Introduction

The PN5180 supports secure firmware update and provides an easy way to upload the firmware via NFC Cockpit tool or by any application hosted on the microcontroller, which implements secure firmware update functionality.

This document describes the process how to upload the new version of the PN5180 FW on the PNEV5180B development board and how to prepare the firmware update software for any microcontroller connected to the PN5180 IC.

In the document descriptions “PNEV5180B” [1] board and LPCXpresso IDE [4] toolchain are used as a reference.

Getting started

This section describes the system and hardware requirements needed to upload the new version of the PN5180 FW.

2.1 System requirements

NFC Cockpit tool requirements:

- Installation is described in AN11744 [3]
- PC with USB port running on Microsoft Windows 7 operating system
- VCOM CDC drivers or libUSB drivers for ABEND interface (both drivers are available in installation package)

Firmware Download Library requirements:

- Running toolchain to develop embedded application, provided examples are prepared for the LPCXpresso IDE.

2.2 Hardware requirements

- Enabled SPI host connection between microcontroller and PN5180
- USB connection between PC and microcontroller

Note:
PN5180 Evaluation board provides all features required to test Secure FW Update and it is recommended to use it for the evaluation.

2.3 Block Diagram Overview

At a very high level, the system is divided into three parts.

1. PC Host
2. Microcontroller Host
3. PN5180

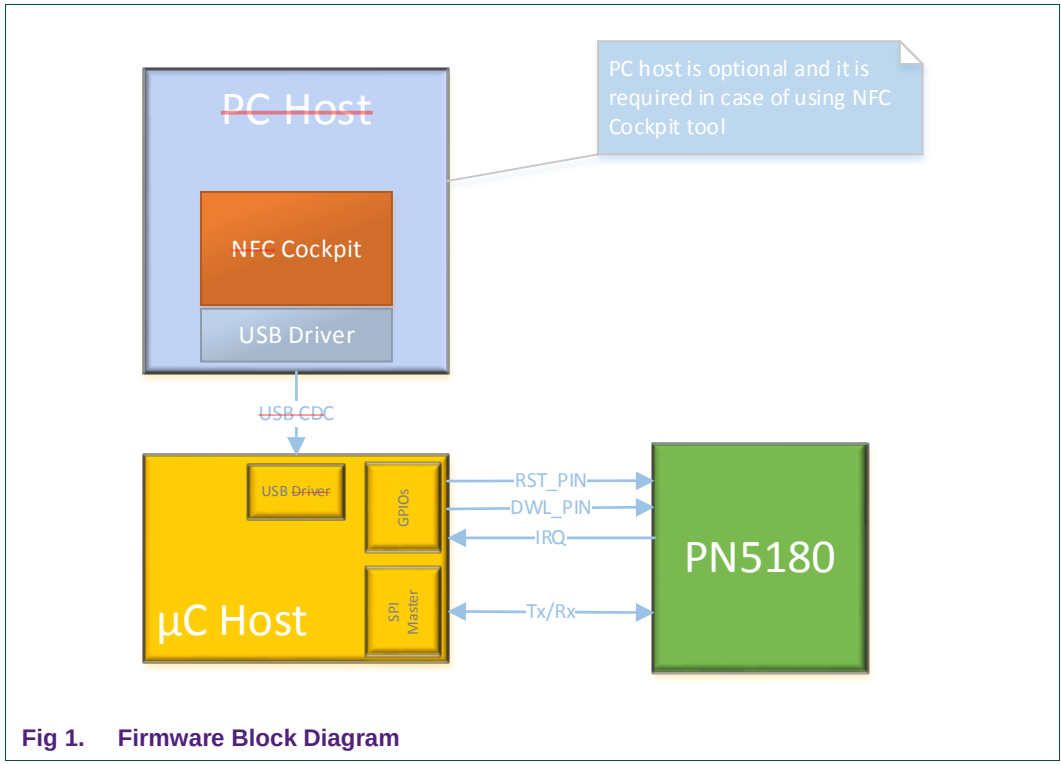


Fig 1. Firmware Block Diagram

2.3.1 PC Host

The PC is hosting NFC Cockpit tool and it should provide USB connection or RS232 Serial Interface.

The PC Host is connected to the Microcontroller host via a USB Serial VCOM interface. Where applicable, the PC Host may be connected to Micro Controller Host over a plain RS232 serial interface too.

PC Host is optional and it is required in case of using NFC Cockpit tool.

2.3.2 Microcontroller Host

Microcontroller, in this setup, works as a medium between PC and PN5180. The purpose of it is to receive a data from the PC over the USB interface and forward them to PN5180 via SPI interface and via versa.

In case of setup, where Secure FW update application is hosted on the Microcontroller Host, application holds firmware data and send them to PN5180 IC via SPI interface.

2.3.3 PN5180

High performance full NFC Forum-compliant frontend IC for contactless communication at 13.56 MHz PN5180 supports secure FW update and guidelines are described in next sections.

3. Upload Firmware via NFC Cockpit Application

The easiest way to upload the PN5180 FW is to use NFC Cockpit application. Detailed description how to install the NFC Cockpit can be found in “AN11744 - PNEV5180B Quick Start Guide” [3].

In this setup NFC Cockpit application, hosted on PC, send FW data to the µC, which reads data received through the USB interface and forward them to the PN5180 via SPI interface.

The figure below shows the high level setup of the required components.

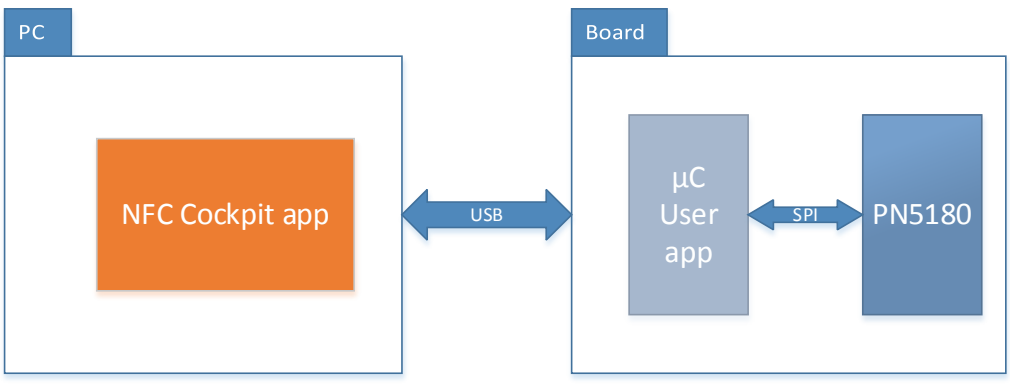


Fig 2. NFC Cockpit architecture

NXP provides Secure FW Update application prepared for the LPC1769 and it is available in the NFC Cockpit installation package.

