# PTX IoTRD Library for Renesas Synergy<sup>™</sup> Platform MCUs

Panthronics AG - IoT Reader Support



Exported on 05/03/2021



## Table of Contents

1 Introduction	5
2 Requirements	6
3 Hardware setup	7
3.1 PTX Evalboard 1v3	8
3.1.1 Purpose	8
3.1.2 Power supply	8
3.1.3 USB Interface	8
3.1.4 Interface switching	9
3.1.5 PMOD	9
4 PTX IoT Reader library	10
4.1 IoT-Reader API Description	10
4.1.1 ptxIoTRd_Init	10
4.1.2 ptxIoTRd_Update_ChipConfig	10
4.1.3 ptxIoTRd_Get_Revision_Info	11
4.1.4 ptxIoTRd_Initiate_Discovery	11
4.1.5 ptxIoTRd_Get_Card_Registry	12
4.1.6 ptxIoTRd_Activate_Card	12
4.1.7 ptxIoTRd_Data_Exchange	13
4.1.8 ptxIoTRd_Bits_Exchange_Mode	13
4.1.9 ptxIoTRd_Bits_Exchange	14
4.1.10 ptxIoTRd_RF_PresenceCheck	14
4.1.11 ptxIoTRd_T5T_IsolatedEoF	15
4.1.12 ptxIoTRd_T3T_SENSFRequest	15
4.1.13 ptxIoTRd_Reader_Deactivation	16
4.1.14 ptxIoTRd_Update_ChipConfig	16
4.1.15 ptxIoTRd_Set_Power_Mode	17
4.1.16 ptxIoTRd_Get_System_Info	17
4.1.17 ptxIoTRd_SWReset	
4.1.18 ptxIoTRd_Deinit	18
4.1.19 ptxIoTRd_Get_Status_Info	19



5 PTX IoT Reader library integration	
5.1 Add include path	20
5.2 Adding library file	21
5.3 Platform-specific implementation	22
5.4 Project configuration in e <sup>2</sup> studio	23
5.4.1 SPI	23
5.4.2 External IRQ	24
5.4.3 Timer IRQ	25
6 PTX IoT demo application	
6.1 Environment	26
6.2 Project structure	26
6.3 Importing the demo project	27
6.4 Running the PTX IOT reader demo application	
6.4.1 Preparing the debug configuration	
6.4.2 Flashing and running/debugging	





## 1 Introduction

This document describes building the PTX IoT demo application using **Renesas e<sup>2</sup> studio** and running it on the Renesas Synergy TB-S3A1 Evaluation board.

To enable using the library on a **broad range of Synergy platform** devices, the demo package contains precompiled libraries for ARM Cortex-M0+ and Cortex-M4 architectures.

The following topics are covered in this document:

- Introduction to PTX\_EvalBoard 1v3
- Overview of PTX IoT library
- PTX IoT library integration
- PTX IoT demo app



## 2 Requirements

The application has the following hardware requirements:

- Renesas Synergy TB-S3A1 kit
- PTX Evalboard 1v3
- USB type C cable
- Micro USB cables 2pcs

Building and running the application require the following tools and software to be pre-installed:

- e<sup>2</sup> studio (tested with version 7.8.0)
- SSP Distribution (tested with version 1.7.5)
- Toolchain GCC ARM Embedded Version 7.2.1.20170904
- PuTTY or similar terminal application



## 3 Hardware setup

1. Make sure that the SIF1 and SIF2 switches from PTX Evalboard are set to the interface type "SPI"



2. Connect the PTX Evalboard to TB-S3A1 PMOD A connector



3. Plugin both micro USB cables



- 4. Plugin the Type C USB cable to supply the boards
- 5. Flash the firmware using the onboard debugger
- 6. Use PuTTY to see the application output

NOTES:

- If the USB port on the PC side can supply 900mA (i.e a USB-C or USB-3 port) the PTX Evaluation Board can achieve its maximum output power.
- The Type C USB cable will supply also the TB-S3A1 3v3 channel, if the pmod 3v3 bridge jumper is connected, no other power supply is needed.
- When the debugger cable is connected, the TB-S3A1 board will supply also the PTX Evalboard if the pmod 3v3 bridge jumper is connected, but it will have a low performance.

## 3.1 PTX Evalboard 1v3

The PTX Evalboard has the purpose of evaluating PTX100R performance.

## 3.1.1 Purpose

The PTX Evalboard shall serve to demonstrate the performance of PTX100R for IoT application: Full support for all technologies (A, B, F, V)



## 3.1.2 Power supply

The PTX Evalboard is powered via a USB-C connection.

If the USB port on the PC side can supply 900mA (i.e a USB-C or USB-3 port) the Evalboard can achieve its maximum output power.

The PTX100R can be operated in two voltage settings (3.7V & 5.4V) which are selected by the switch on the left side of the board.

