the ones in the screenshot in Figure 2-9). Now everything needed for the TPS-1 configuration is set up and you can click <Send configuration> (3).

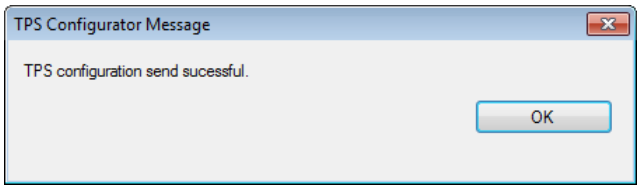
Note: The MAC addresses in Figure 2-9 are only an example. Please use the reserved MAC addresses that are given on the sticker of your TPS-1 Solution Kit board. For the 'Destination IP' address please use **always** '**192.168.16.227**'. Edit the 'Destination IP' address accordingly, if a different address is shown.

Figure 2-9: Ethernet settings in TPS Configurator



The TPS Configurator will inform you with the dialog in Figure 2-10 if the download was successful.

Figure 2-10: Configuration sent confirmation

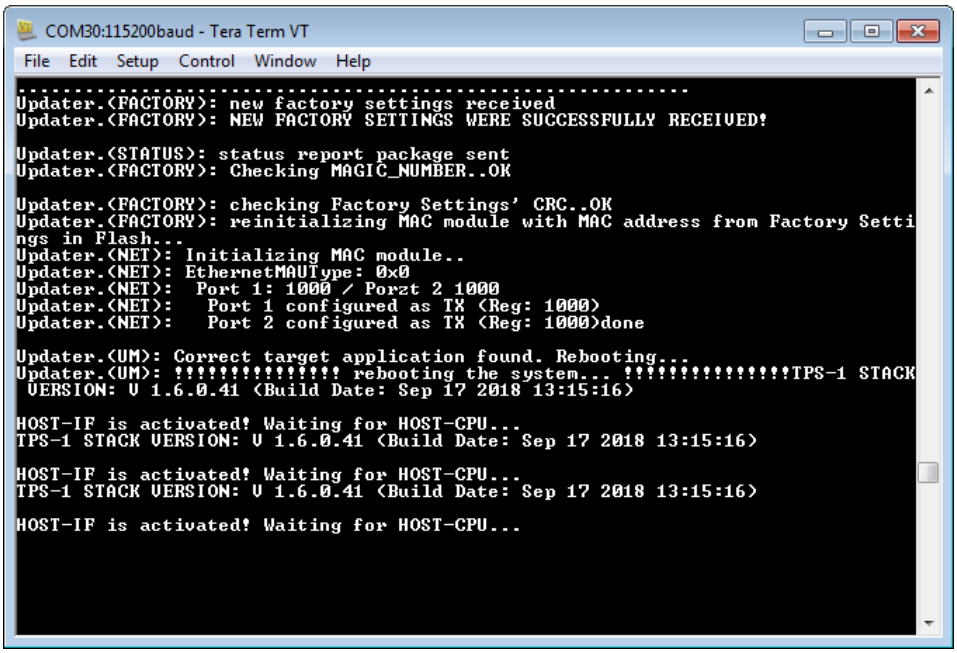


The new configuration is permanently stored in the serial Flash; to change it, please use the TPS Configurator again.

Note: The TPS-1 waits for factory settings for a limited time; it may happen, that the wait is finished by a timeout, before such settings are sent by the TPS Configurator. In this case, please reset the TPS-1 board and observe in the terminal window whether TPS-1 is ready to receive factory settings.

After the new configuration was sent, the TPS-1 reports over the UART, that the new configuration was received and reboots with the new configuration. The boot process is executed until the <Waiting for Host> message occurs – the TPS-1 must now do a handshake with the host CPU before it is ready to start PROFINET communication. Figure 2-11 shows the TPS-1 UART output (this image has been captured with a stack version V1.6.0.41, if you are using another version of TPS-1 stack, it will show up on the UART) after an additional power cycle reset, after the configuration was received.

Figure 2-11: TPS-1 UART output after configuration was received



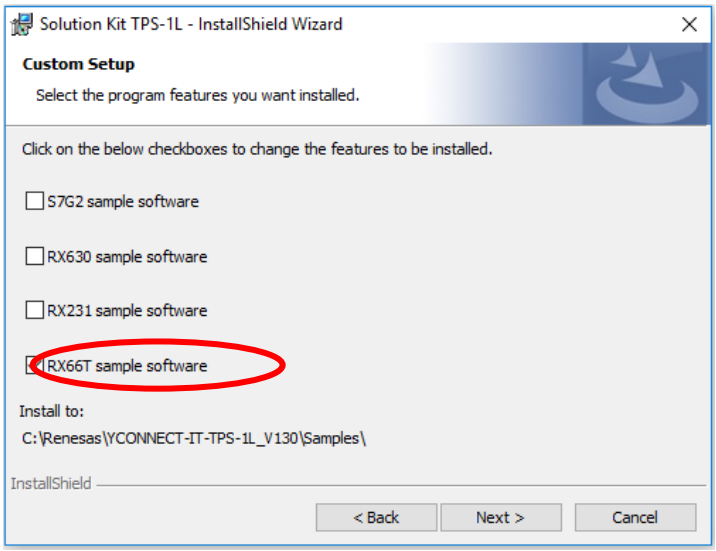
Chapter 3 RX66T Host Processor Connection

The steps described in this chapter of the Getting started document use primarily the e²studio package, a fully featured IDE for the RX66T (and other) microcontrollers. We assume that e²studio is already installed according to the instructions given in the installer.

3.1 Loading and Building Sample Project for RX66T

Please install the sample programs from the Renesas CD that came together with your YCONNECT-IT-TPS-1L kit; the installer program can be found in the <Sample Software> directory. When running the installer, please make sure that at least the RX66T software is selected as shown Figure 3-1.

Figure 3-1: Sample program installation

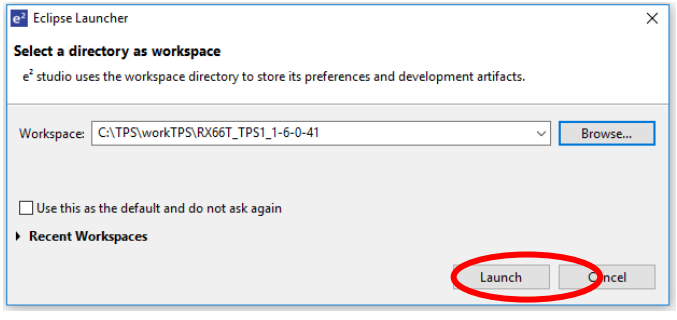


By default, the sample programs are copied to the <C:\Renesas\YCONNECT-IT-TPS-1L_V1x0\Samples\> directory (“x” is the version of your CD); the installation path can be modified according to your personal taste.

The folder, where you have stored the RX66T sample program must then be selected after starting e2studio as shown in Figure 3-2.

Note: The file and directory names used in the subsequent screenshots are not mandatory and deviate partially from the recommendations given in Chapter 2.2.

Figure 3-2: e² studio Eclipse Launcher window



When you click <Launch>, e²studio will open the welcome screen in Figure 3-3; on the welcome screen, please select <Import existing projects> to come to the <Import> dialog in Figure 3-4.

Figure 3-3: e²studio welcome screen

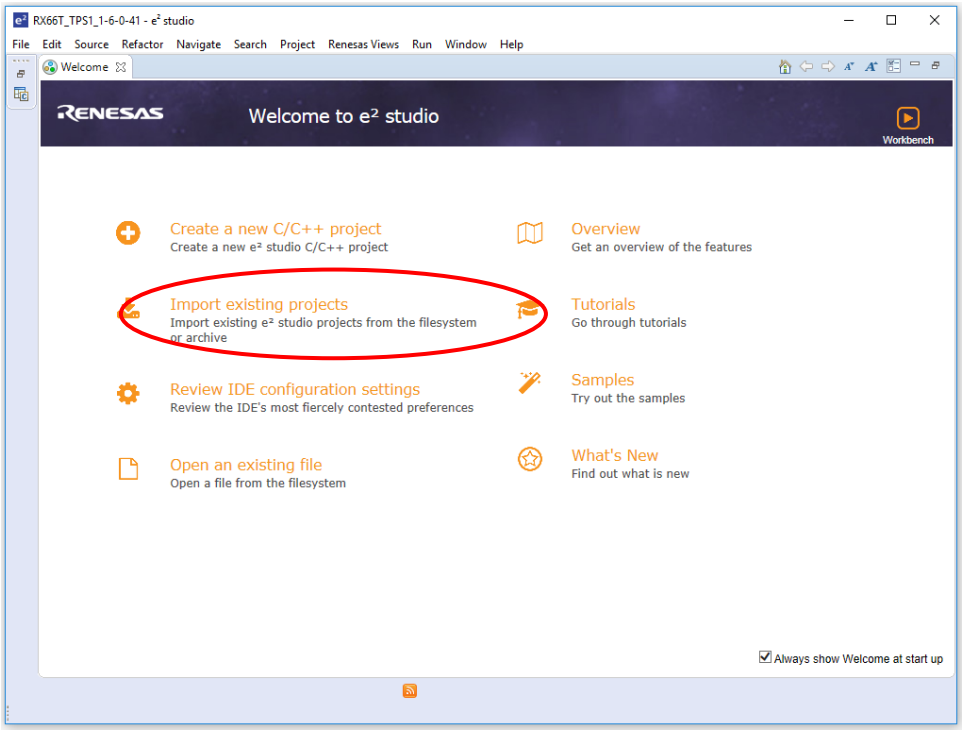
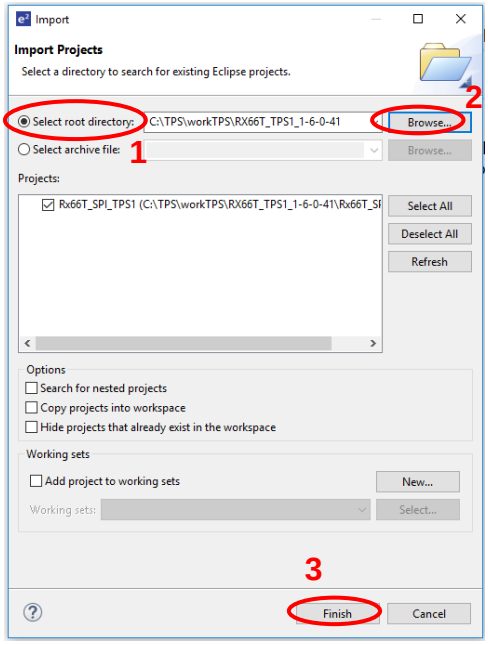