Current version of the NFC Cockpit supports “Secure Firmware Upload” only with the provided application hosted on the LPC1769.

3.1 LPC Firmware for PNEV5180B Development Board

After successful NFC Cockpit installation, a SW package with LPC FW for the LPC1769 is available in the installation folder.

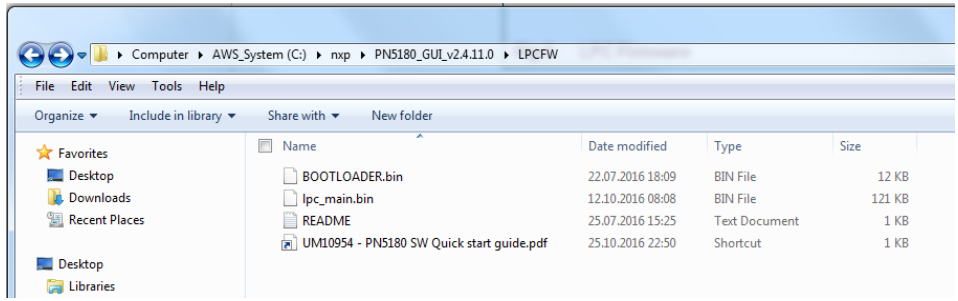


Fig 3. LPC Firmware

LPC firmware provides an implementation of the USB driver and implementation of the secure firmware update functionality. On the PNEV5180B development board, the LPC firmware is installed by default and it is ready to use.

In case the LPC firmware needs to be updated please follow guidelines described in AN11744 [3] chapter 3.

3.2 Update Firmware via NFC Cockpit Tool

PN5180 NFC Cockpit tool support secure PN5180 firmware update. After successful installation and start of the NFC Cockpit tool, the communication link between PC and PN5180 is establish automatically.

At startup NFC Cockpit tool check the firmware version of the PN5180 IC and compare it with the latest known version. In case the firmware should be updated the tool reminds with the message in the popup window, see figure below - Fig 4.

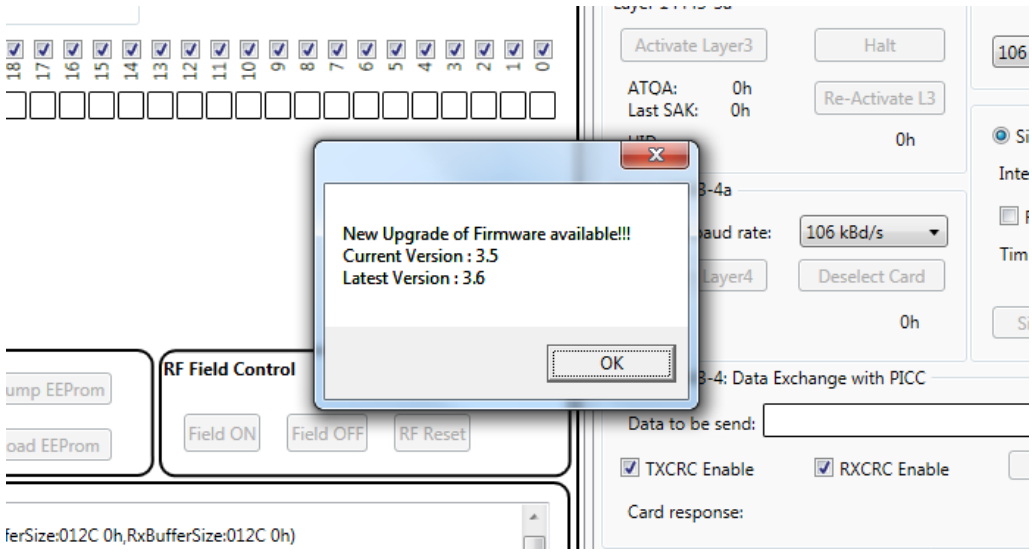
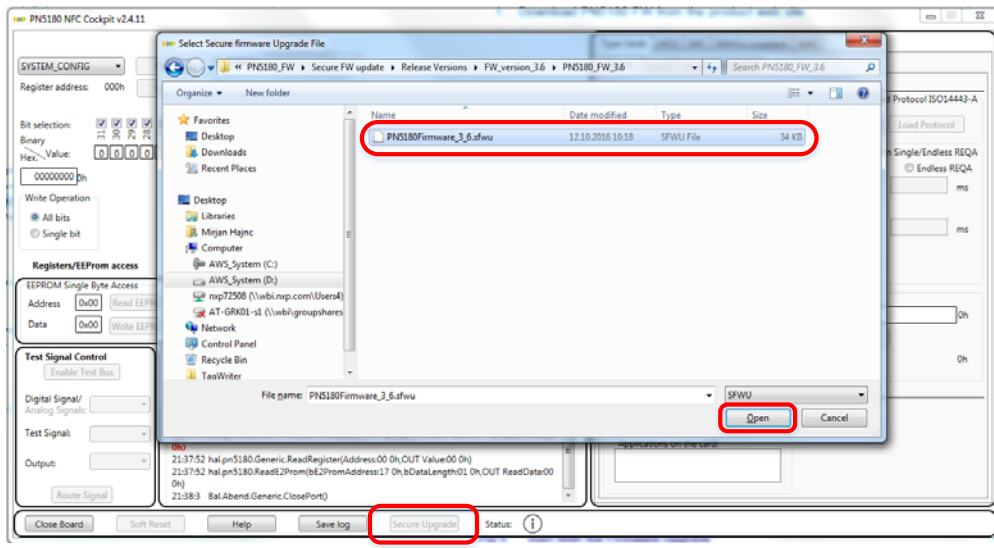


Fig 4. Application Message to Upgrade the Firmware

The latest versions of the PN5180 FW are available on the www.nxp.com on the PN5180 product page [5].

To start the firmware upload please follow guidelines described below:

1. Download PN5180 FW from the product web site
2. Extract downloaded zip file to an empty folder
3. Open NFC Cockpit tool and click “Secure Upgrade” button
4. Browse to the folder created in step 2 and select “*.*sfwu*” file
5. To start the FW upload click on “Open” button.



(1) Click on the “Secure Upgrade” button
(2) Browse to the folder with FW
(3) Select the firmware, e.g. “PN5180Firmware_3_6.sfwu”
(4) To start with the FW upgrade click on “Open” button

Fig 5. Start With the Firmware Upgrade

After that application will start with the FW upgrade.