## 3.1.3 USB Interface

The USB Interface is handled by an FTDI FT231X USB/UART bridge.



Driver installation is automatic on Windows 10.

## 3.1.4 Interface switching

The evaluation board allows to switch between the three interfaces supported by the PTX100R IC using the SIF1 / SIF2 switches:



SIF1	SIF2	Interface
0	0	SPI
0	1	UART → USB
1	0	I2C

For the PTX IOT demo app, the SPI interface must be selected.

SPI and I2C are connected to a 2x5pin header on the bottom side of the board:



## 3.1.5 PMOD

SPI is also available via a PMOD 2x6pin connector. The PMOD connector allows connecting to a multitude of MCU demo boards.



The 3.3V supply for the host MCU can also be provided by the Demo Board if the jumper (PMOD 3v3 bridge) is placed



## 4 PTX IoT Reader library

PTX IoT Reader library provides an API for a set of functions:

- initialize the IOTRD API and NSC Stack
- initialize the PTX100R chip
- · select a specific card in case multiple cards and/or protocols were discovered
- retrieve card details like technical and/or activation parameters etc.
- exchange RF-data and -bitstreams
- stop RF-communication

## 4.1 IoT-Reader API Description

This chapter contains an overview of the functions provided by the IoT-Reader API.

**Note:** A detailed description of all functions including parameters and types can be found in the file ptx\_IOT\_READER.h.

## 4.1.1 ptxIoTRd\_Init

#### Declaration

ptxStatus\_t ptxIoTRd\_Init( ptxIoTRd\_t iotRd, ptxIoTRd\_InitPars\_t initParams);

#### Description

Initialize software and hardware components for IoT-Reader operation. This function has to be called before any other API functions. It performs software initialization and configuration for PTX100R.

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component (to be allocated by user).
initParams	NSC initialization parameters.

#### **Return Value**

Status Success or failure, refer to ptxStatus\_t for details.

## 4.1.2 ptxIoTRd\_Update\_ChipConfig

#### Declaration

1 ptxStatus\_t ptxIoTRd\_Update\_ChipConfig ( ptxIoTRd\_t iotRd, uint8\_t nrConfigs, ptxIoTRd\_ChipConfig\_t configParams);

#### Description

Updates the RF- and System-Configuration parameters of the NFC hardware. This function allows to change RF- and System-Configuration parameters at runtime.



#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component.
nrConfigs	Number of RF-/System-configurations to set
configParams	Pointer to n-configuration parameters sets

#### **Return Value**

Status Success or failure, refer to ptxStatus\_t for details.

### 4.1.3 ptxIoTRd\_Get\_Revision\_Info

#### Declaration

ptxStatus\_t ptxIoTRd\_Get\_Revision\_Info ( ptxIoTRd\_t iotRd, ptxIoTRd\_RevisionType\_t revisionType, uint32\_t revisionInfo);

#### Description

Get various revisions of system (C-Stack, DFY-Code/-Toochain, Chip-ID, Local changes etc.

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component (to be allocated by user).
revisionType	Revision type
revisionInfo	Pointer to variable holding revision information

#### **Return Value**

Status Success or failure, refer to ptxStatus\_t for details.

## 4.1.4 ptxIoTRd\_Initiate\_Discovery

#### Declaration



#### Description

This function starts the RF-Discovery procedure as defined in the NFC-Forum.

#### **Input Parameters**



Name	Description
iotRd	Pointer to stack component.
discover	config Pointer to RF-Discovery structure (if set to NULL-default values will be used).

#### **Return Value**

Status Success or failure, refer to ptxStatus\_t for details.

## 4.1.5 ptxIoTRd\_Get\_Card\_Registry

#### Declaration



#### Description

Access the internal card registry. This function can be used to access the internal card registry to retrieve a cards detailed information.

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component.
cardRegistry	Pointer to pointer to keep reference to card registry

#### **Return Values**

Status Success or failure, refer to ptxStatus\_t for details.

## 4.1.6 ptxIoTRd\_Activate\_Card

#### Declaration



#### Description

Selects / Activates a given card in case of multiple available cards

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component.
cardParams	Pointer to card (within registry) to select / activate



Name	Description
protocol	RF-protocol to activate

#### **Return Value**

Status Success or failure, refer to ptxStatus\_t for details.

## 4.1.7 ptxIoTRd\_Data\_Exchange

#### Declaration

1 ptxStatus\_t ptxIoTRd\_Data\_Exchange ( ptxIoTRd\_t iotRd, uint8\_t tx, uint32\_t txLength, uint8\_t rx, uint32\_t rxLength, uint32\_t msAppTimeout);

#### Description

Protocol-based or raw RF data exchange

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component.
tx	Pointer to buffer holding data to send
txLength	Length of data to send
rx	Pointer to buffer holding received data
rxLength	Size of buffer holding received data / length of received data
msAppTimeout	Application timeout

#### **Return Values**

Status Success or failure, refer to ptxStatus\_t for details.