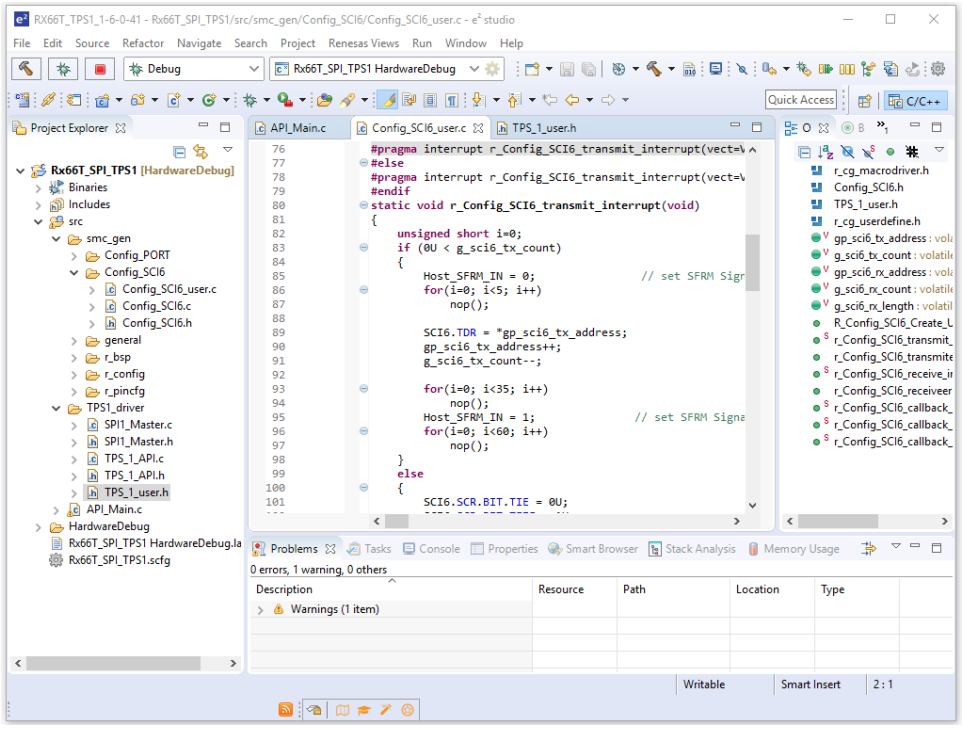
Figure 3-4: Import an existing project



Here you have to choose whether an existing project or an archive file should be imported. Please check <Select root directory> (1) and browse (2) to the directory, to which the sample software installer had copied the SW projects (see Figure 3-1). Selecting this directory will then list the available projects in the <Import> dialog and you can leave the dialog with <Finish> (3).

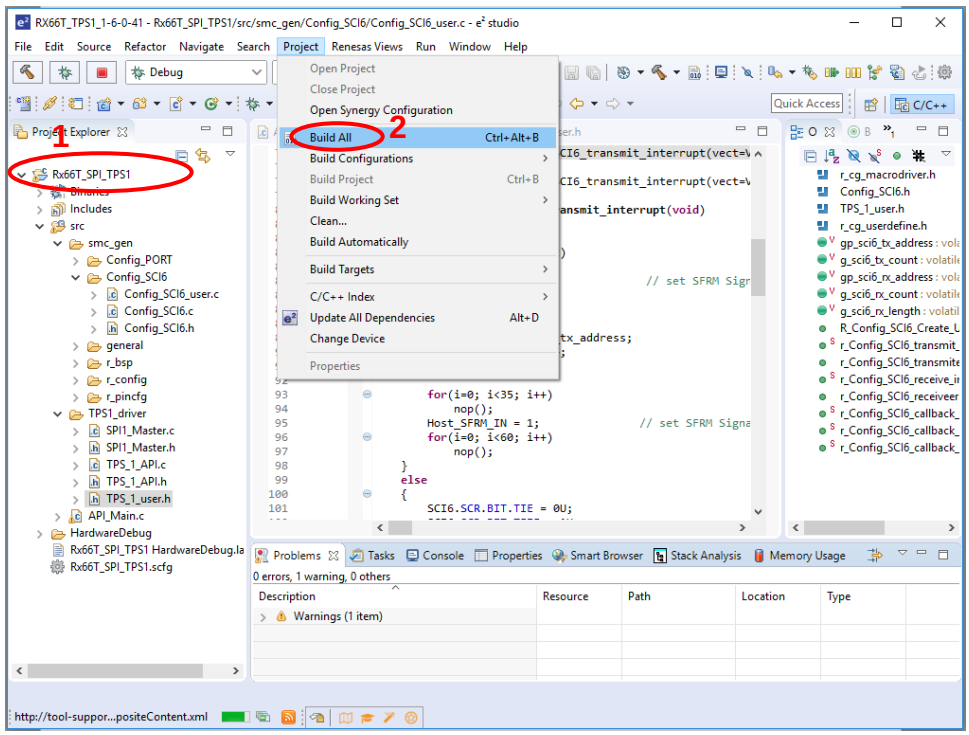
Now the project window will be opened – typically in the C/C++ perspective, in which code can be entered and where program compilation is done. An example for the project window can be seen in Figure 3-5.

Figure 3-5: e²studio workbench main screen



Now the workspace is prepared for further usage and the project can be built and debugged. In the C/C++ perspective of e2studio select the <RX66T_SPI_TPS1> portion in the <Project Explorer> area (1) and select <Build all> from the <Project> menu (2) as shown in Figure 3-6.

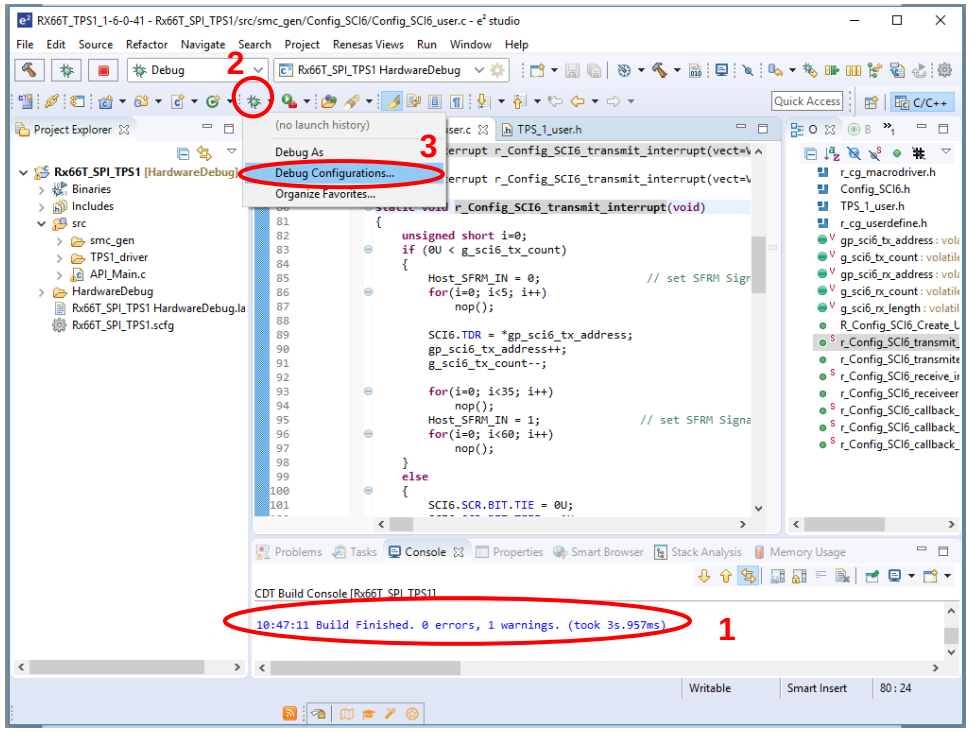
Figure 3-6: Building the project



While e2studio is building the project, a progress indicator is opened and additional information is output in the <Build console> area of e2studio. After a while the build should complete with no errors (warnings can be ignored) – see (1) in Figure 3-7.

When the build is completed, click on the drop-down menu beside the debugger icon (2) and select <Debug Configurations> from the menu (3)

Figure 3-7: Invoking the debugger



e²studio will now open a dialog shown in Figure 3-8 in which debug configurations can be managed and configured. In this dialog double-click <Renesas GDB HardwareDebug> and select the <RX66T_SPI_TPS1 HardwareDebug> configuration (1).

Make sure that

- the settings on the <Main> tab (2) is the same as in Figure 3-8
- the settings on the <Debugger> tab (1) and the <Connection Settings> tab (2) under the <Debugger> tab are the same as in Figure 3-9

and then leave the dialog with <Debug> (3). Click <Yes> in the confirmation dialog.

Note: Power settings in the <Connection Settings> tab (2) is not mandatory. In this case our RX66T CPU card getting 3.3V supply voltage from E1 emulator. For others power supply option please have a look on RX66T CPU card user's manual (R12UZ0029EJ0100)