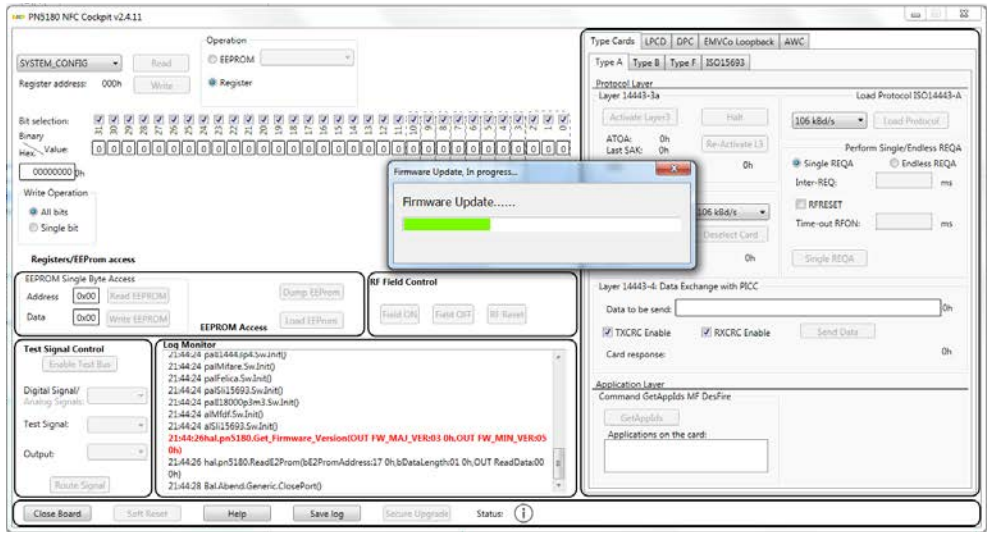


Fig 6. Firmware Upgrade Status

Restart PN5180 to finish installation, the latest version is printed in the Log window.

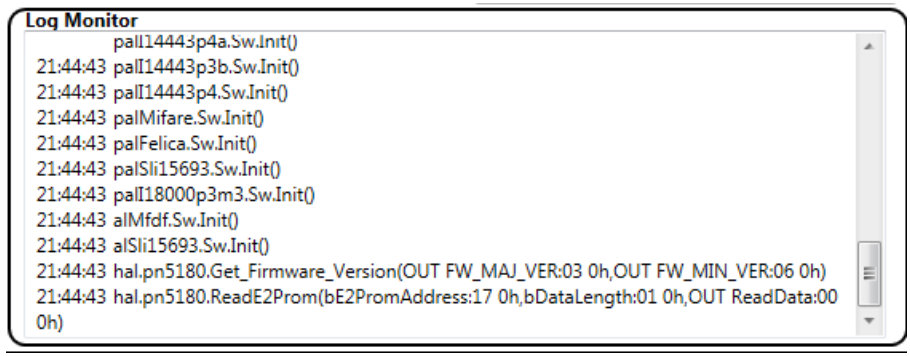


Fig 7. Current Version on the Firmware Printed in Log Monitor

Note:
The PN5180 FW contains all the EEPROM settings including all analog settings, the DPC calibration and the AWC/ARC settings for the standard PN5180 evaluation board with the standard 65mm x 65mm antenna. Any individual settings need to be saved (“Dump EEPROM”) before the PN5180 FW update and restored afterwards (“Load EEPROM”).

4. Upload PN5180 Firmware by Secure FW Update Application

In this setup Secure Firmware Update application contains raw PN5180 firmware data and send them to the PN5180 IC via SPI interface. Secure FW update application is hosted on the target microcontroller.

This chapter describes how to implement Secure Firmware Update application on the target host.

The figure below Fig 8 shows the high level architecture setup.

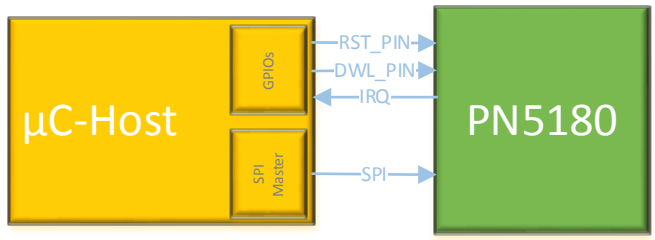


Fig 8. Secure Firmware Update Architecture

NXP provides a reference Secure FW Update application prepared for the LPC1769 and it is available on the PN5180 product page [5].

4.1 Secure FW Update Application

Secure FW Update application is hosted on the target microcontroller and contains PN5180 firmware data. Main task of it is to prepare data frames from provided raw data and send them to the PN5180 IC via SPI interface.

Before the implementation, it is necessary to download PN5180 firmware from the website [5], extract it and import header file with firmware data to the project.

Preconditions:

1. Imported header file, with PN5180 firmware data, to the project
2. Established SPI interface between µC host and PN5180

4.1.1 SPI Interface

This part of the application implements SPI communication between microcontroller host and PN5180.

The PN5180 provides two different way of SPI handling: one in the normal operation mode, the other one for the secure firmware download. The PN5180 enters the secure firmware download mode with the AUX2/DWL pin at high during the startup.

In the secure firmware download mode the PN5180 uses a different way of BUSY handling. For details refer to PN5180 datasheet [2].

To provide this functionality, it is mandatory that "*phPlatform*" and relevant OSAL and BAL components of the NFC Reader Library [6], are ported to the target platform. To confirm that the NFC Reader library was successfully ported to the target platform, all reference examples, should run without any issue. Porting guidelines of the NXP Reader Library is not part of this documentation.

4.1.2 Command Frame Structure

All messages transmitted between host and the PN5180 must be in "Direction Byte - *Header - Frame – End*" format, see figure below. Detailed description of the data frame format and Secure Firmware Update can be found in the PN5180 Datasheet [2], chapter "Secure Firmware Update".

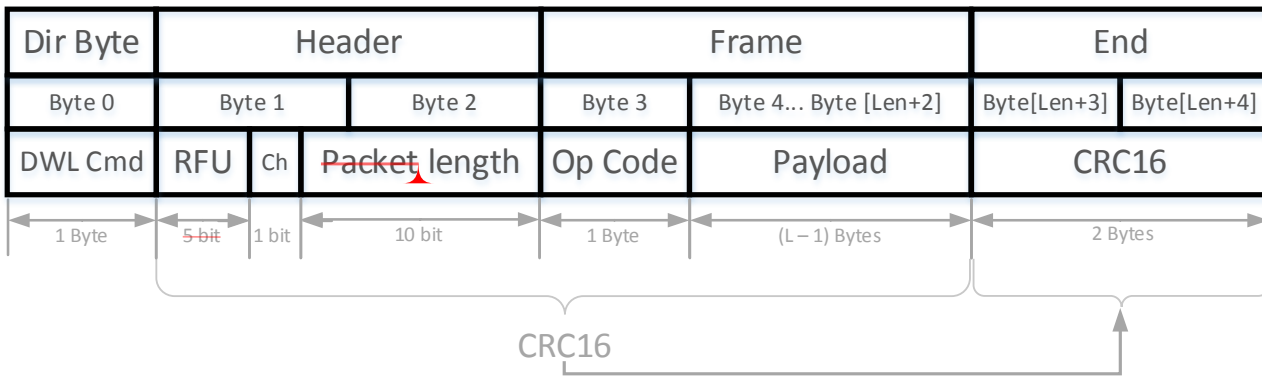


Fig 9. Command Frame Structure

Direction Byte (1 Byte)

- Download Command – 0x7F

Header (2 Bytes)

- RFU (bit 11..15)
- Chunk flag used for fragmentation (bit 10)
    - Chunk bit = 1 for frames 1 to (N – 1)
    - Chunk bit = 0 for the last frame
- Length of the frame (bit 0...9 bit), max length of the frame is 256 bytes.