## 4.1.8 ptxIoTRd\_Bits\_Exchange\_Mode

#### Declaration

ptxStatus\_t ptxIoTRd\_Bits\_Exchange\_Mode ( ptxIoTRd\_t iotRd, uint8\_t enable);

#### Description

Enables/Disables the bit-exchange mode required to call ptxIoTRd\_Bits\_Exchange.

#### **Input Parameters**



Name	Description
iotRd	Pointer to stack component.
enable	Enable/Disable flag

#### **Return Values**

Status Success or failure, refer to ptxStatus\_t for details.

## 4.1.9 ptxIoTRd\_Bits\_Exchange

#### Declaration



#### Description

Exchanges a bitstream based on NFC-A technology

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component.
tx	Pointer to buffer holding data bytes to send
txPar	Pointer to buffer holding parity bits to send
txLength	Length of tx- and txPar-buffers
rx	Pointer to buffer holding received bytes
rxPar	Pointer to buffer holding received parity bits
rxLength	Length of received bytes / parity bits
numTotBits	Total number of received bits
msAppTimeout	Application timeout

#### **Return Value**

Status of operation Success or failure, refer to ptxStatus\_t for details.

### 4.1.10 ptxIoTRd\_RF\_PresenceCheck

#### Declaration



#### Description

Executes a presence check method on ISO-DEP cards or NFC-DEP targets

#### Input Parameters

Name	Description
iotRd	Pointer to stack component.
presCheckType	Presence-check method type.

#### **Return Value**

Status of operation

## 4.1.11 ptxIoTRd\_T5T\_IsolatedEoF

#### Declaration

ptxStatus\_t ptxIoTRd\_T5T\_IsolatedEoF ( ptxIoTRd\_t iotRd, uint8\_t rx, uint32\_t rxLength, uint32\_t msAppTimeout);

#### Description

Sends an EoF-packet according to T5T protocol

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component.
rx	Pointer to buffer holding received bytes
rxLength	Size of buffer holding received data / length of received data
msAppTimeout	Application timeout

#### **Return Value**

Status of operation

## 4.1.12 ptxloTRd\_T3T\_SENSFRequest

#### Declaration

ptxStatus\_t ptxIoTRd\_T5T\_IsolatedEoF ( ptxIoTRd\_t iotRd, uint16\_t systemCode, uint8\_t requestCode, uint8\_t tsn, uint8\_t rx, uint32\_t rxLength, uint32\_t msAppTimeout);



#### Description

Sends a SENSF\_REQ-packet according to T3T protocol.

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component.
systemCode	T3T System-code
requestCode	T3T Request-code
tsn	T3T Number of timeslot(s)
rx	Pointer to buffer holding received bytes
rxLength	Size of buffer holding received data / length of received data
msAppTimeout	Application timeout

#### **Return Value**

Status of operation

## 4.1.13 ptxIoTRd\_Reader\_Deactivation

#### Declaration

#### Description

Stops any finished RF-communication and deactivates the reader and / or the remove device.

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component.
deactivationType	Type of deactivation (IDLE, DISCOVERY, Sleep)

#### **Return Value**

Status of operation

## 4.1.14 ptxloTRd\_Update\_ChipConfig

#### Declaration



#### ptxStatus\_t ptxIoTRd\_Update\_ChipConfig ( ptxIoTRd\_t iotRd, uint8\_t nrConfigs, ptxIoTRd\_ChipConfig\_t configParams);

#### Description

Updates RF- and System-parameters at runtime.

#### Input Parameters

Name	Description
iotRd	Pointer to stack component.
nrConfigs	Number of RF-/System-configurations to set
configParams	Pointer to n-configuration parameters sets

#### **Return Value**

Status of operation

### 4.1.15 ptxIoTRd\_Set\_Power\_Mode

#### Declaration

1	<pre>ptxStatus_t ptxIoTRd_Set_Power_Mode ( ptxIoTRd_t iotRd, uint8_t</pre>
	newPowerMode);

#### Description

Puts chip into stand-by or wakes it up from stand-by

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component.
newPowerMode	Type of stand-by operation

#### **Return Value**

Status of operation

## 4.1.16 ptxIoTRd\_Get\_System\_Info

#### Declaration

1	ptxStatus_t ptxIoTRd_Get_System_Info (    ptxIoTRd_t iotRd, ptxIoTRd_SysInfoType_t infoType, uint8_t infoBuffer, uint8_t infoBufferLength);
Description	



#### Enables / Disables immediate writing to a given log file

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component
infoType	Information identifier
infoBuffer	Buffer to store information
infoBufferLength	Length of information

#### **Return Value**

Status of operation

## 4.1.17 ptxIoTRd\_SWReset

#### Declaration

1	nty Status t nty IsTDd SulDeset	
1	preserver preserver	. ( prxioika_t iotka);

#### Description

Performs a soft-reset of the PTX100R

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component.