Figure 3-8: Debug configuration setting 1

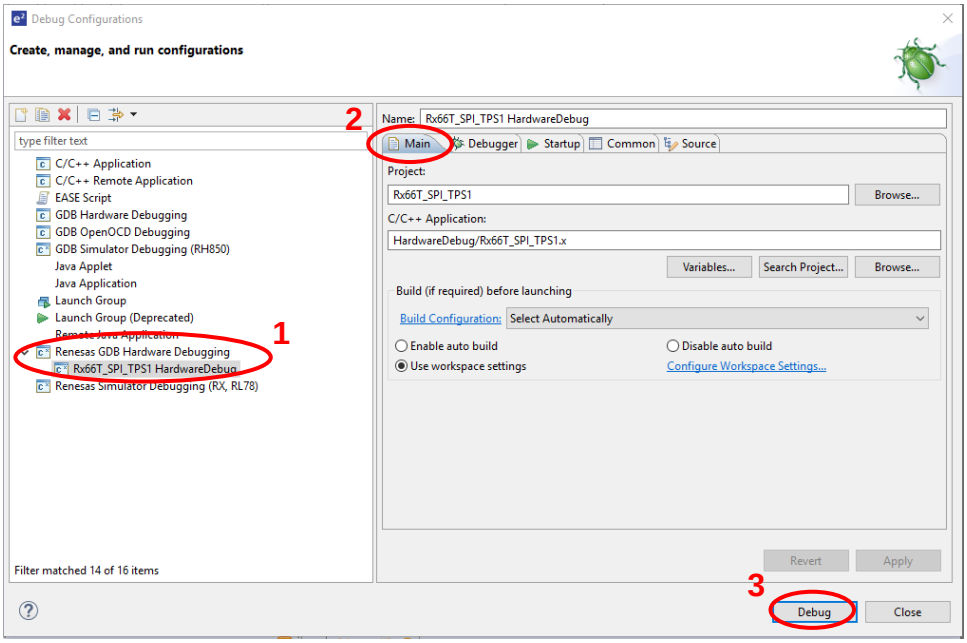
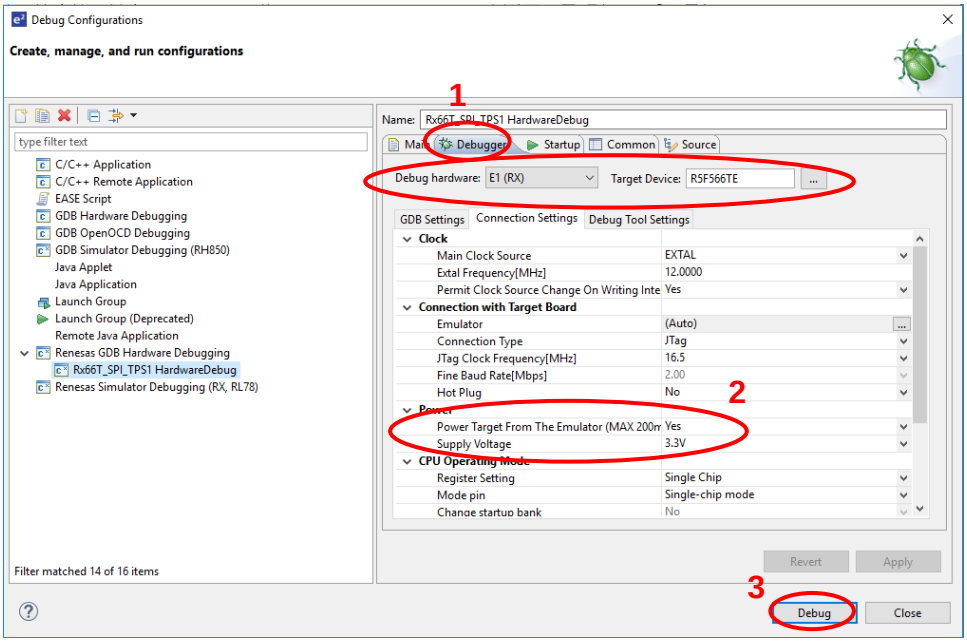
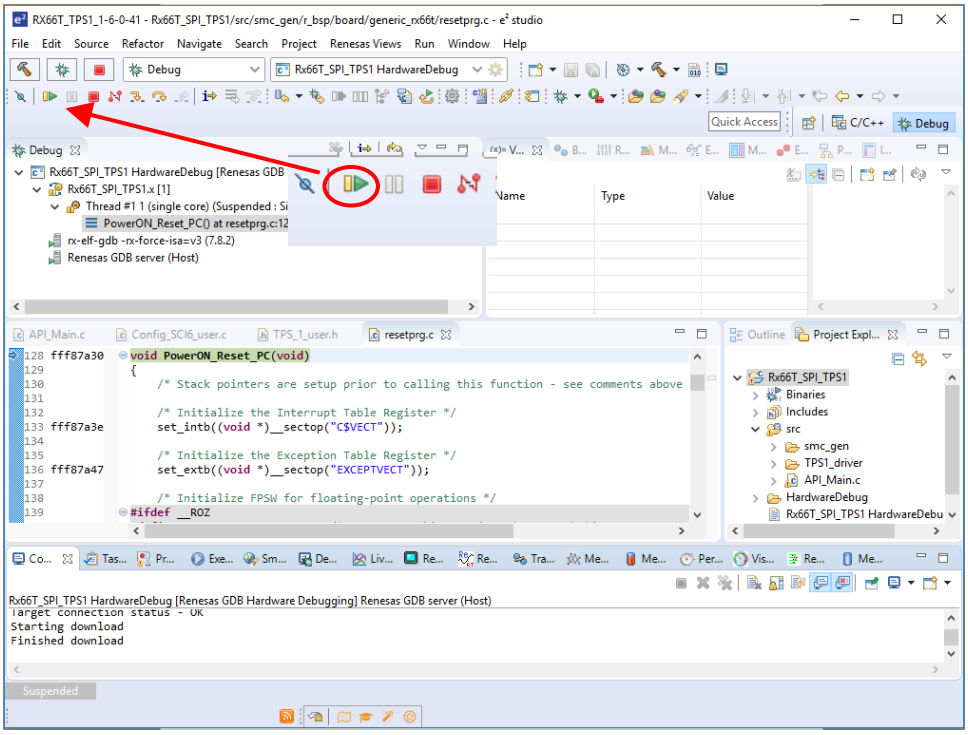


Figure 3-9: Debug configuration setting 2



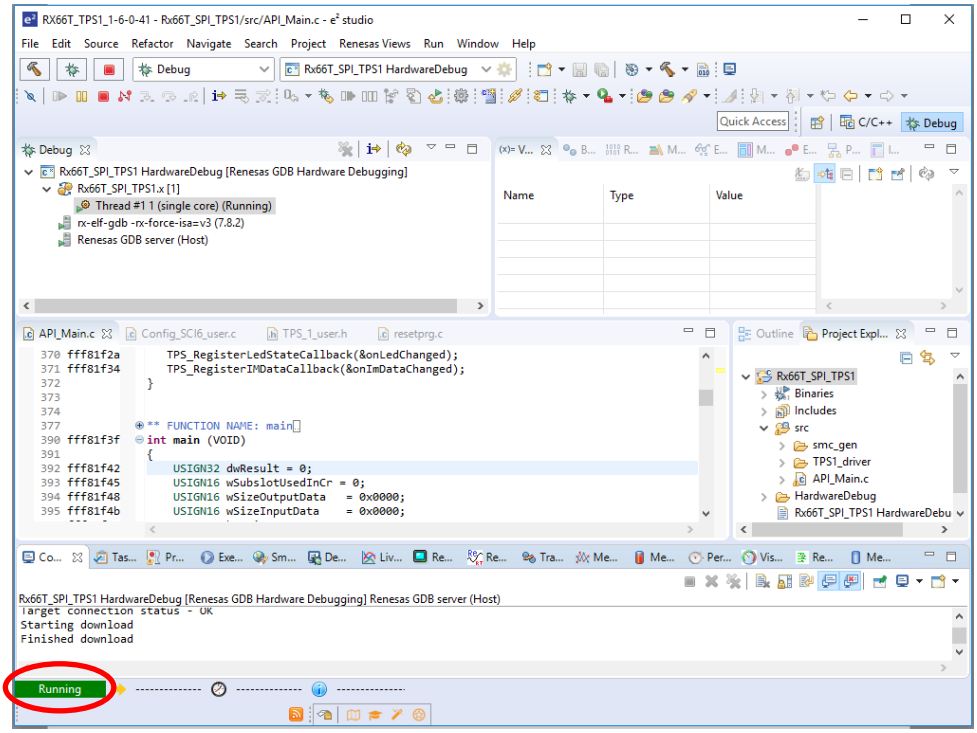
e²studio now changes from the <C/C++ Perspective> to the <Debug Perspective> in Figure 3-10; the program execution is first suspended at the <PowerON_Reset_PC> function. If you now start the RX66T by clicking the <Run> button, it will run until the <main()> function is reached.

Figure 3-10: Debug perspective



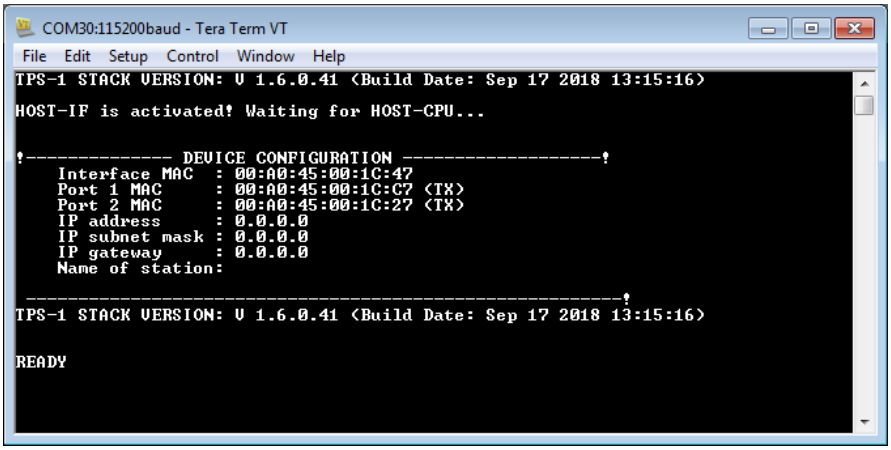
Another click on the <Run> button will unleash the RX66T and the RX66T sample application will be continuously executed. This is indicated in the <Debug Perspective> as shown inFigure 3-11.

Figure 3-11: Debug perspective when the processor is running



In reaction to the running sample application, the TPS-1 will move from the “waiting for host CPU” status to the “Device ready” status in Figure 3-12 and TPS-1 stack version will also show up in this picture, here it’s showing V1.6.0.41

Figure 3-12: TPS-1 information the device ready status



It may be required to reset the TPS-1 board and/or restart program execution on the RX66T in e^{2}studio to get the TPS-1 and the RX66T synchronized.

3.2 Changing Host Communication Mode

The TPS-1 can use different interfaces for communication with its host CPU. The most frequently used communication method is using the SPI interface – this is why serial communication was also used in this manual as an example.

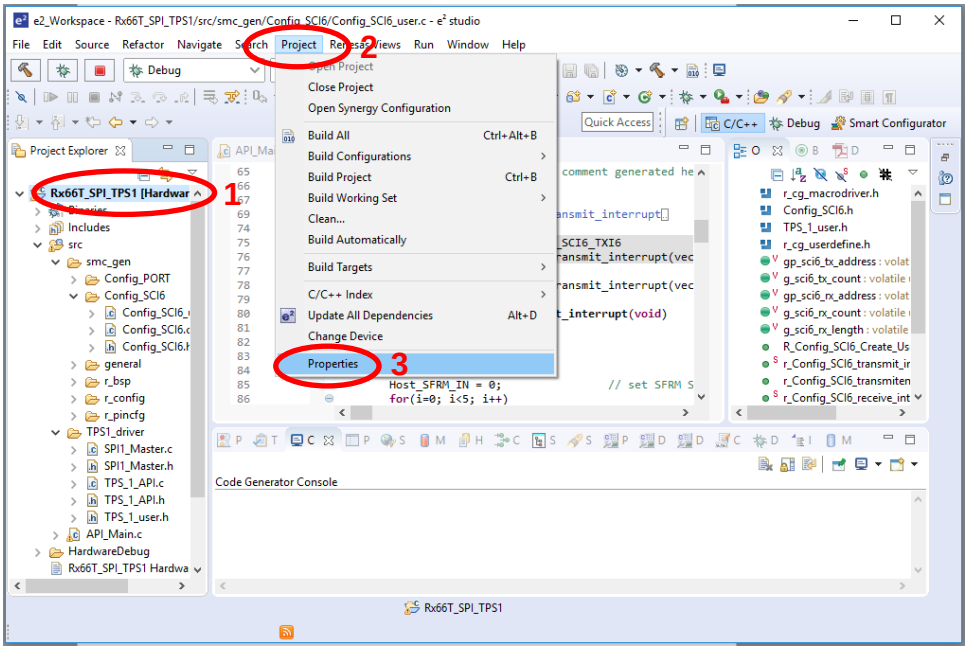
Note: Parallel host communication requires hardware changes on the RX66T CPU card referenced in this manual. In case that you want to try this, please contact the TPS-1 support. In the following we will only describe the required changes in the software.