Frame (Len Bytes)

- Command (1 Byte)
- Payload of the command: (Len -1) Byte

End (2 Bytes)

- CRC16 - The CRC16 is compliant to X.25 (CRC-CCITT, ISO/IEC13239) standard with polynomial x^16 + x^12 + x^5 +1 and preload value 0xFFFF.

4.1.3 Supported Commands in Secure Firmware Download Mode

Following commands are supported in Secure Firmware Download Mode. Detailed description can be found in PN5180 datasheet [2].

Table 1. Secure Firmware Download Commands

Command	Command Code	Description
RESET	0xF0	This command resets the PN5180 IC
GET_VERSION	0xF1	This command provides the IC version and firmware version
GET_SESSION_STATE	0xF2	This command provides the session state and lifecycle.
GET_DIE_ID	0xF4	The command returns the die Identifier
CHECK_INTEGRITY	0xE0	This command checks the integrity of components.
SECURE_WRITE	0xC0	Writes chunks of data to the IC
READ	0xA2	This command reads the EEPROM from specified address.

4.1.4 Command Code Response

A response message is always a multiple of 4 bytes. The first byte of the response is used to indicate the status of the last executed command.

Table 2. Secure Firmware Command Status Return Codes

Command	Command Code	Description
OK	00	command processed properly
ERROR	01 - FF	any response different from 0x00 indicates an error

4.1.5 Application Sequence Flow

Steps required to send a command in Secure Firmware Download mode:

1. Dump and save all USER EEPROM settings
2. Set DWL_REQ pin on PN5180 to high
3. Reset PN5180
4. The PN1580 boots in download mode
5. Send Command
    a. Prepare a data frame
    b. Wait until BUSY PIN goes LOW
    c. Send data frame to PN5180 via SPI
6. Read Response
7. Reset PN5180
8. Restore all USER EEPROM settings

In the figure below the high level application flow diagram is presented.

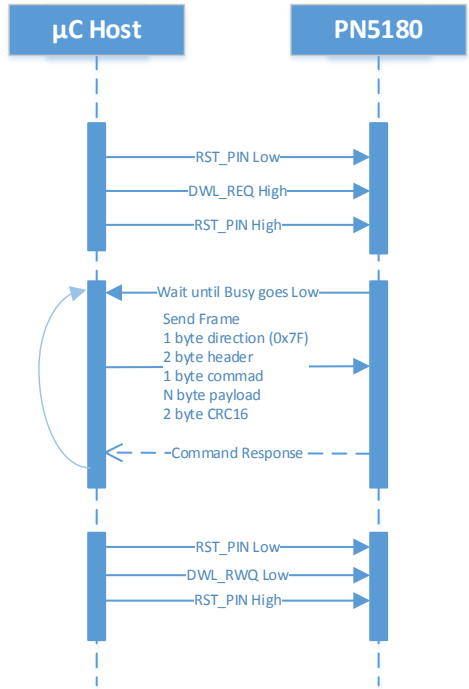


Fig 10. Application Flow Diagram

Note:
As Secure FW Update overwrites user EEPROM settings, it is recommended to dump and save EEPROM before firmware update.

4.1.6 Format of the Prepared Firmware Data

NXP provides raw PN5180 firmware data and it is available in the Docstore [7]. Provided data is available as an array of bytes and it should be imported to the Secure Firmware Update application.

Byte array, of the provided firmware, is composed of multiple parts. Each part consists of header and chunk data. Header is 2 byte long and it specifies size of the chunk data. Size of the chunk is arbitrary big. The whole chunk defines the frame data of the “Secure Write” command.

Block diagram below shows structure of the provided firmware.