#### **Return Value**

Status Success or failure, refer to ptxStatus\_t for details.

## 4.1.18 ptxIoTRd\_Deinit

#### Declaration



#### Description

Close the IoT-Reader Component. This function closes the IOT and releases the resources used. It must be called as the last function before the stop of the library usage.

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component.
<b>.</b>	

#### **Return Value**



Status Success or failure, refer to ptxStatus\_t for details.

## 4.1.19 ptxIoTRd\_Get\_Status\_Info

#### Declaration



#### Description

#### Retrieves current operating state of chip

#### **Input Parameters**

Name	Description
iotRd	Pointer to stack component
statusType	Status type identifier
systemState	Pointer to variable holding
system	state

#### **Return Value**

Status of operation



## 5 PTX IoT Reader library integration

The precompiled libraries for the supported architectures can be found in the PtxIot/lib subfolder. The appropriate one of them for the particular MCU must be fed to the linker during build time.

Even without using an IDE, it is very easy to work with the library. The following prerequisites must be met:

- the "C" or "C++" **compiler** shall be able to find the include files in the inc folder. (Use the -/ option)
- the **linker** shall be able to find the library file. (Use the -*L* option)
- the **linker** needs to be told explicitly to use the library. (Use the -*l* option)

With these rules kept in mind, it is easy to employ the library in any codebase with any build system.

This chapter explains the steps to add the library to an **e<sup>2</sup> studio** project.

## 5.1 Add include path

For the compiler to find the header (.h) files containing the API functions provided by Panthronics AG, the folder PtxIot/inc needs to be added to the list of user defined include directories. This can be done by navigating to **Project menu**  $\rightarrow$  **Properties**  $\rightarrow$  **C/C++ Build**  $\rightarrow$  **Settings**  $\rightarrow$  **Tool Settings**  $\rightarrow$  **GNU ARM Cross Compiler**  $\rightarrow$  **Includes**. Clicking on the **Add...** button on the right side of the small toolbar and using **Workspace** button in the popup window, we can navigate to the folder and click **OK**.