If you would like to use parallel host communication instead, please modify the software project as described in this chapter.

In Figure 3-5 the RX66T project had been loaded into e2studio. Before building the project as shown in Figure 3-6, the following changes have to be made in the project configuration:

In the C/C++ perspective select the <RX66T_SPI_TPS1> portion in the <Project Explorer> area (1), open the <Project> menu (2) and select <Properties> (3) as illustrated in Figure 3-13.

Figure 3-13: Opening the <Project properties> dialog

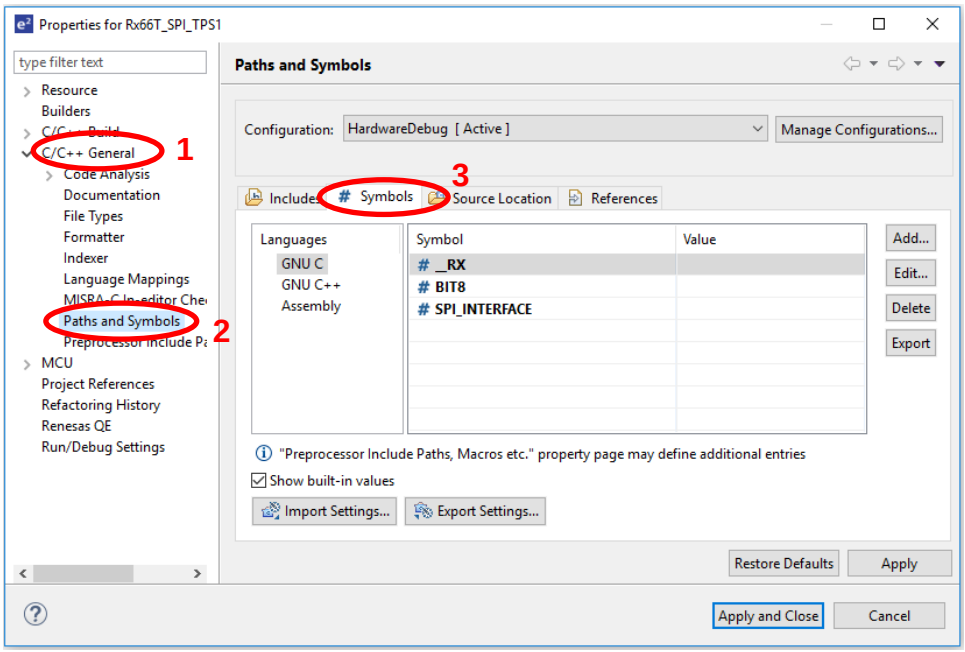


In the <Properties> dialog expand <C/C++ General> (1) on the left-hand side and select <Paths and Symbols> (2) to come to the dialog in Figure 3-14. On the <Symbols> tab the currently used symbols are listed. They can be changed with the <Add>, <Delete> and <Edit> buttons.

Example: For reconfiguring the project to a 16-bit parallel interface between TPS-1 and the RX66T, you must replace the defines as follows_

- <SPI_INTERFACE> by <PARALLEL_INTERFACE>
- <BIT8> by <BIT16>

Figure 3-14: Modifying the host CPU communication method



Chapter 4 PROFINET Configuration

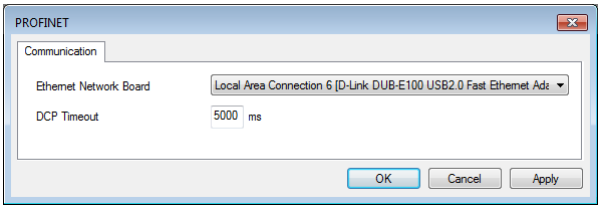
This chapter describes the steps that are required to set up the PROFINET communication between the PC running a PROFINET software controller named < PROFINET Smart Control> and the TPS-1 as a PROFINET device. Please refer to the document <PROFINET Configurator - Quick Start Guide.pdf> to get familiar with the PROFINET Configurator as we describe only the steps which are required to set-up the TPS-1 Low-cost Solution Kit demo.

Note: Make sure to run the PROFINET Configurator as administrator.

When the PROFINET Configurator has been started, it is recommended to specify the network adapter to be used – normally, but not necessarily the same adapter that was used for the stack update in Chapter 5.

Select <PROFINET Configuration> from the <Extras> menu, chose the desired network adapter in the dialogue shown in Figure 4-1 and click <Ok>.

Figure 4-1: Network adapter selection



Next you can load the “TPS-1” example as a starting point. This example is stored in the <tps-1.zcp> file in the <\PROFINET Configurator\Example Project> folder on the Phoenix Contact Software CD or TPS work directory (mentioned in Chapter 2.2). The screen like Figure 4-2 should appear.

The <tps-1> IP address range in the <Bus Structure> (1) menu must be set accordingly to a range between 192.168.16.1 and 192.168.16.254 (2). Please enter the value for the last IP-address first.

Now the PROFINET Smart Control settings have to be checked in the <Bus Structure> menu at level <Phoenix Contact Software PROFINET 192.168.16.25>. At this stage the DNS/PROFINET device name, IP address and subnetmask of the PROFINET Smart Control have to be set as shown in Figure 4-3.

Figure 4-2: PROFINET Configurator with “TPS-1” example

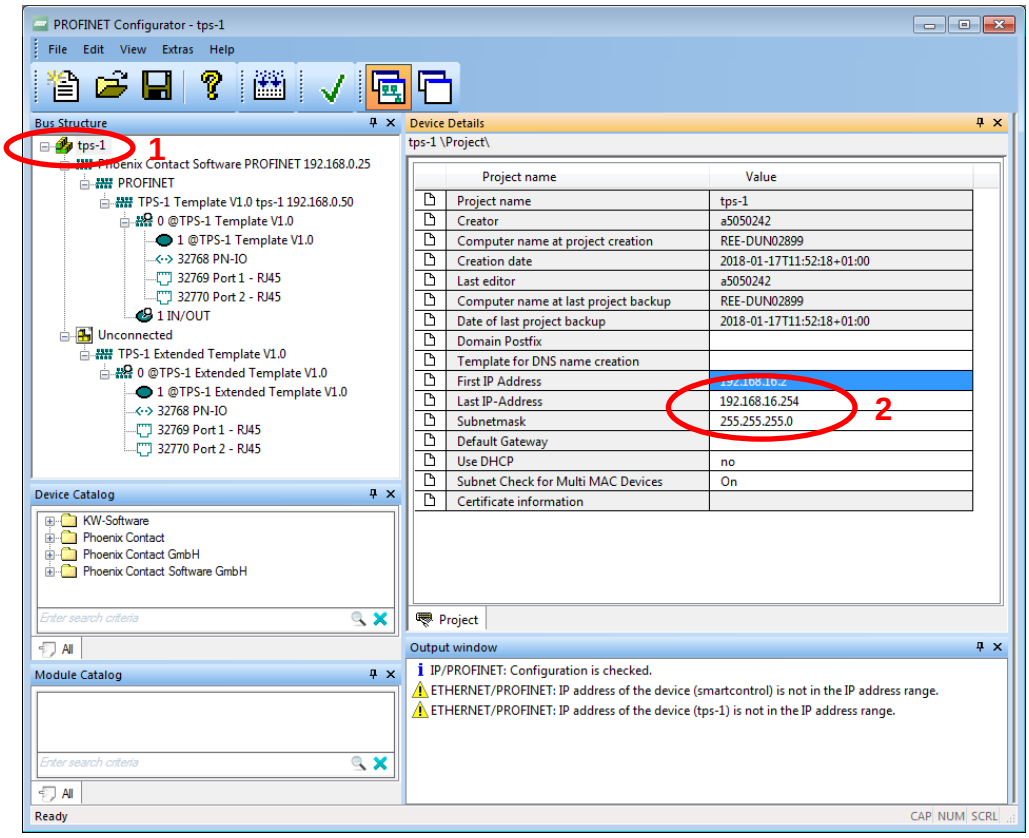
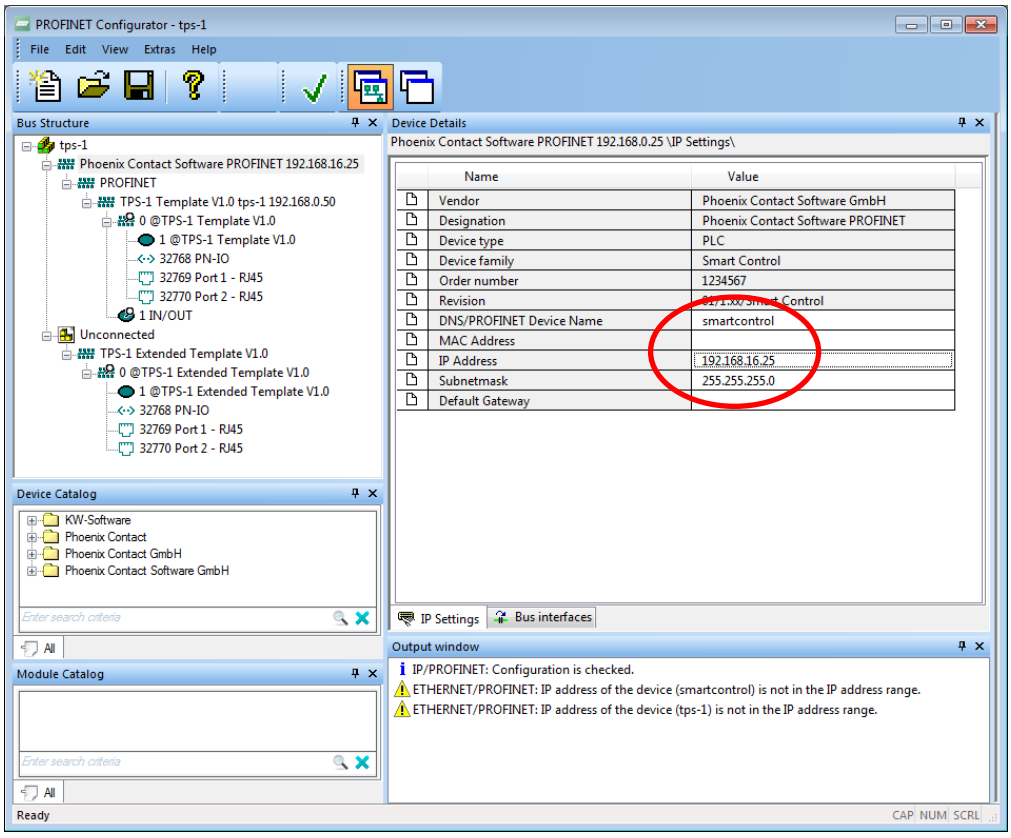
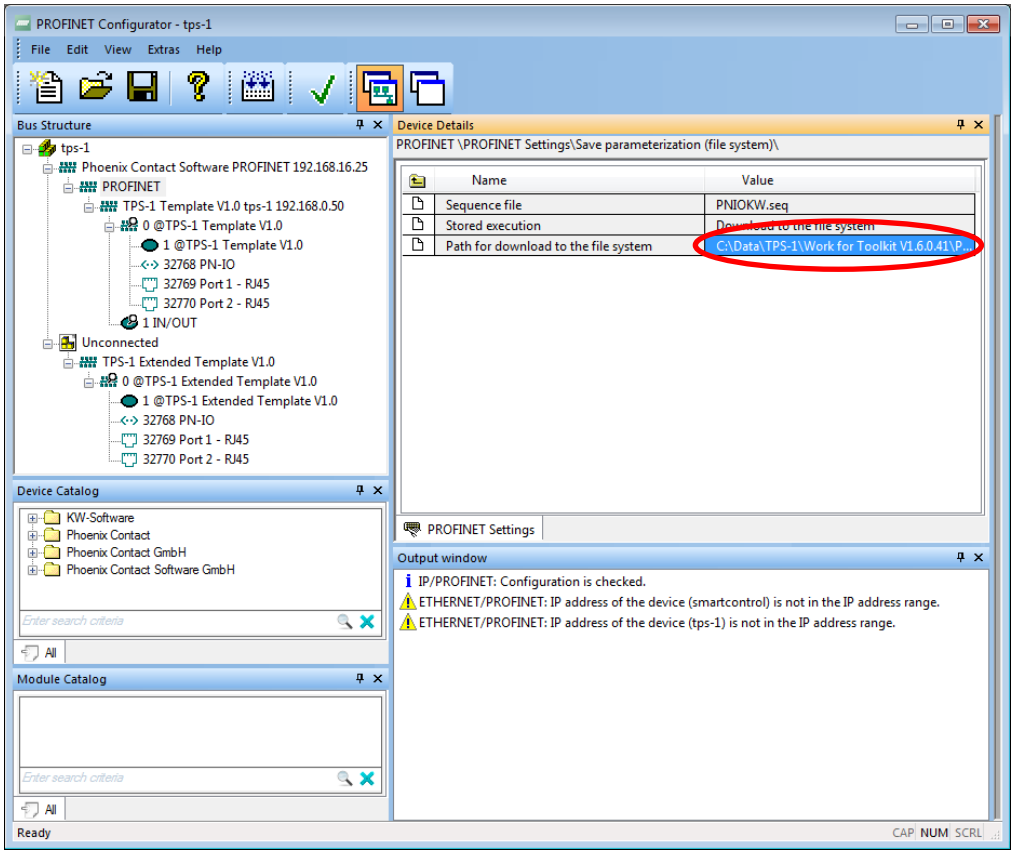