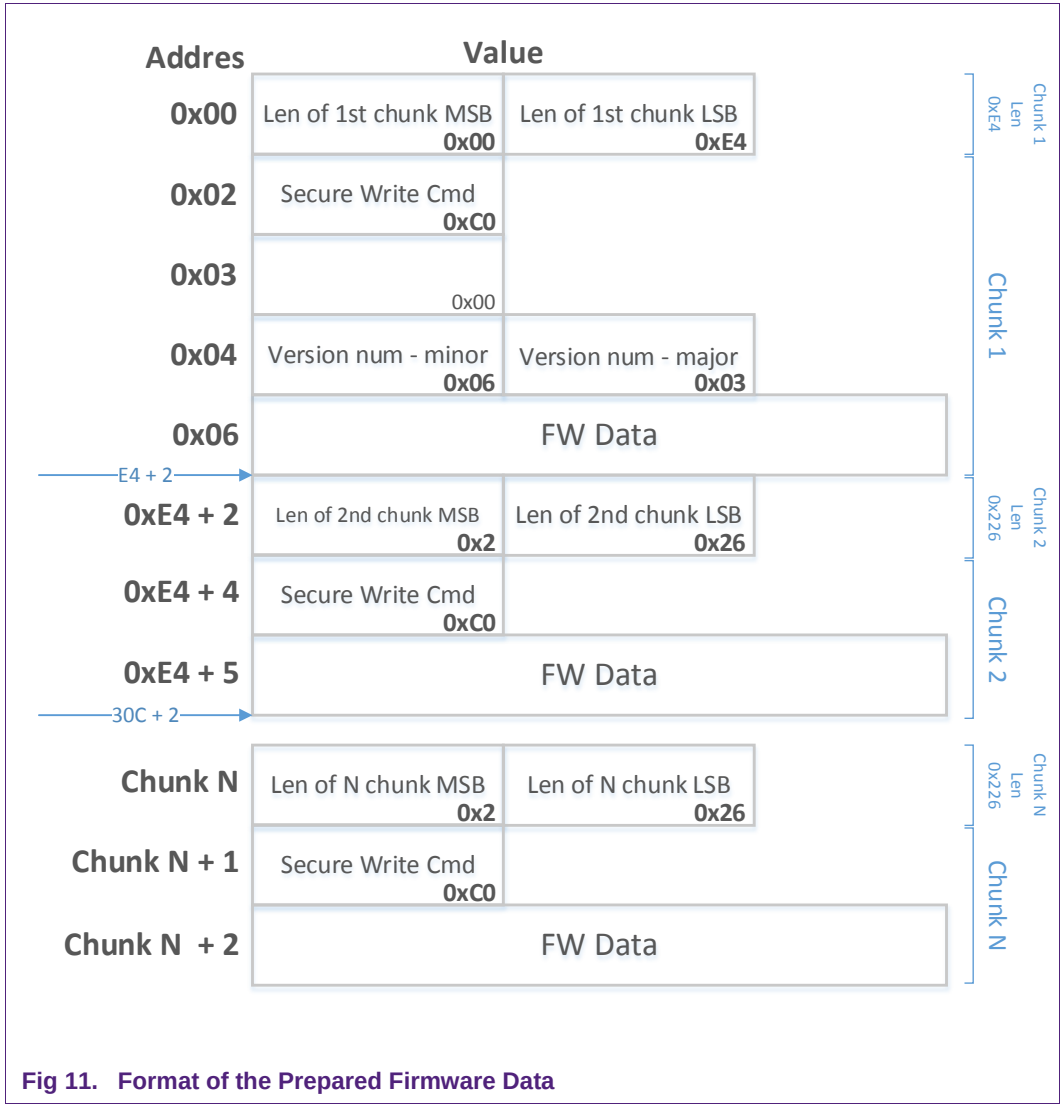


Fig 11. Format of the Prepared Firmware Data

4.1.7 Preparing Chunks from Prepared Firmware Data

Prepared firmware data is provided in zip file, which is available on the Docstore [7]. To be able to use those date, C/C++ header file (*.h, e.g. PN5180Firmware_3_6.h) should be included to the application project.

In this chapter it is described how to build packets, from prepared firmware data, which needs to be sent from the microcontroller host to the PN5180.

In general two different types of chunks (packets) have to be prepared:

1. First group, all Chunks except the last one:
   - Chunk bit is set to 1
2. Last chunk
   - Chunk bit is set to 0

The figure below shows how to prepare chunk packets and which data needs to be copied from provided firmware to the chunk structure.

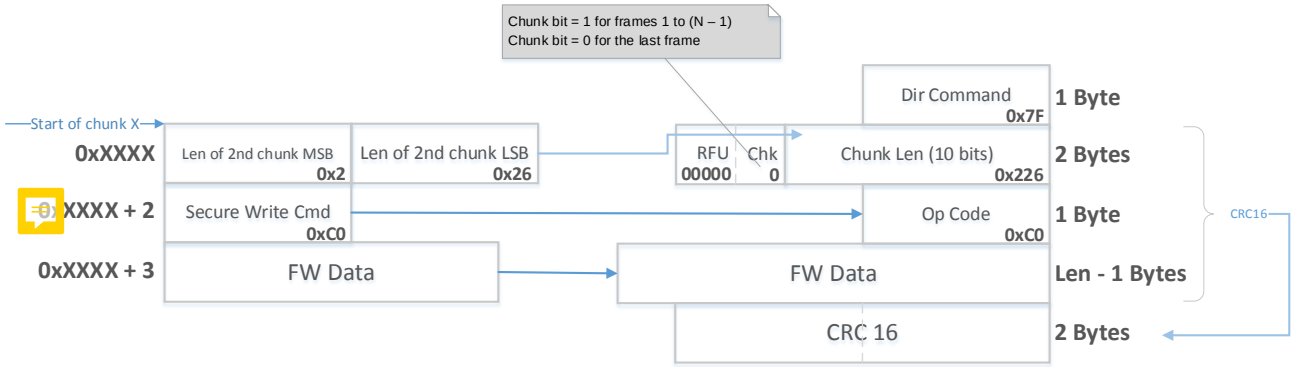


Fig 12. How to Build Chunk Structure

Required steps to prepare chunk frames are described in chapters below.

4.1.7.1 Direction command

First byte of the download command must be a “Direction” Byte.

Direction Byte is 0x7F.

4.1.7.2 Build Chunk Header

Chunk header consists of two bytes, details in PN5180 data sheet [2].

Header (2 bytes)

- RFU (bit 11..15)
- Chunk flag used for fragmentation (bit 10)
    - Chunk bit = 1 in case of fragmentation, for frames which are not last
    - Chunk bit = 0 for the last frame
- Length of the frame (bit 0...9 bit), max length of the frame is 256 bytes.

When frame length is greater than 256 Bytes, frame needs to be fragmented to smaller chunk packets. In that case “Chunk bit” is used to define the last frame.

Ten bits of the chunk header defines the length of the chunk frame. Chunk frame length is defined by first two bytes of the provided data or in case of the fragmentation the size of the frame.

The length defines size of the “Operational Command Code” (1 byte) and “Payload” data ((len-1) bytes) size. In case of fragmentation the length is size of the “Payload”.

4.1.7.3 Build Chunk Frame

Chunk frame consists of:

- “Operational Command Code” (1 byte)
- “Payload” (n bytes).

Operational Command Code is one byte long and follows length bytes as a third byte in the provided structure.

Next bytes defines Payload data. The entire data, following the Operational Command Code, must be copied to the Payload. The length, of the data to be copied, is defined in the chunk header minus one byte of the Operational Command code.

In case of fragmentation, only the first chunk contains Operational Code, following frames contains remaining data.

4.1.7.4 Build CRC16

Last two bytes of the chunk packet are reserved for CRC16. CRC calculation includes chunk header, Operational Command Code and Payload. Two bytes of CRC16 are not included in length defined in chunk header.

4.1.7.5 Example of Chunks

The figure below shows how to build first chunk from provided firmware data.

First two bytes defines the length of the firmware data of the first chunk. The difference between the first chunk and others is that the “Write Command” (0xC0) is followed by three bytes which defines the firmware version. All three bytes are considered as payload data and they are mandatory.