Settings Settings C/C++ Build Build Variables Environment Logging Settings Tool Chain Edit: C/C++ General Git MCU Project References Renesa Q E Run/Debug Settin Select one or more Workspace Joc/\$(ProjName)/src/PLA1)* Select one or more Workspace Folders Select one or more Workspace	a filter text					
Resource Builders CCr+s Build Build Variables Environment Logging Settings Tool Settings © Toolchain & Build Steps © Build Artifact © Binary Parsers © Error Parsers Tool Chain Edit © Col Settings © Toolchain & Build Steps © Build Artifact © Binary Parsers © Error Parsers Tool Chain Edit © Col Settings © Toolchain & Build Steps © Build Artifact © Binary Parsers © Error Parsers Tool Chain Edit © Col Settings © Toolchain & Build Steps © Build Artifact © Binary Parsers © Error Parsers Tool Chain Edit © Col Settings © Toolchain & Build Steps © Build Artifact © Binary Parsers © Error Parsers Tool Chain Edit © Col Settings © Toolchain & Build Steps © Build Artifact © Binary Parsers © Error Parsers Tool Chain Edit © Col Settings © Toolchain & Build Steps © Build Artifact © Binary Parsers © Error Parsers Tool Settings © Toolchain & Build Steps © Build Artifact © Binary Parsers © Error Parsers Tool Settings © Toolchain & Build Steps © Build Artifact © Binary Parsers © Error Parsers Tool Settings © Tool Settings © Toolchain & Build Steps © Binary Parsers © Error Parsers © Debugging © Settings © Bit Mult AEML Corece Accemption Setime © Cok Cancel Workspace _ CrS(ProjName)/Stol/inc) Tory = Steps = Stol © Cok Cancel Workspace _ File system © Cok Cancel File System Parser Defaults Apply	be filter text	Settings				⇔ ▼ ⇔ ▼
Build Variables Environment Logging Settings Tool Chain Edit C/C++ General Git MCU Project References Renesas QE Raw/Debug Settin Task Repository Select one or more Workspace Joc/\$(ProjName)/src/PLAT)* Select one or more Workspace Folders Select one or more Workspace Fo	Resource Builders C/C++ Build	Configuration: Release	e [Active]		∽ Ma	nage Configuration
Settings Tool Chain Edite C/C++ General Bit MCU Project References Renesas QE Run/Debug Settin Task Repository Task Tags Viriation © OK Cancel Project References Restore Defaults Apply Restore Defaults Apply	Build Variables Environment Logging	Tool Settings Too	olchain 🎤 Build Steps	😤 Build Artifact 📓 Binary F	Parsers <sup>10</sup> Error Parsers	5
Tool Chain Editc C/C++ General Warnings Debugging Debugging C/C++ General Git WU Project References Renesas QE Run/Debug Settin Task Repository: Sworkspace_loc/\$(ProjName)/src/Synergy_gen)" S(workspace_loc/\$(ProjName)/src/Synergy_gen)" S(workspace_loc/\$(ProjName)/src/Synergy_gen)" S(workspace_loc/\$(ProjName)/src/Synergy_gen)" S(workspace_loc/\$(ProjName)/src/Synergy_gen)" S(workspace_loc/\$(ProjName)/refunction) Select one or more Workspace Folders Debug Polot Debug Cancel Workspace File system Restore Defaults Apply	Settings	Target Processor		Include paths (-I)		🔊 🔊 😨 🤴
C/C++ General Warnings Debugging Store Child ABM Croce Accombiant Store Child ABM Croce Accombiant Stor	Tool Chain Edito	🖉 Optimization		"\${workspace_loc:/\${ProjNa	ame}/Ptxlot/inc}"	· · · · · · · · · · · · · · · · · · ·
Sit & Debugging `StWorkspace_loc/\$(ProjName)/src/spregy_gen)" MCU Project References Renesas QE Run/Debug Settin Task Repository S (workspace_loc/\$(ProjName)/Src/spregy_gen)" S (workspace_loc/\$(ProjName)/Src/spregy_gen)" S (workspace_loc/\$(ProjName)/Src/spregy_gen)" S (workspace_loc/\$(ProjName)/Src/spregy_gen)" S (workspace_loc/\$(ProjName)/Ptxlot/inc) S	C/C++ General	避 Warnings		"\${workspace_loc:/\${ProjNa	ame}/src/PLAT/S3A1}"	
WCU       Image: Selection of the	Git	Debugging		"\${workspace_loc:/\${ProjNa "\$/workspace_loc:/\${ProjNa	ame}/src/PLAT}	
Project References   Birectory pair   Birectory:   \$\u00ed Morkspace_loc:/\$(ProjName)/Pbdot/inc)   ask Raps   \$\u00ed Morkspace_loc:/\$(ProjName)/Pbdot/inc)   ask Raps   \$\u00ed Morkspace Folders   \$\u00ed Bebug   \$\u00ed Bebug <tr< td=""><td>MCU</td><td>e<sup>2</sup> Add directory nath</td><td>Accomblar</td><td>athorkspace_ioc./ atrioin</td><td>×</td><td>fg/bsp}"</td></tr<>	MCU	e <sup>2</sup> Add directory nath	Accomblar	athorkspace_ioc./ atrioin	×	fg/bsp}"
elenesas QE   birectory:   \$\u00ed{Workspace_loc:/\${ProjName}/Ptxlot/inc}   \$\u00ed{Workspace_loc:/\${ProjName}/Ptxlot/inc}   \$\u00ed{Folder selection   \$\u00ed{Vorkspace_loc}   \$\u00ed{Folder selection   \$\u00ed{Vorkspace_loc}	Project References	Add directory path			~	fg/driver}"
sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp)" sp) sp) sp) sp) sp) sp) sp) sp) sp) sp)	Renesas QE	Directory:				
ask Repository ask Tags	un/Debug Settin	the second secon	(Newse) (Dealer Africa)			sp}"
Inver/api,	ask Repository	\${workspace_loc:/\${Pro	pjName}/Ptxlot/inc}			sp/cmsis/Include
Selection     Selection <td>ask idgs</td> <td></td> <td></td> <td></td> <td></td> <td>river/instances)"</td>	ask idgs					river/instances)"
* © Ptxlot   > © doc   > © inc   > © lib   Flash Image ze   ? OK Cancel   Restore Defaults Apply	Select one or n	nore Workspace Folders				amowork /al/tv\"
Se cfg Se doc Se inc Se lib Flash Image Ze Restore Defaults Apply	> 🧼 Debug		Cancella	el Workspace	File system	🕘 ଲା ଭା ଚିମ 🖇
Se doc Se inc Se lib Flash Image Ze OK Cancel Restore Defaults Apply	> > Debug > > Ptxlot		↑ Canc	el Workspace	File system	A 2 5 4
Flash Image ize       OK     Cancel         Restore Defaults     Apply	> 🍅 Debug 🕆 🗁 Ptxlot > 🗁 cfg		∧ Canc	el Workspace	File system	
OK       Cancel         OK       Cancel         Restore Defaults       Apply	> Debug > Ptxlot > Cfg > doc		Canc	el Workspace	File system	<u>କ</u> ିକ୍ଷ ଭାତ୍ତି (
OK     Cancel       Restore Defaults     Apply	> Debug > Ptxlot > pcfg > doc > conc > conc > conc		Cance Flash Image	el Workspace	File system	<u>କ</u> ା କ୍ଷା ଲା ହା (
Restore Defaults     Apply	> Debug > Ptxlot > pcfg > doc > poinc > plib		<ul> <li>Cance</li> <li>Flash Image</li> <li>ize</li> </ul>	el Workspace	File system	<b>ि</b> स्त्र छ हा (
Restore Defaults Apply	> Debug > Ptxlot > pcfg > poinc > poinc > poinc > poinc	OK Cancel	<ul> <li>Cance</li> <li>Flash Image</li> <li>ize</li> </ul>	el Workspace	File system	<u>ର</u> କ୍ଷ ହ ତା (
	> Debug > Ptxlot > pcfg > podoc > poinc > polic	OK Cancel	<ul> <li>Cancella</li> <li>Flash Image</li> <li>ize</li> </ul>	el Workspace	File system	<b>2</b> से 20 हा ई