Figure 4-3: PROFINET controller settings



To set the path for the file system to save the parameter file <IPPNIO.xml> for the device configuration, go to the <PROFINET> level in the <Bus structure> area and double-click on the directory symbol close to <Save parametrization>. Then specify the path according to Figure 4-4. It is recommended, but not mandatory to store the file within your working file structure for the TPS-1 development tool kit (see Chapter 2.2).

Figure 4-4: Path for file system



Finally, the TPS-1 device configuration has to be checked and it must be verified that the <tps-1> device name is available in the network. These items will be seen in the <Bus Structure> area at level <TPS-1 Template V1.0 tps-1 192.168.0.50>. In the <Device Details> area use once the <PROFINET Settings> and then the <PROFINET Stationnames> tab as shown in Figure 4-5 and Figure 4-6 and verify that all settings are made properly.

In the <PROFINET Settings> set the IP address to 192.168.16.2 and in <PROFINET Stationnames> check, if a device with the name <tps-1> is found when you push the <Refresh> button. If <tps-1> is not found, select the device and give it the name <tps-1> with the <Assign Name> button. The TPS-1 device may be listed with another IP-address as in Figure 4-6; this mismatch is automatically sorted out when the PROFINET network is started.

Figure 4-5: TPS-1 device settings

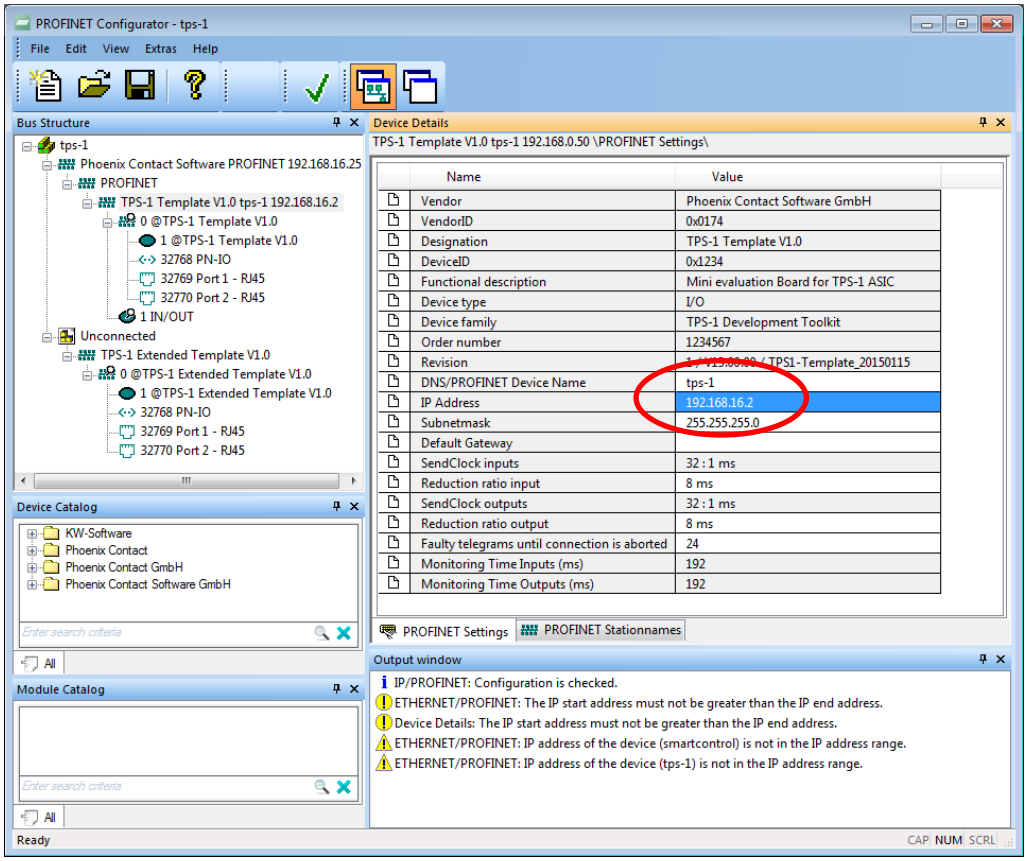
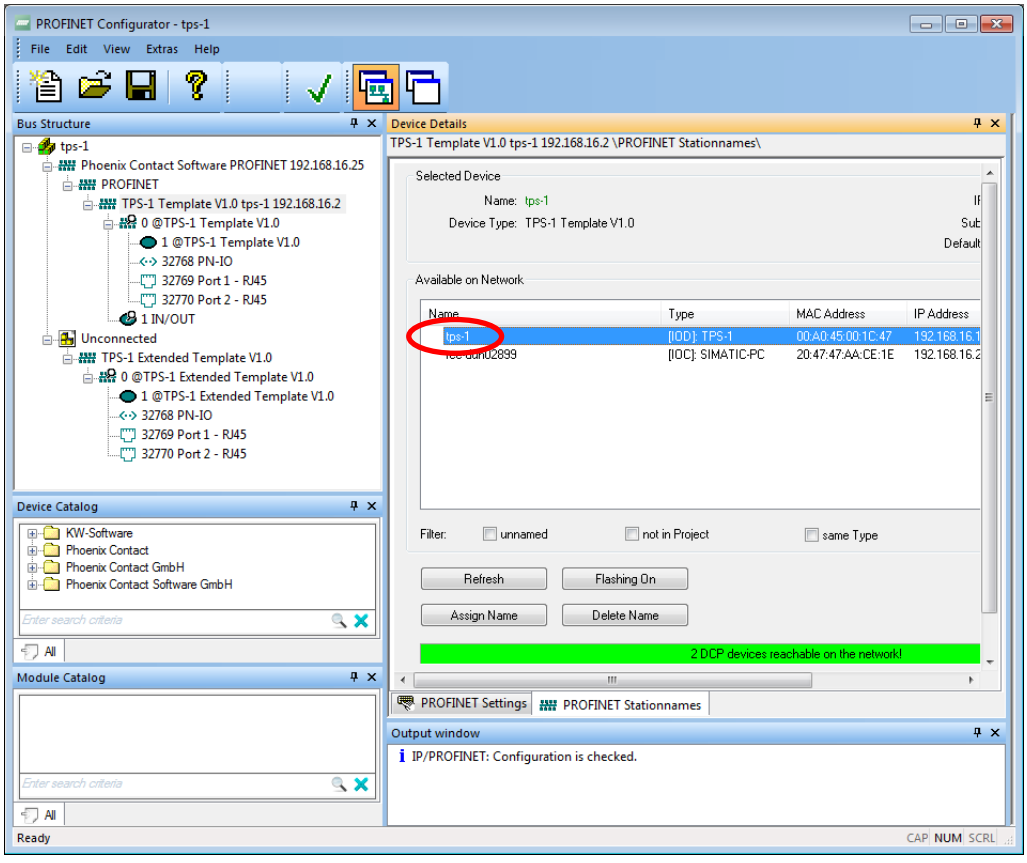
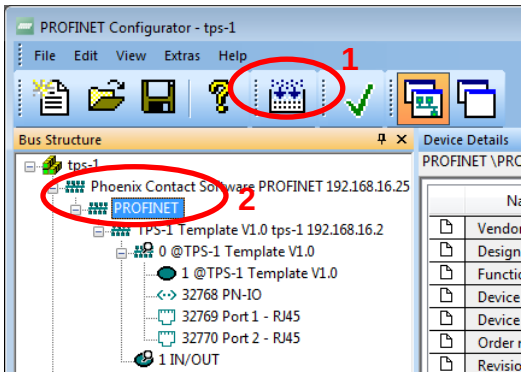