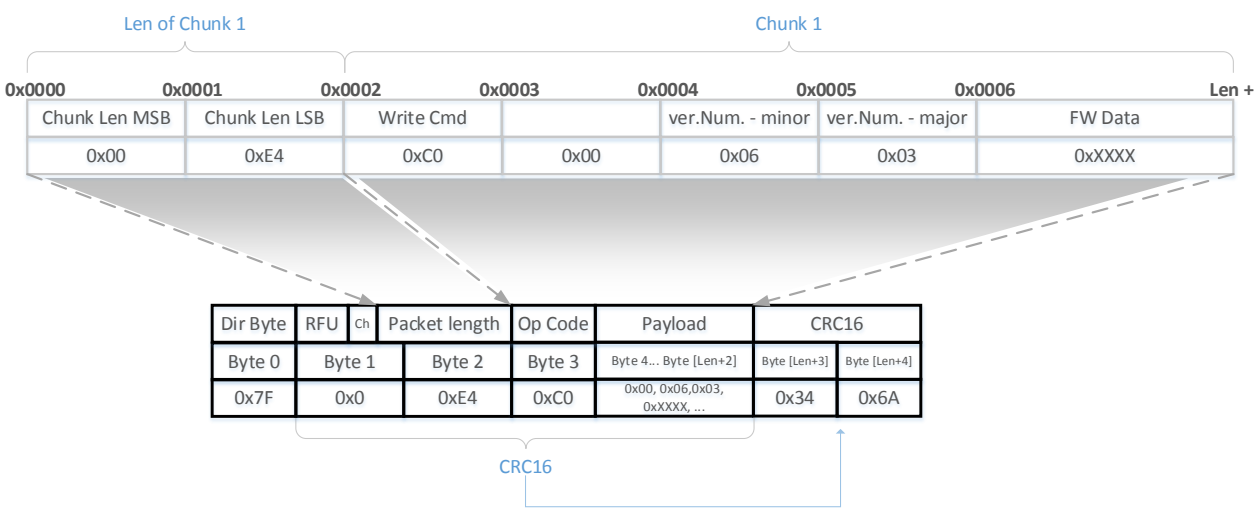


Fig 13. First Chunk

Next figure shows how to build chunks after the first one.

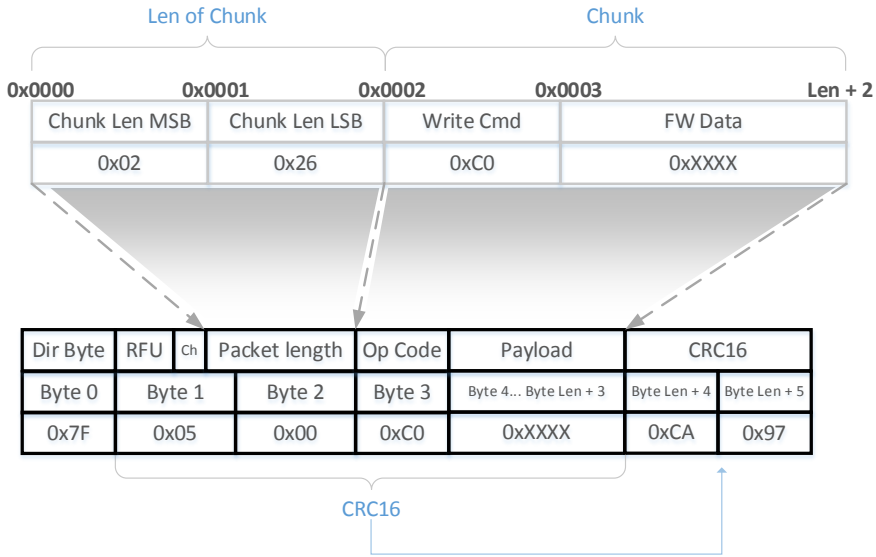


Fig 14. Chunk Structure

In case when frame Payload size is larger than 256 Bytes, fragmentation is required. Figure below shows how to split data to smallest chunks.

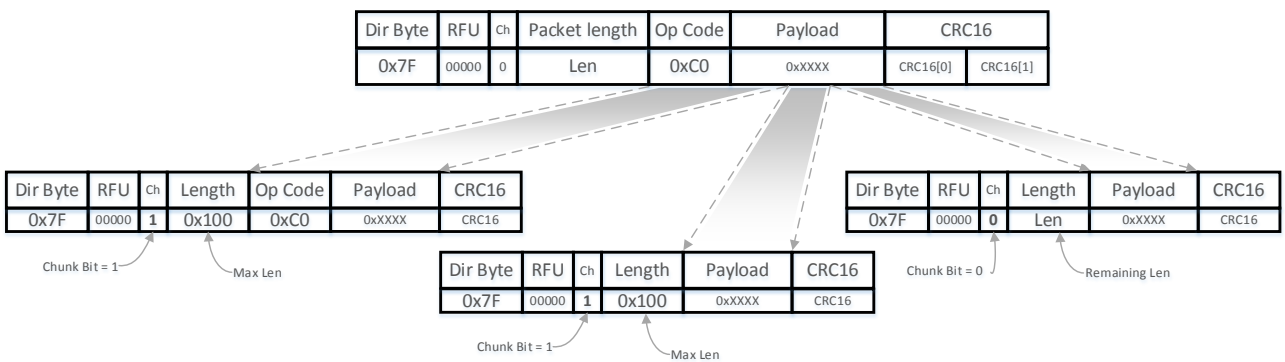


Fig 15. Chunk Fragmentation

4.1.8 Firmware Flash Activity Flow

Diagram below shows activities required to flash new version of the firmware.