## 5.2 Adding library file

There is an individual subdirectory in the PtxIot/lib folder for each supported MCU architecture, which contains the library file libPtxIot.a.

The library file, like the header files, also needs to be found by the linker. Similarly to the include folders' settings, the proper library folder must be added to the *Library search path* list (see bottom right pane on image).

Moreover we need to add the **Ptxlot** to the **Libraries** list. This will be rendered into *-lPtxlot* command line switch during linker invocation, which tells the linker to use the file libPtxlot.a.



e <sup>2</sup> Properties for iotr	d_s3a1_demo	- D X
type filter text	Settings	⇔ ▼ ⇔ ▼ ▼
<ul> <li>Properties for lot</li> <li>type filter text</li> <li>Resource Builders</li> <li>C/C++ Build Build Variables Environment Logging Settings Tool Chain Editc</li> <li>C/C++ General Git</li> <li>MCU Project References Renesas QE Run/Debug Settin</li> <li>Task Repository Task Tags</li> <li>Validation</li> </ul>	Settings Configuration: Release [Active] Tool Settings Toolchain P Build Steps Target Processor Optimization Warnings Debucaina Warnings Debucaina Warnings Debucaina Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Settings Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellaneous Miscellane	Anage Configurations          Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Image Configurations         Imag
	♥ SNU ARM Cross Print Size	
< >		Restore Defaults Apply
Ū.		Appry and Close Cancel

## 5.3 Platform-specific implementation

As the library is common for all implementations, it must not use any hardware dependent resources. This is where the PLAT module comes into play by defining the interface functions to be implemented by the user. The required function declarations can be found in ptxPLAT.h. These functions must be individually implemented for the actual hardware.

The ptxPLAT.c code contains the basic low-level functions, that the library will use for allocating memory area for objects, initializing and controlling timers, do communication across SPI, etc.



The **TB-S3A1** platform-specific implementation can be found in ptxPLAT.c, but thank to Renesas Synergy platform API, the implementation is compatible for any different Synergy MCU. In such case the device pin assignment might be the only configuration to be adjusted.

## 5.4 Project configuration in e<sup>2</sup> studio

To be able to implement PTX communication over the PMOD B connector, the following peripherals should be configured:

- SPI
- External IRQ
- Timer IRQ



## 5.4.1 SPI

- SPI channel 0
- SPI bitrate 1000000
- SPI callback NULL

Please note, the current implementation does not use IRQ for SPI.



∠ Common	
Parameter Checking	Default (BSP)
Module g_spi0 SPI Driver on r_rspi	
Name	g_spi0
Channel	0
Operating Mode	Master
Clock Phase	Data sampling on odd edge, data variation on even edge(CPHA=0)
Clock Polarity	Low when idle
Mode Fault Error	Disable
Bit Order	MSB First
Bitrate	10000000
Callback	NULL
SPI Mode	SPI Operation
Slave Select Polarity(SSL)	Active Low
Select Loopback1	Normal
Select Loopback2	Normal
Enable MOSI Idle	Disable
MOSI Idle State	MOSI Low
Enable Parity	Disable
Parity Mode	Parity Odd
Select SSL Level After Transfer	SSL Level Do Not Keep
Clock Delay Enable	Clock Delay Disable
Clock Delay Count	Clock Delay 1 RSPCK
SSL Negation Delay Enable	Negation Delay Disable
Negation Delay Count	Negation Delay 1 RSPCK
Next Access Delay Enable	Next Access Delay Disable
Next Access Delay Count	Next Access Delay 1 RSPCK
Receive Interrupt Priority	Priority 5
Transmit Interrupt Priority	Priority 5
Transmit End Interrupt Priority	Priority 5
Error Interrupt Priority	Priority 5
Byte Swap(Only for S5 series MCU's)	Disable

SPI channel should also be enabled from PINS/Peripherals/Connectivity:SPI/ SPI0

## 5.4.2 External IRQ

- IRQ trigger Rising
- IRQ callback ptx\_IRQ



Property	Value
▲ Common	
Parameter Checking	Default (BSP)
<ul> <li>Module g_external_irq External IRQ Driver on r_icu</li> </ul>	
Name	g_external_irq
Channel	1
Trigger	Rising
Digital Filtering	Disabled
Digital Filtering Sample Clock (Only valid when Digital Filte	PCLK / 64
Interrupt enabled after initialization	True
Callback	ptxIRQ
Pin Interrupt Priority	Priority 12