Figure 4-6: TPS-1 device name in network



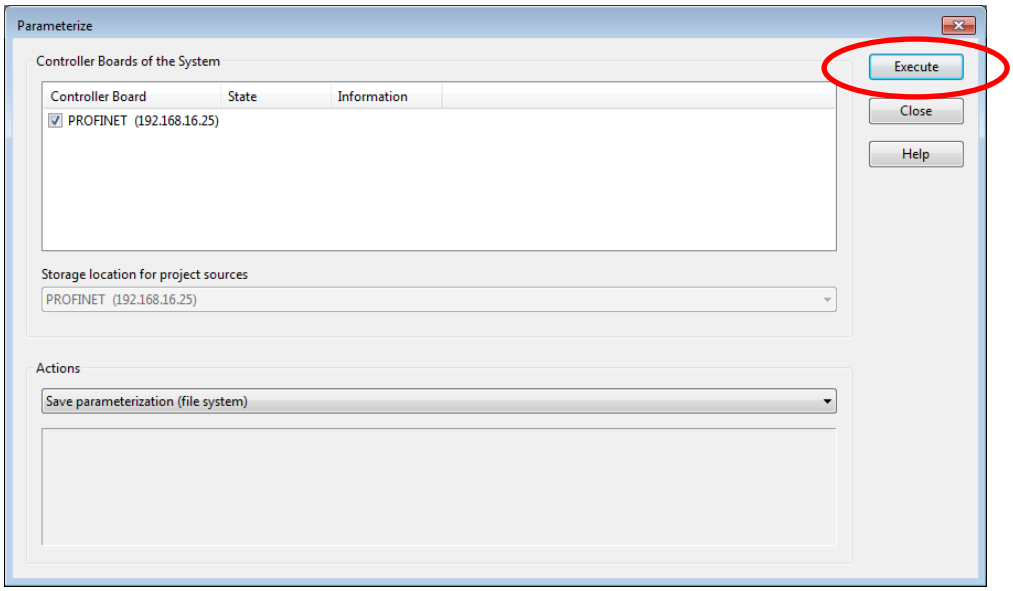
Now the configuration is finished and the device configuration file <IPPNIO.xml> shall be generated by pressing the <Parameterize> button (1) in the main menu, when in the <Bus Structure> window the level <PROFINET> is highlighted (see (2) in Figure 4-7).

Figure 4-7: Generating the <IPPNIO.xml> file



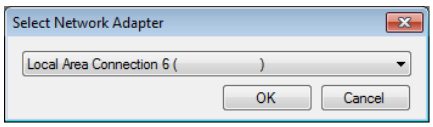
In the parametrization process you are now asked for the controller (i.e., the network adapter) for which you would like to parametrize (see Figure 4-8). In this dialog the settings are pre-defined according to the settings, that you have previously made in the PROFINET Configurator, and you can simply click <Execute>.

Figure 4-8: Selecting the PROFINET Controller for parametrization



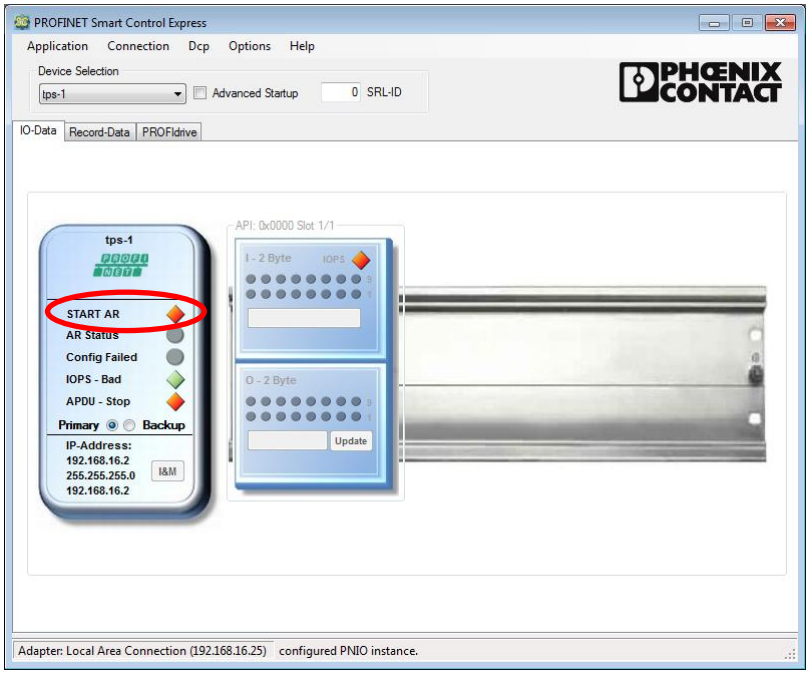
To run the demo the PROFINET Smart Control has to be opened (as administrator). PROFINET Smart Control first asks you for the network adapter that should be used for the PROFINET connection in the dialog in Figure 4-9. Specify the same network adapter that was used in the PROFINET Configurator and continue.

Figure 4-9: Network adapter selection for PROFINET Smart Control



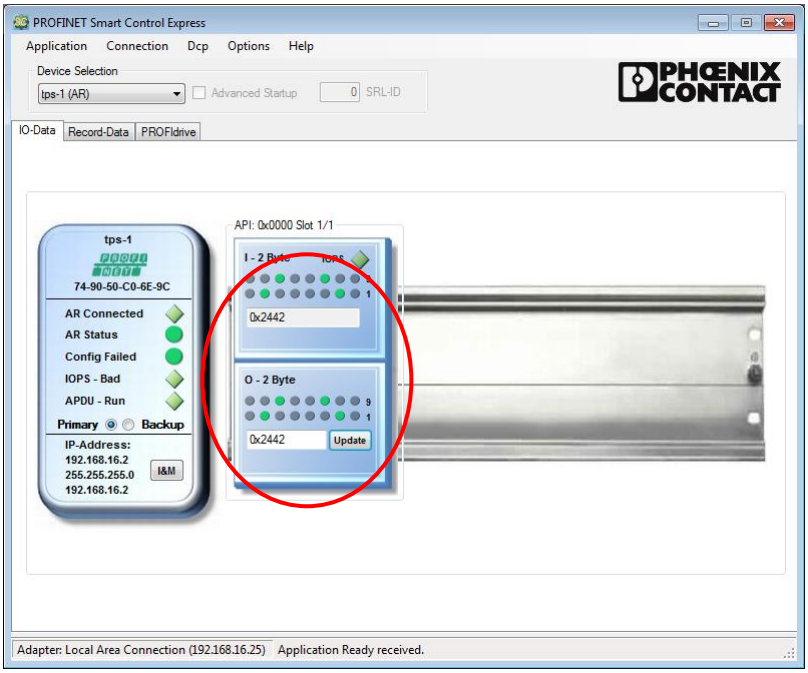
Load the <IPPNIO.xml> file, that you have just generated with the PROFINET Configurator, using <Application> and <Parse XML-File> in the menu. Then choose the device <tps-1> in the <Device Selection> box and the PROFINET Smart Control will pop-up similarly as shown in Figure 4-10.

Figure 4-10: PROFINET Smart Control start



Now start the PROFINET connection by pressing the <START AR> button and after the <AR STARTED> response was indicated, values can be inserted in the <Slot1 O-2 Byte> output field by pressing the <Update> button as shown in Figure 4-11.

Figure 4-11 PROFINET Smart Control operation



When output data is entered in the <Slot1 O-2 Byte> output field and when the <Update> button was pressed, the data travels

- via PROFINET from the PC to the TPS-1
- from TPS-1 to the RX66T
- the RX66T mirrors the data back to TPS-1
- from the TPS-1 back to the PC via PROFINET

The output data is then visualized in the <Slot1 I-2 Byte> input field (see Figure 4-11).

Chapter 5 Firmware Image Replacement on TPS-1

At delivery the external flash that is connected to the TPS-1 has the default image of the TPS-1 stack pre-loaded. This default image is normally used for production purposes and has therefore the release version of the PROFINET stack. To ease the first steps with the TPS-1 Low-cost Solution Kit we recommend to replace the release version of the stack with the debug version. The debug version provides lots of information via the TPS-1 UART.

Note: 1. The steps described in this chapter are optional but recommended.

2. The <Netnames+> and TPS Firmware Updater programs used in this chapter must be run with administrator privileges.

The replacement of the stack version can be regarded as an update process. Therefore, we will use the TPS Firmware Updater for its execution.

5.1 Generate Image File

The stack image has to be prepared for download. In this process the binary images for the stack are “personalized” to a specific PROFINET device in a larger PROFINET network. This step doesn’t look meaningful in a one-device-network, but it makes absolute sense in a multi-device network, in which some devices should be updated and others shouldn’t.

Two files for download via the Firmware Updater can be prepared. These files are composed from the files <TPS_Updater.bin> respectively <TPS_Stack_Debug.bin> and an editable file named <hdr.txt> that contains PROFINET device specific information like VendorID and DeviceID. If you followed the file structure recommendation given in Chapter 2.2, these files are found in <Work Directory>\<TPS Stack>.

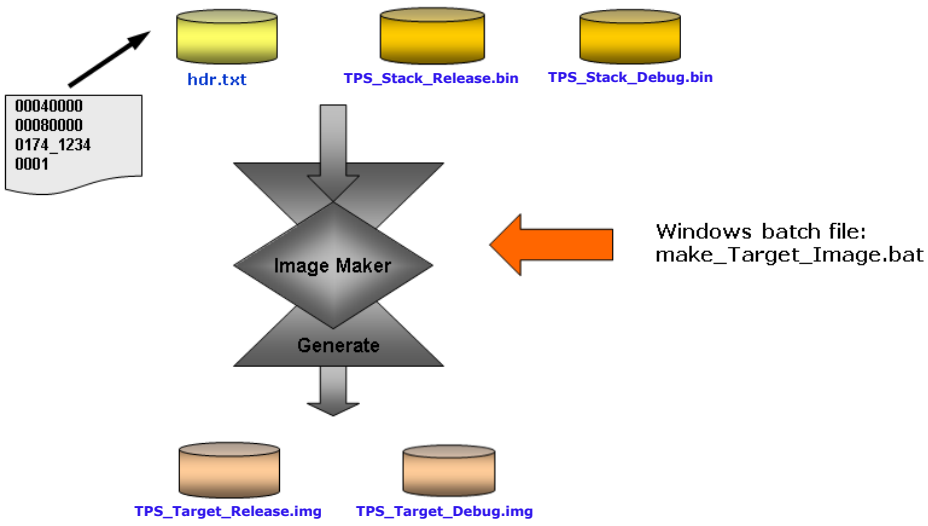
Remark: The TPS-1 solution kit is provided with two versions of the PROFINET stack:

- A normal version in <TPS_Stack_Release.bin>
- A debug version with extended messages in <TPS_Stack_Debug.bin>

In this <Getting Started> document we recommend to use the debug version.