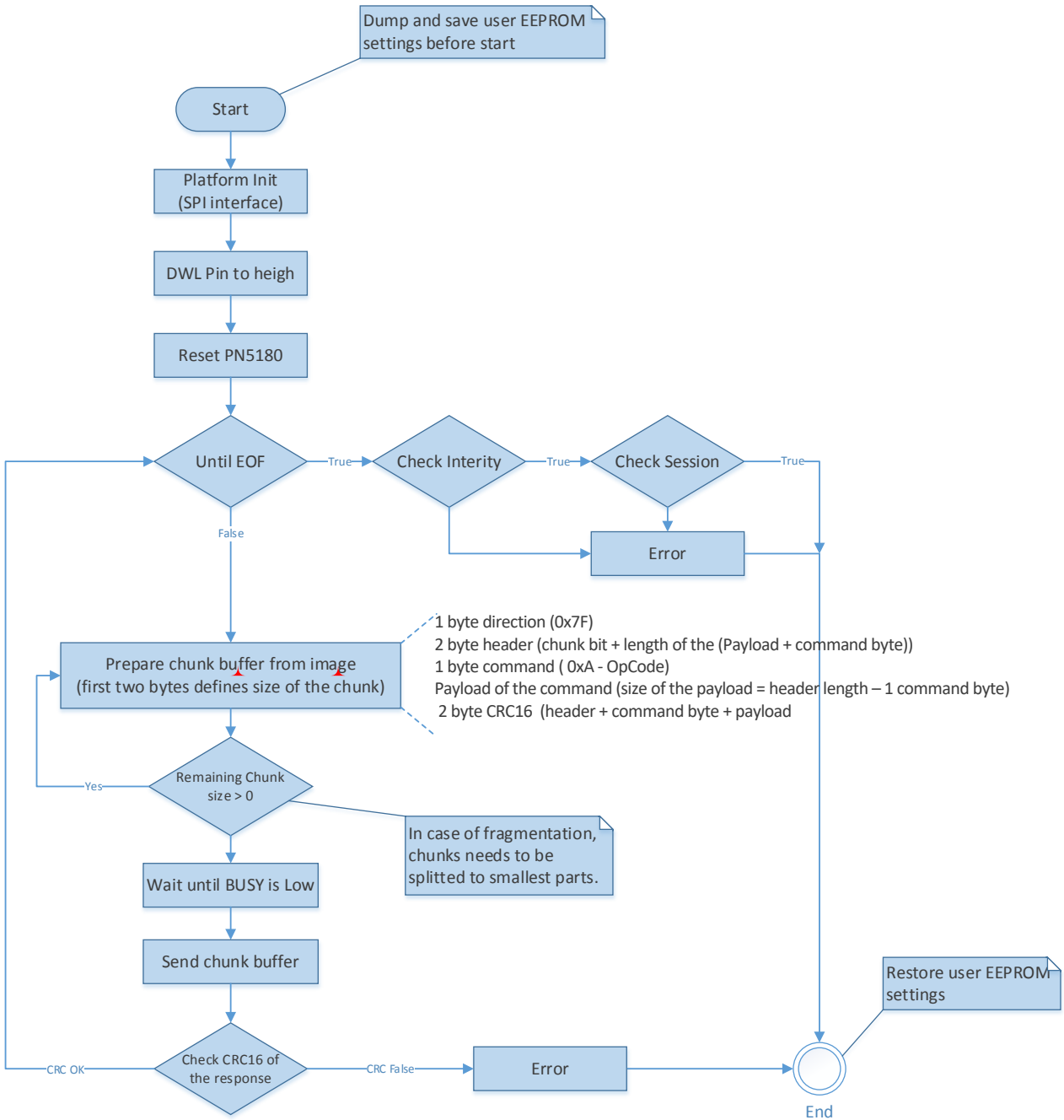


Fig 16. Activity Flow of the Firmware Update Application

Note:
As Secure FW Update overwrites user EEPROM settings, it is required to dump and save EEPROM before firmware update.

4.2 Reference Application

NXP provides “DownloadLibEx1” application as a reference example which demonstrates how to flash new firmware by application hosted on the target microcontroller. This example provides implementation of all commands in Secure Firmware Download mode and it can be downloaded from the PN5180 product page [5].

Reference application package is prepared for the LPC1769 platform together with the PN5180B development board and it contains project for the LPCXpresso IDE [4].

Reference example use Secure Download library, which provides implementation of the secure download APIs. It is recommended to use it in the customer application.

4.2.1 Preconditions

For running this application it is required to setup the system comprising of LPC1769, PN5180 Development board and LPC-Link2, see figure below. Detailed description how to setup development environment is described in UM10954 - PN5180 SW Quick start guide.

To use PN5180 prepared software package all components listed in the table below are required:

Table 3. Development Environment

Item	Version	Description
PNEV5180B	1.0 or higher	PN5180 Customer Evaluation board (hardware)
LPC-Link 2	1.0	Standalone debug adaptor (hardware)
LPCXpresso IDE	8.2.4 or higher	Development IDE (PC software)

Next figure show how to connect PNEV5180B Development board with LPC-Link2 and PC, in this configuration development board is powered by USB.

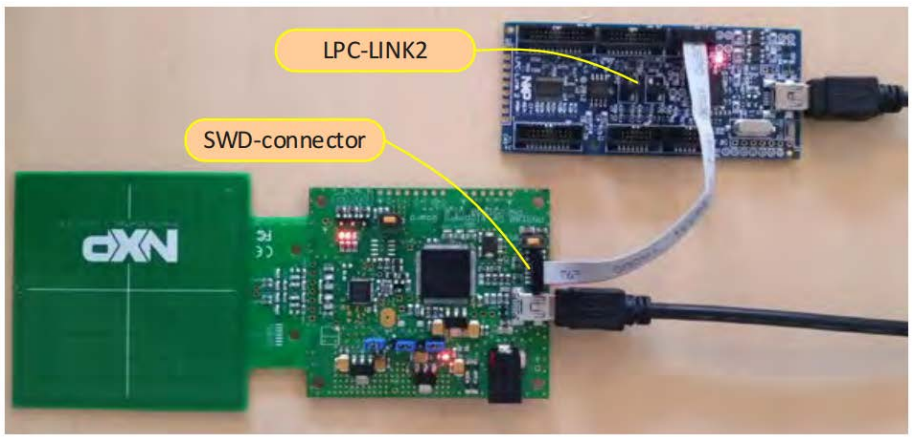


Fig 17. System Setup

Before continuing, it is necessary to download the latest application package ("PN5180_SecureFwUpdateLibrary_v01.00.zip") and extract it to an empty folder.

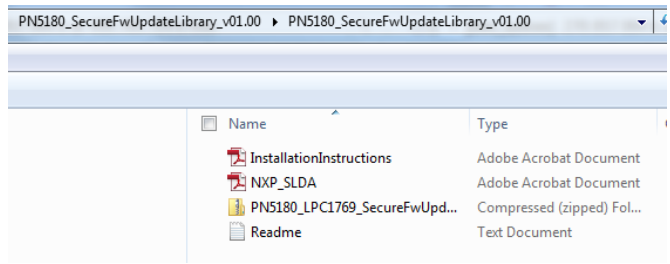
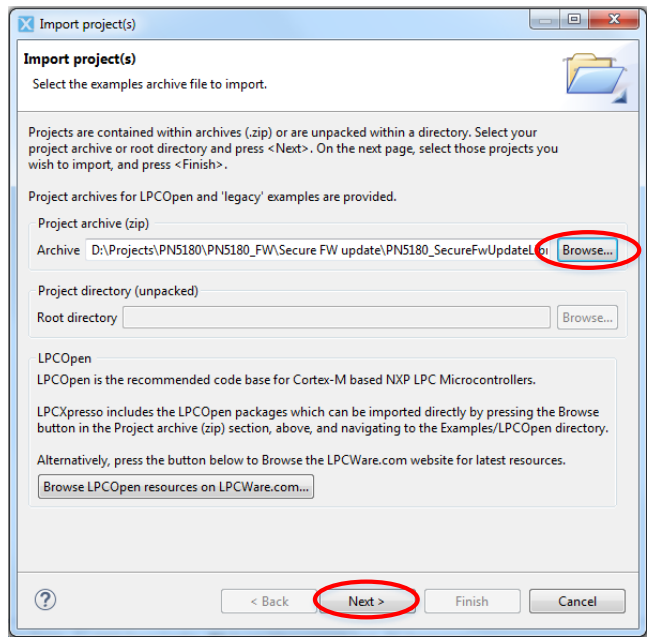


Fig 18. Reference Application Package Content

4.2.2 Import Reference Project

To import project follow steps below:

1. Open LPCXpresso IDE in new workspace
2. Import "PN5180_LPC1769_SecureFwUpdateLibrary_v01.00.00.zip"



(1) In Quickstart window, select "Import project(s)"
(2) Browse to the "PN5180_LPC1769_SecureFwUpdateLibrary_v01.00.00.zip"