## 5.4.3 Timer IRQ

- Timer period 10 ms
- Timer callback ptxTimerIRQ

Property	Value
✓ Common	
Parameter Checking	Default (BSP)
<ul> <li>Module g_timer Timer Driver on r_gpt</li> </ul>	
Name	g_timer
Channel	0
Mode	Periodic
Duty Cycle Range (only applicable in PWM mode)	Shortest: 2 PCLK, Longest: (Period - 1) PCLK
Period Value	10
Period Unit	Milliseconds
Duty Cycle Value	50
Duty Cycle Unit	Unit Raw Counts
Auto Start	True
GTIOCA Output Enabled	False
GTIOCA Stop Level	Pin Level Low
GTIOCB Output Enabled	False
GTIOCB Stop Level	Pin Level Low
Callback	ptxTimerlRQ
Overflow Interrupt Priority	Priority 12



## 6 PTX IoT demo application

PTX IoT demo application is meant to demonstrate the performance of the PTX100R for IoT applications, offering full support for all technologies (A, B, F, V).

This application implements all steps required to initialize the PTX100R, to discover, activate, communicate and to deactivate a tag as described in the chart below:



## 6.1 Environment

In this document we are going to use the **e<sup>2</sup> studio 7.8** as our code editor and our target is the **TB-S3A1** evaluation board with an ARM cortex-m4 microcontroller having a hardware floating point unit.

The current demo package contains precompiled libraries for ARM Cortex-M0+ and Cortex-M4 architectures. To use an MCU of a currently unsupported architecture, another precompiled library is required, which is built for exactly the particular hardware. For more information, please contact our support.

## 6.2 Project structure

File/Folder	Description
Ptxlot/	PTX IoTRD library
Ptxlot/ptx_IOT_READER.h	IoT demo app API available with the PTX IoTRD library
src/PLAT/	HAL configuration for communication: SPI, Timer, and IRQ
src/consoleThread_entry.c	Thread containing the implementation of Console Framework over USB



File/Folder	Description
src/ptxThread_entry.c	Thread containing the implementation of the PTX IOT demo application

## 6.3 Importing the demo project

To get started with the demo project quickly, it needs to be imported in  $e^2$  studio. This can be done by **File**  $\rightarrow$  **Import...**  $\rightarrow$  **Existing Projects into Workspace** and selecting the folder the archive had been extracted to.

e <sup>2</sup> Import	_	
Import Projects Select a directory to search for existing Eclipse projects.		<u> </u>
Select root directory: C:\Work\iotrd_s3a1_demo_	~	B <u>r</u> owse
○ Select <u>a</u> rchive file:	~	Browse
Projects:		
iotrd_s3a1_demo (C:\Work\iotrd_s3a1_demo_)		<u>S</u> elect All
		Deselect All
		R <u>e</u> fresh
Options Search for nested projects Copy projects into workspace Hide projects that already exist in the workspace		
Working sets		
Add project to working sets		Ne <u>w</u>
Working sets:	×	S <u>e</u> lect
? < Back Next > Einish	5	Cancel

After importing, the project is ready for compiling. With selecting **Project**  $\rightarrow$  **Build All** from the menu, the compiling process will start and a similar table will summarize the memory usage after a successful build:

1	arm-none-	eabi-si:	zefor	mat=berk	eley "io	otrd_s3a1_demo.elf"
2	text	data	bss	dec	hex	filename
3	97120	635	40608	138363	21c7b	iotrd_s3a1_demo.elf



## 6.4 Running the PTX IOT reader demo application

Once the project is compiled, it needs to be flashed into the MCU and run.

## 6.4.1 Preparing the debug configuration

Since the debug configuration is not part of the project, it cannot be imported, therefore a new one needs to be created for the workspace used. In the **Run**  $\rightarrow$  **Debug Configurations...** menu. The TB-S3A1 evaluation board has a **SEGGER J-Link** on-board debug probe, which is supported by the **Renesas GDB Hardware Debugging** driver, therefore a new configuration needs to be created and configured according to the following images:

				15 -	_		
Create, manage, and run co	nfigurations			1	Create, manage, and run co	figurations	3
					Run Break Time Measurement O 10000.0MHz.	perating Frequency must not be empty and must be between 0	1.001 &
3 🕞 🗶 🖻 🕸 🔻	Name: demo				🖸 🗟 🗶 🖻 🔅 👻	Name: demo	
type filter text	Main & Debugger > Sta	rtun 🗖 Commor	Source		type filter text	Main & Debugger  Startup  Common  Source	
C/C++ Remote Application	Project:				C/C++ Remote Application A FASE Script	Debug hardware: J-Link ARM V Target Device: R7FS3A17	/C
GDB Hardware Debugging	iotrd_s3a1_demo			Browse	GDB Hardware Debugging		
GDB OpenOCD Debugging	C/C++ Application:				GDB OpenOCD Debugging	GDB Settings Connection Settings Debug Tool Settings	
GDB Simulator Debugging	Debug\iotrd_s3a1_demo.elf				GDB Simulator Debugging	GDB Connection Settings	
Java Applet		Variables	Search Project	Browse	Java Application	Autostart local GDB server Host name or IP addre	ess: localhost
Launch Group	Build (if required) before lau	nching			s Launch Group	GDB port number:	61234
Launch Group (Deprecated)	Build Configuration: Use Ac	tive		~	Launch Group (Deprecated)	GDB Command:	
Remote Java Application	O Enable auto build	C	Disable auto build		Remote Java Application	arm-none-eabi-gdb	Browse Variables.
Renesas GDB Hardware Del	Use workspace settings	Co	onfigure Workspace S	ettings	Tenesas Gob Hardware Der Tenesas Gob Hardware Der		
Renesas Simulator Debugg ¥					🗈 Renesas Simulator Debugg 🗡		
< >			0	4	< >	Revert	Apply
ilter matched 14 of 16 items			Ke <u>v</u> ert	Аррі <u>у</u>	Filter matched 14 of 16 items	THE KUT	white
-							
(?)			<u>D</u> ebug	Close	(f)	Debug	Close
					1		NT

Note: The Target Device needs to be set to the actual MCU used, it will not be taken over from the project.

## 6.4.2 Flashing and running/debugging

Finally, the debug session can be started by pressing the **Debug** button. As soon as the firmware is running on the board, a new **USB Serial Device** will be discovered by the operating system.



🛃 Device Manager – 🗆																							
<u>F</u> ile <u>A</u> ction <u>V</u> iew <u>H</u> elp																							
Internet in the second																							
> 🖵 Network adapters																							
> E Portable Devices																							
🗸 🛱 Ports (COM & LPT)																							
<ul> <li>IKALOGIC COM PORT (COM20)</li> <li>Standard Serial over Bluetooth link (COM10)</li> </ul>																							
												<ul> <li>Standard Serial over Bluetooth link (COM9)</li> <li>USB Serial Device (COM11)</li> <li>USB Serial Port (COM1)</li> </ul>											
> 📇 Print queues																							
<	>																						

A new connection can be established using PuTTY (or any other app) by connecting to this serial port using a baud rate of 115200.

If ? is sent to the console, the following message should appear

8	Сом	12 - PuTTY												_		×	(
>?	Help	Menu															
	:	PTX100R	S3A1	IOT	Demo	v1.0.0	place	a	NFC	tag	in	the	RF	field			

If an NFC card is present in the field, the details will be read and data exchange will be triggered as shown in the below picture.



# **Copyrights & Disclaimer**



The material herein may not be reproduced, adapted, merged, translated, stored, or used without the priorwritten consent of the copyright owner. Devices sold by Panthronics AG are covered by the warranty and patentindemnification provisions appearing in its General Terms of Trade. Panthronics AG makes no warranty, express, statutory, implied, or by description regarding the information set forth herein. Panthronics AG reserves the right to change specifications and pricesat any time and without notice. Therefore, prior to designing this product into a system, it is necessary to check with Panthronics AG for the most up to date information. Applications requiring extended temperature range, unusual environmental requirements, orhigh reliability applications, such as military, medicallife-support or life-sustaining equipment are notrecommended. This product is provided by Panthronics AG "AS IS" and any express orimplied warranties, including, but notlimited to the implied warranties of merchantability and fitnessfor a particular purpose are disclaimed. Panthronics AG shall not be liable to recipient or any third party for anydamages, including but not limited to personal injury, propertydamage, loss of profits, loss of use, interruption of business orindirect, special, incidental or consequential damages, of anykind, in connection with or arising out of the furnishing, performance or use of the technical data herein.

Legal Notice - Purchase of Panthronics ICs with MIFARE Classic<sup>®</sup> compatibility:

PTX100R IC offers modes to be compatible with MIFARE Classic<sup>®</sup> RFID tags, allowing to build MIFARE Classic<sup>®</sup> compatible reader systems. MIFARE<sup>®</sup> and MIFARE Classic<sup>®</sup> are trademarks of NXP B.V., High Tech Campus 60 NL-5656 AG EINDHOVEN, NL. Purchase of Panthronics' MIFARE Classic<sup>®</sup> compatible products does not provide a license of any NXP rights, in particular does not provide the right to use MIFARE<sup>®</sup> or MIFARE Classic<sup>®</sup> as a trademark to brand such systems.

Copyright Panthronics AG, Sternäckerweg 16, 8041, Graz