The file integration is done by a service program called <TPS_Image_Maker>, which is started using a batch file called <make_Target_Image.bat>. Figure 5-1 illustrates this process.

Figure 5-1: Download file creation process



Run the <make_Target_Image.bat> file in the <Work Directory>\<TPS Stack> directory. As a result, the files <TPS_Target_Debug.img> and <TPS_Target_Release.img> are generated.

The same process must be applied to the <TPS_Updater.bin> file in the TPS_Updater_ETH> directory by executing the <make_Updater_Image.bat> file.

All three <*.img> images together with the descriptor file <174_1234_01upd.dat> must be copied to the <FW_Updater_Dir> directory. Operation of the update process is controlled by a descriptor file.

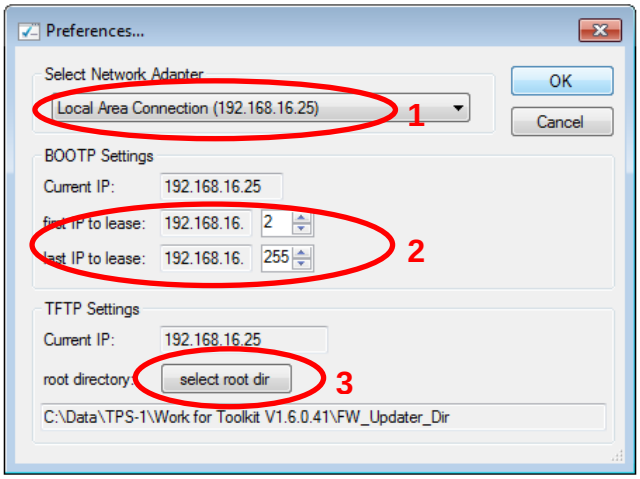
5.2 TPS-1 Firmware Update

5.2.1 TPS Firmware Updater preferences setting

After these preparations (generate image files) we'll work with the TPS Firmware Updater. In the TPS Firmware Updater window select <Tools> and <Preferences>; this will show you the dialog in Figure 5-2, in which the following settings have to be made:

- Select the network adapter to which the TPS-1 board is connected (1)
- Select the "window" of IP addresses to lease; typically, a range of addresses in the subnet that is visible by the selected network adapter (2)
- Set the <root directory> to the directory (3) in which the download images and the <174_1234_01upd.dat> file have been stored (typically <Work Directory>\<FW_Updater_Dir> - i.e., different from Figure 5-2.)

Figure 5-2: Preferences setting in TPS Firmware Update



5.2.2 Transfer the Updater Image

At first the Updater must be written into the Flash. To initiate an update of the TPS_Updater the entry "UpdateUpdater = 1" must be set in the descriptor file. In this case the TPS-1 requests the file „TPS_UpdaterTarget.img" that is defined by the value UpdaterFName.

After the download the TPS-1 reboots. If NextBoot = 1 the TPS-1 reboots directly into update mode. In the initial installation process this is useful because the firmware image must always be sent after the updater has been installed.

With descriptor file as shown below the TPS-1 will repeatedly download the updater image until the descriptor file is edited to download the target image.

```
UpdateTarget = 0
TargetFName = TPS_Target_Debug.img
NextBoot = 1
UpdateUpdater = 1
UpdaterFName = TPS_UpdaterTarget.img
```

5.2.3 Transfer the Firmware Image

After programming the updater image, you must edit the descriptor file for downloading the target image as shown below.

```
UpdateTarget = 1
TargetFName = TPS_Target_Debug.img
NextBoot = 0
UpdateUpdater = 0
UpdaterFName = TPS_UpdaterTarget.img
```

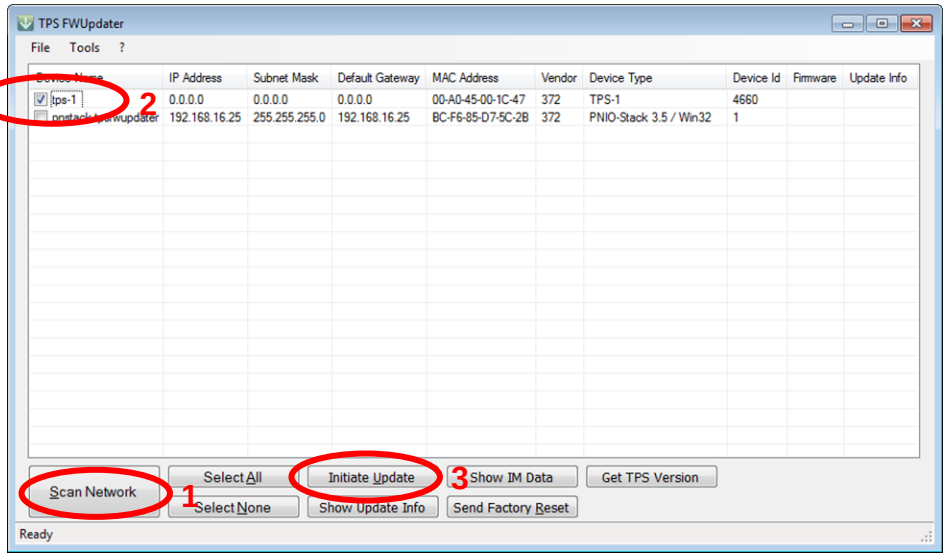
After the next reboot the edited file is transferred to the TPS-1. The TPS-1 executes the commands and starts the transfer of the TPS_Target_Debug.img.

Because the NextBoot = 0, TPS-1 will quit the download loop and boot into normal operation mode after the firmware file has been installed. The result is documented in a status file that also contains error codes.

5.2.4 Finalize the Firmware Update

To finalize the firmware update, click <Scan network> in the Firmware Updater as shown in Figure 5-3. You should then see two devices listed – similar to the display in <Netnames+>. Then check the TPS-1 based device (2) and click <Initiate Update> (3). Click <Yes> in the subsequent confirmation dialog.

Figure 5-3: Firmware Updater main window



Now the actual download is executed. The dialog in Figure 5-4 illustrates the download progress; after some seconds the Firmware Updater informs you that the download has been performed with no errors (see Figure 5-5).

Figure 5-4: Update progress indicator

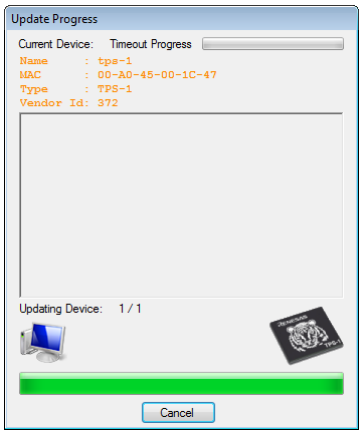
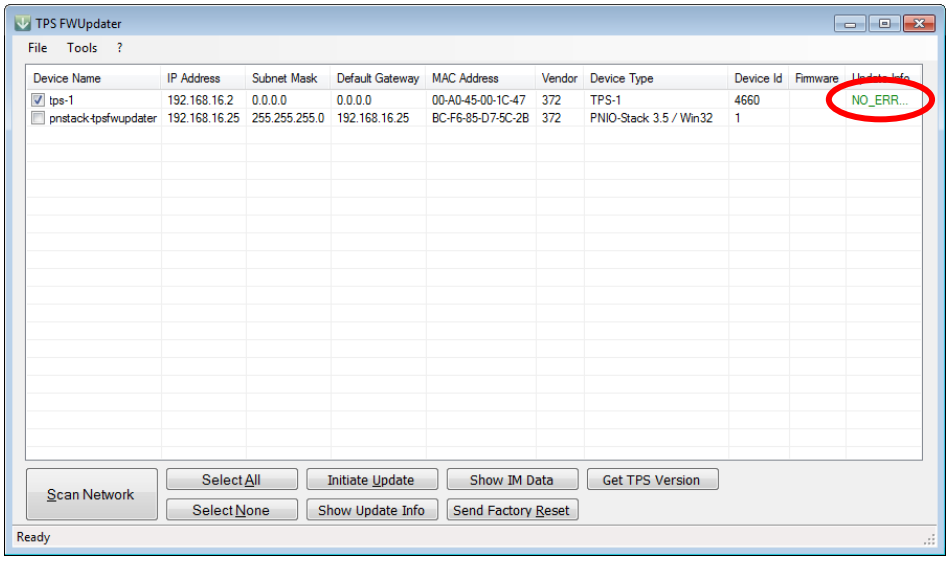
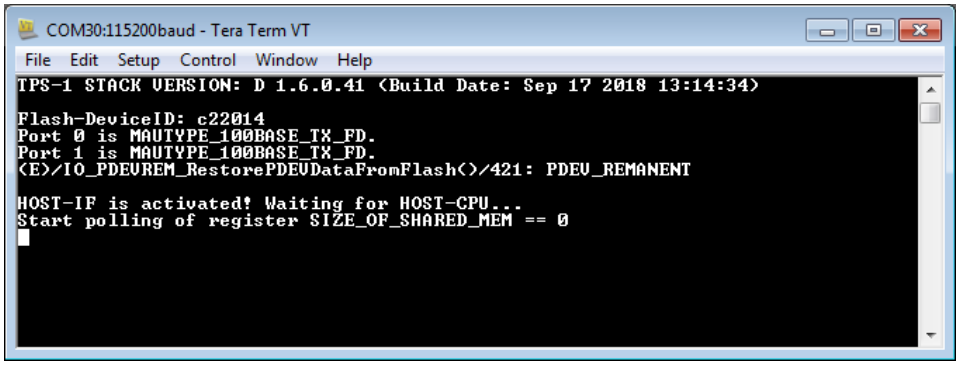


Figure 5-5: Update successfully completed



After a reset, the TPS-1 will now execute the debug version of the PROFINET stack(here it is V1.6.0.41, check the current stack version) and run it to a point where it is waiting to communicate with the host CPU as shown in Figure 5-6.