Fig 19. Figure title here

3. In next step select all projects and finish import wizard

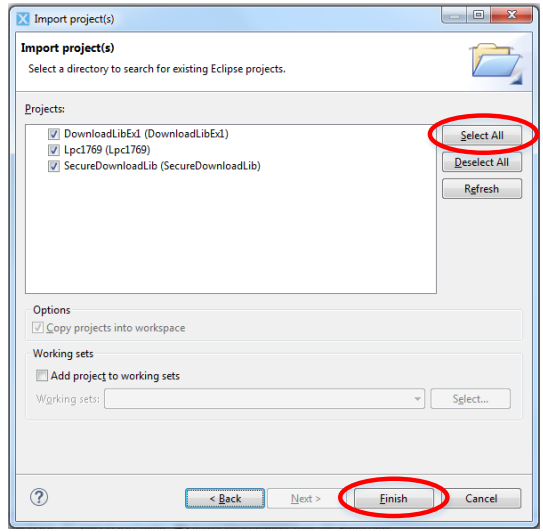


Fig 20. Select Projects

4.2.3 Build, Run and Debug Project

After successful project import, as seen on the figure below, build can be performed.

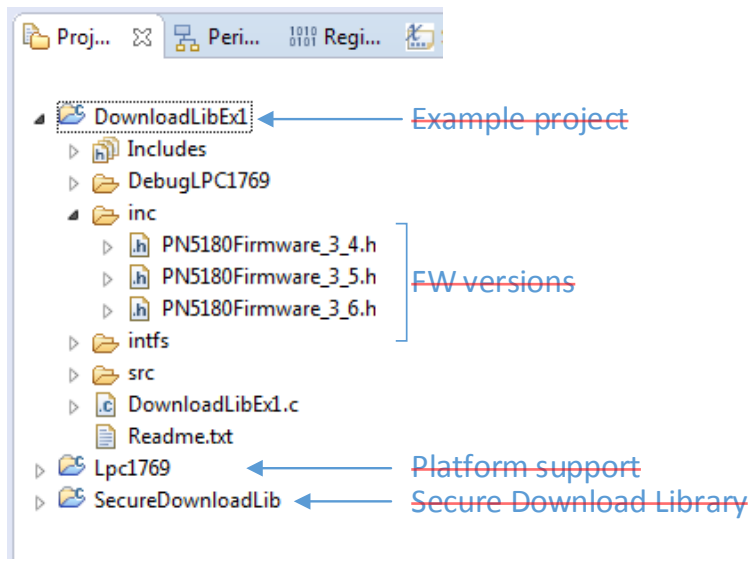


Fig 21. Project Explorer Window

To build or debug application, highlight the “DownloadLibEx1” project in the Project Explorer window and click on “Debug” in the Quick start Panel, as shown in Fig 13. The LPCXpresso IDE builds application, flash application binary and then it starts with debugging.

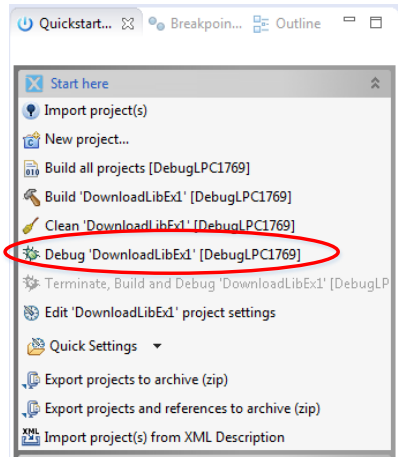


Fig 22. Launch Debug Session

After that application starts and provides options to select tasks.

```
Download Library Example:
 Initialization SUCCESS
 Select the Option
 - Enter 0 to EXIT.
 - Enter 1 for FW Version.
 - Enter 2 to Get DieID.
 - Enter 3 to perform SOFT RESET.
 - Enter 4 to CheckSessionState.
 - Enter 5 to CheckIntegrity.
 - Enter 6 to READ.
 - Enter 7 for Firmware Update 3.4.
 - Enter 8 for Firmware Update 3.5.
 - Enter 9 for Firmware Update 3.6.

Select Option   1
GetFirmwareVersion func
FW ver: 3.4

Select Option   2
GetDieId func
DieID: 00 00 00 00 00 01 7C 0A 3F 0D 39 E2 4E B6 22 46

Select Option   3
PerformSoftReset func
Select Option   9
Firmware upload func
Successful firmware upload

Select Option   1
GetFirmwareVersion func
FW ver: 3.6
```

Fig 23. Application Debug Printouts

4.2.4 Secure Download Library

“Secure Download Library” is part of the reference application and provides implementation of all commands available in secure download mode. Secure download library is written in C programing language and it can be used in any custom application.

Secure Download Library contains implementation of the platform (SPI interface) and this part of the library should be adopted in case it is used on any other platform.

It is highly recommended to use Secure FW Update library in the customer application.

Table lists all APIs supported by the library:

Table 4. API provided by Secure Download Library

API	Description
~~phDlhalHw~~_Pn5180_Download_Init	Initialize download library.
~~phDlhalHw~~_Pn5180_Download_CheckIntegrity	Returns the integrity information of the existing firmware.
~~phDlhalHw~~_Pn5180_Download_CheckSessionState	Check and return current download session state.
~~phDlhalHw~~_Pn5180_Download_GetFirmwareVersion	Read the current PN5180 IC firmware.
~~phDlhalHw~~_Pn5180_Download_GetDieId	This function will return the Die-ID read out from the IC.
~~phDlhalHw~~_Pn5180_Download_PerformSoftReset	Issues a soft reset command.
~~phDlhalHw~~_Pn5180_Download_Read	This function can be used to read the User Area from EEPROM.
~~phDlhalHw~~_Pn5180_Download_PerformSecureFirmware Update	Perform a Secure Firmware Update and checks the integrity of updated firmware.

5. References

[1] OM25180FDK – PN5180 development kit

[2] PN5180 datasheet, www.nxp.com

[3] AN11744 PNEV5180B Quick Start Guide

[4] LPCXpresso IDE - http://www.nxp.com/products/software-and-tools/software-development-tools/software-tools/lpc-microcontroller-utilities/lpcxpresso-ide-v8.2.2:LPCXPRESSO

[5] Product page of the PN5180 http://www.nxp.com/products/identification-and-security/nfc-and-reader-ics/nfc-frontend-solutions/high-performance-multi-protocol-full-nfc-forum-compliant-frontend:PN5180

[6] NFC Reader Library - Software support for NFC Frontend solutions http://www.nxp.com/products/identification-and-security/nfc-and-reader-ics/nfc-controller-solutions/nfc-reader-library-software-support-for-nfc-frontend-solutions:NFC-READER-LIBRARY

[7] PN5180 docstore folder https://www.docstore.nxp.com/flex/DocStoreApp.html#/p//Reader-ICs/Infrastructure-NFC-Ctless-SAM/PN5180