Figure 5-6: Start-up messages of the debug version of the stack

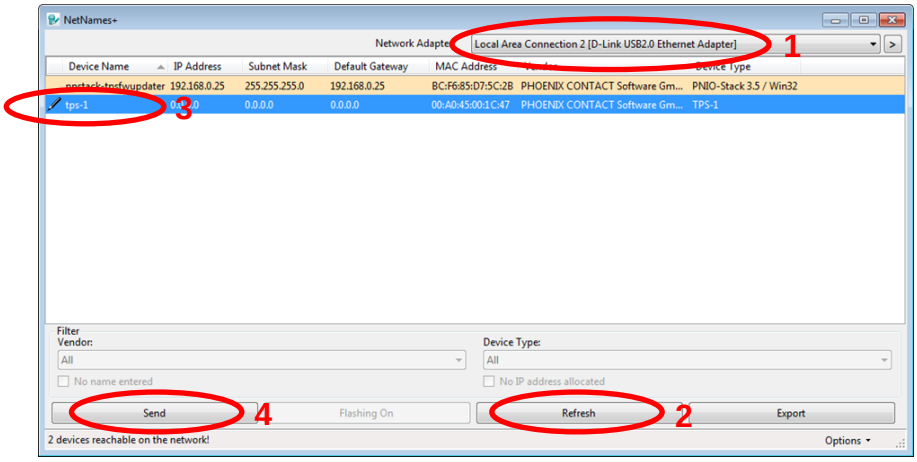


5.2.5 Netnames+ Tool

Start the <Netnames+> tool to assign a name to the TPS-1 based PROFINET device. <Netnames+> is automatically installed together with the PROFINET Configurator and can be found in the Windows start menu. Additionally, please start the TPS Firmware Updater.

In the <Netnames+> window select the network adapter, to which your TPS-1 board is connected (1), then click the <Refresh> button (2). You should now see two “devices” listed: the TPS-1 and the Firmware Updater itself as shown in Figure 5-7. Double-click on the <Device name> field for the TPS-1-based device (3) and enter <tps-1> as device name.

Figure 5-7: Netnames+ display after scanning the network



When the new name has been entered, it must be informed to the TPS-1 by clicking the <Send> button (4). With another <Refresh> action you can check whether the device name has been successfully transferred.

Chapter 6 Erasing the Flash Memory

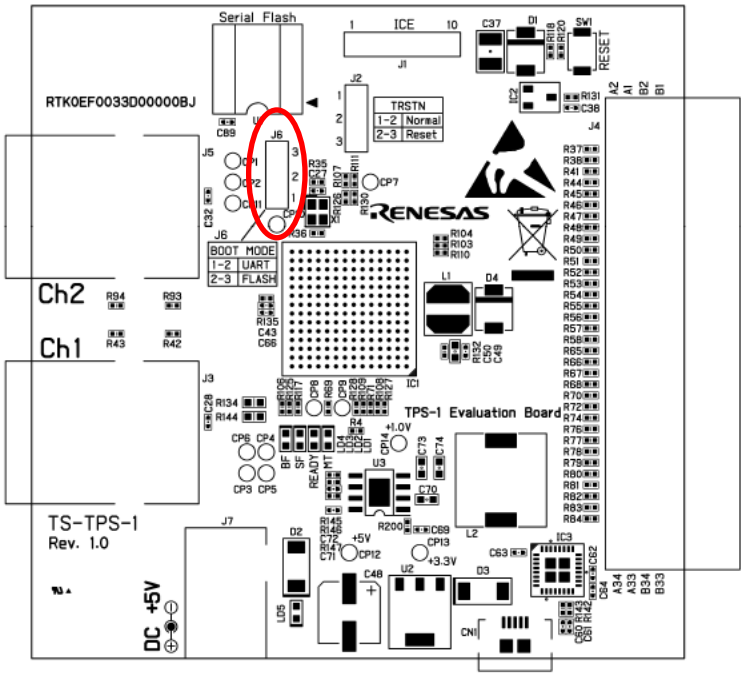
Though TPS-1 and the related development tool kit are easy to use, you may reach a point where you want to start from scratch again. On delivery of the TPS-1 Low-cost Solution Kit the serial flash connected to the TPS-1 is pre-programmed with the PROFINET stack default image. To return to this status a programmer for the serial flash is required. If case that you do not have a programmer available, an alternative solution is possible.

The TPS-1 development toolkit includes a little utility that erases the flash completely. This chapter will show how this program is used.

Note: During the following steps you should watch the UART communication of TPS-1 in a terminal emulation program as shown in Chapter 2.

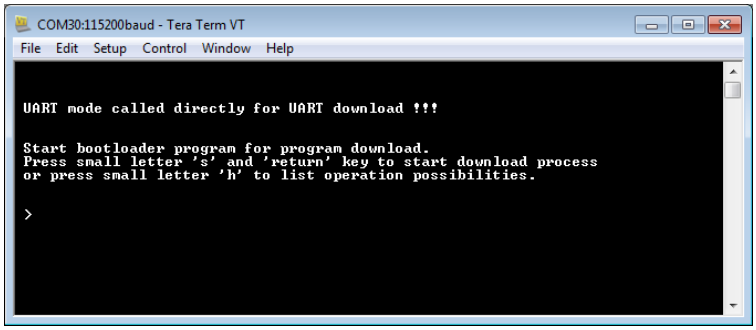
To erase the serial flash memory, power off the board, move jumper J6 on the TPS-1 board from position 1-2 to position 2-3 and power the board up again. After power up put J6 back to position 1-2 (while the board is powered on). See Figure 6-1 for the position of J6.

Figure 6-1: Jumper position on TPS-1 board



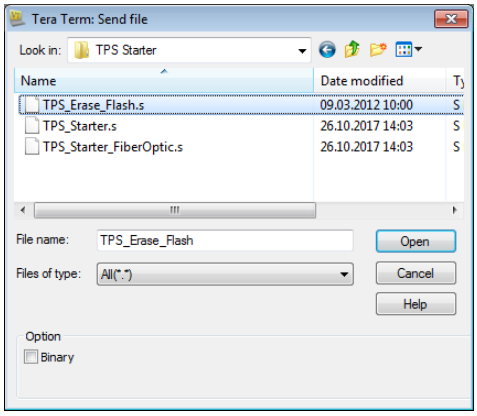
After reset the TPS-1 will report via UART that it had been forced to boot up in UART mode. In this mode you can manually download programs to the TPS-1 via UART; Figure 6-2 shows a screen shot.

Figure 6-2: UART output in case of UART boot



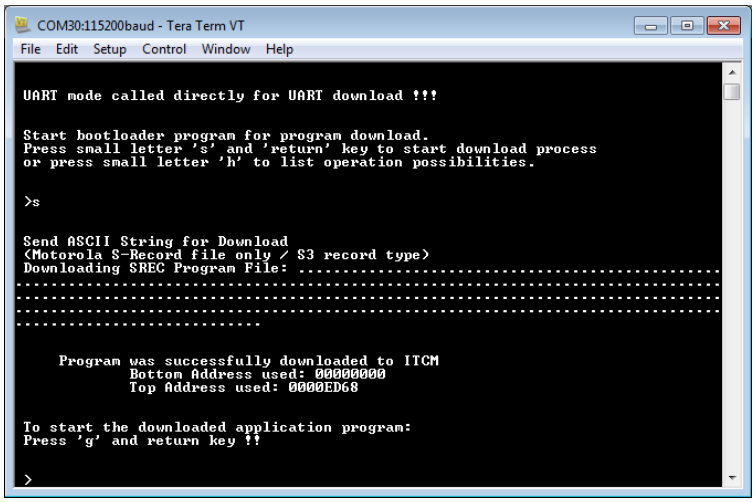
Next press <s> in the terminal window in order to start download of an S-record file. Then select <Send File> in the <File> menu of the terminal emulation program (Figure 6-3) and look for the file <TPS_Erase_Flash.s> in your working directory structure. Send the file to the TPS-1 with <Open> and wait until it is completely downloaded.

Figure 6-3: Selecting the download file



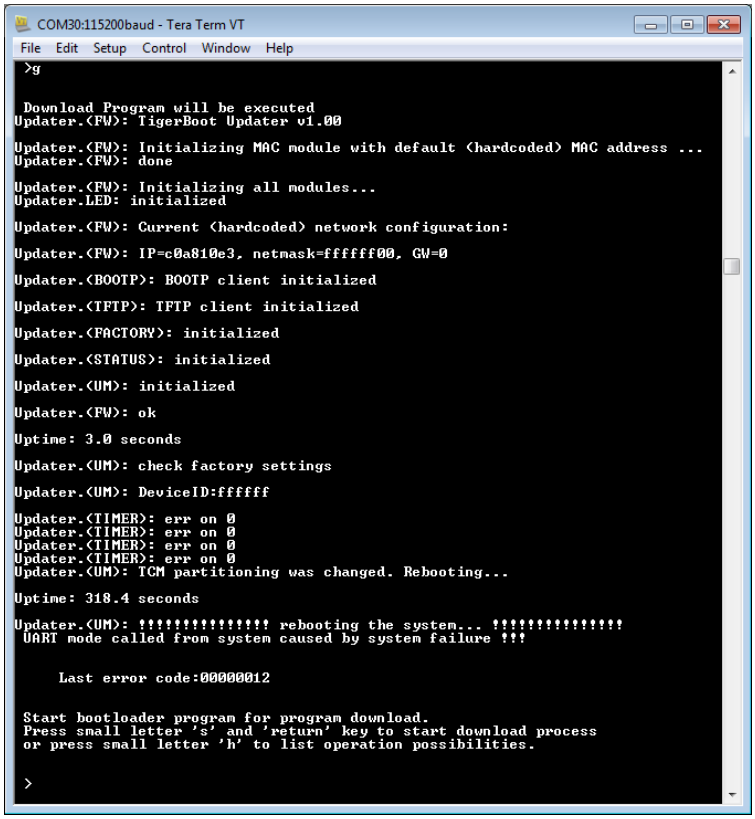
Successful download is confirmed with the message in Figure 6-4.

Figure 6-4: TPS-1 download completion



Next press <g> to execute the downloaded program. The flash will now be erased and the TPS-1 will automatically be rebooted. It will then report error message 0x00000012 which stands for an empty flash as shown in Figure 6-5.

Figure 6-5: UART messages while serial flash is erased



You can now continue to bring up the board again according to the instructions in Chapter 2.3. After that, please continue with Chapter 3 of this manual (R21UT0247ED0102).

For the “Normal” (i.e., non-low-cost) TPS-1 Solution Kit (YCONNECT-IT-TPS1), please follow the <Getting started> document for YCONNECT-IT-TPS1.

Note: The latest version of the <Getting Started> document for YCONNECT-IT-TPS1 (R21UT0221EDxxxx) is included in the CD that comes with the TPS-1 Low-cost Solution kit YCONNECT-IT-TPS1L.

Revision History

Document number	Location	Revised item
R21UT0247ED0100		First edition
R21UT0247ED0101		2nd minor update changed the SPI connection in Table 2-1 update the file names various minor changes (typos etc.)
R21UT0247ED0102		3rd minor update update Chapter 2.2 and 2.3, re-arrange chapters, moved Chapter 3.3 to Chapter 5 and Chapter 5 to Chapter 6 various minor changes (typos etc.)

TPS-1 Low-cost Solution Kit

Renesas Electronics Corporation

R21UT0247ED0102