

# AN13936

## PN7642 frequently asked questions

Rev. 3.0 — 26 March 2025

Application note

### Document information

Information	Content
Keywords	FAQ, PN76, PN7642, secure key mode (SKM), MbedTLS, Crypto, Secure Key Store, frequently asked questions
Abstract	This document covers frequently asked questions in a question-answer style.



## 1 Introduction

---

This document is a collection of frequently asked question about PN7642 environment.

Most of the questions are covered in greater detail in other documents. It is highly recommended to get yourself familiar with the PN7642 and related documentation.

## 2 Firmware

### 2.1 Can the firmware be downgraded to another version?

Downgrading is only possible with minor versions. You cannot downgrade to another major version.

Minor downgrade: v02.07 → v02.01 = **possible**

Major downgrade: v02.xx → v01.xx = **not possible**

### 2.2 What is the maximum size of an NXP firmware update file (.esfwu)?

Maximum size of secure firmware update file (includes secure flash code, RF settings) ==> (0xD240) ==> **53,824 bytes**

Maximum size of secure firmware update file including ROM area settings and log area ==> (0xD8C0) ==> **55,488 bytes**

Maximum size of secure firmware update file including ROM area settings, and log area, and NXP configuration area ==> 0xDD10 ==> **56,592 bytes**

### 2.3 If an invalid command with a valid CRC is received in download mode. Does PN7642 stay in download mode?

Yes. If a valid packet format, regardless of the content, is received, the PN7642 stops the HIF-Timeout timer (per default 500 ms) and stays in download mode. Only a valid exit command or VEN reset puts PN7642 out of the download mode.

### 2.4 Do the TPT keys change on a firmware update?

PN7642 C100 device comes by default with firmware v01.00. If you update a PN7642 C100 device to a newer firmware version like v02.02, the TPT keys remain the same.

The TPT\_KEY is bound to the device version and independent of the firmware version. Updating the firmware does not change any key within the secure key store.

### 2.5 What is the difference between "xx.0x" and "xx.Fx" firmware version?

NXP always publishes two different firmware files for one version. The firmware file with "F" in its name is not updating the user settings area. The firmware update file with "0" instead of "F" always updates all the user settings as well.

PN7642Firmware\_02.00.esfwu → Is updating user configuration. Can be used to revert to factory settings.

PN7642Firmware\_02.F0.esfwu → Is not updating user configuration. Can be used after you have configured your PN7642 to your hardware.

## 2.6 What is the difference between PN7642 C100 and C101?

The PN7642 is available in two versions: C100 and C101. The major difference between the two versions is the default firmware version and Pin-less download functionality.

- **C100**
  - Firmware version: v01.00
  - *Pin-less download*: **disabled** by default
- **C101**
  - Firmware version: v02.00
  - *Pin-less download*: **enabled** by default

The *Pin-less download* feature sets whether PN7642 probes DWL\_REQ pin at boot-up to go to bootloader mode or not.

- If *Pin-less download* is disabled, PN7642 goes to bootloader mode only if DWL\_REQ is high while booting.
- If *Pin-less download* is enabled, at power-on reset (POR), PN7642 goes to bootloader mode. The state of the DWL\_REQ pin does not matter and is not considered. The PN7642 remains in bootloader mode for some time (500 ms by default, user configurable) to listen for bootloader commands.

If PN7642 version C100 is updated with the same firmware version as PN7642 version C101, both versions have the same functionality. The only difference is the default configuration of the *Pin-less download* feature.

For firmware version v02.00 and above, the *Pin-less download* feature can be enabled on PN7642 version C100 with the API `PN76_Sys_OTPConfigs_EnableDwnldReqLessBoot()`.

**Note:** *OTP stands for One Time Programmable and is not reversible!*

**CAUTION:** *If Pin-less download is enabled— either via the API on PN7642 version C100, or by default on PN7642 version C101— USB Mass-storage mode (USB download) is not available.*

## 2.7 How can I set the timeout in Pin-less download mode?

If Pin-less download mode is enabled (default on C101, configurable on C100) the default value to wait in bootloader is 500 ms. This can be configured by the user with the following API:

### ◆ PN76\_Sys\_OTPConfigs\_DwnldReqLessBoot()

**PN76\_Status\_t** PN76\_Sys\_OTPConfigs\_DwnldReqLessBoot ( **uint32\_t** dwTimeOut )

This instruction is used to set timeout for download request pin less boot.

#### Parameters

**dwTimeOut** Maximum wait timeout before switching to Normal Operation Mode in flash (One unit is 5.66 Micro sec).

#### Return values

<b>PN76_STATUS_SUCCESS</b>	Operation Successful.
<b>PN76_STATUS_RESOURCE_ERROR</b>	download request pin less feature NOT available.
<b>PN76_STATUS_PARAMETER_ERROR</b>	In case the PinLess feature is NOT enabled.
<b>PN76_STATUS_AUTH_ERROR</b>	NO access permissions to do the requested settings.
<b>PN76_STATUS_MEMORY_ERROR</b>	Flash/EE memory read/write operation failed.

Figure 1. Configure Pin-less download timeout

Being an OTP (One Time Programmable) API call, it shall be used carefully.

Another option to try different timeout values is the API "**PN76\_Sys\_Set\_HIF\_Timeout()**". This API writes the timeout value into EEPROM. Be careful with EEPROM writes, those shall be used under stable conditions and not be interrupted (see [Section 9.2](#)).

## 2.8 How can I set the application firmware version?

At the creation of a *.esfwu* file for the customer application, a version field is present. This version belongs to the user application and can be read out by the bootloader and the application itself.

This version can only be set by the *.esfwu* creation and is updated by the NXP bootloader. It cannot be set from the application space or any other way.

## 2.9 What happens if a firmware update fails?

If the firmware update procedure is interrupted or fails, the PN7642 remains in download mode (bootloader).

In case the firmware update was started via IAP (In-Application Programming), the chip can only be recovered by using the host-interface to update the firmware.

Reading the firmware version is not a reliable source to see if a firmware update was successful. If the signature of the first frame matches (genuine and authentic firmware file) the new firmware version is already written. If the update is interrupted later on, the firmware version remains the new one, even though the update was not successful.

To see if an update was successful, check the session state "DL\_GET\_SESSION\_STATE". For more information, refer to [\[4\]](#).

For information on how to update the firmware, see [\[6\]](#).

## 3 Secure key store and cryptography

### 3.1 What is the SKM?

The acronym SKM stands for "Secure Key Mode". The secure key mode is a special mode of the PN7642, where you open a session to interact with the secure key store. Only if a valid session in the secure key mode is opened, keys can be provisioned, deleted, and purged from the secure key store.

### 3.2 When is the SKM authentication counter increased?

The authentication counter in SKM is increased:

1. If the challenge and response are proper.
2. If the challenge and response are not proper, or if invalid values are used for the key (128-bit or 256-bit key).
3. Any internal errors (failures) are returned by the crypto algorithms for decryption of the challenge, or for access to key store/internal SGI IPs.
4. Any internal error results in secure memory read comparison.

In all these conditions, the authentication counter is increased when APIs are called or when SKM boot mode is entered to work on keys.

### 3.3 Key store is locked, what can I do?

Once the key store is locked, it stays locked. There is no way to unlock the key store. All provisioned keys can be used and purged. But no new authentication to the key store is possible.

**The counter for faulty authentications is not reset by any action. After ten unsuccessful authentications, the key store is locked.**

Take special care with the examples, like the MIFARE DESFire example, that the correct TPT\_KEYS are set.

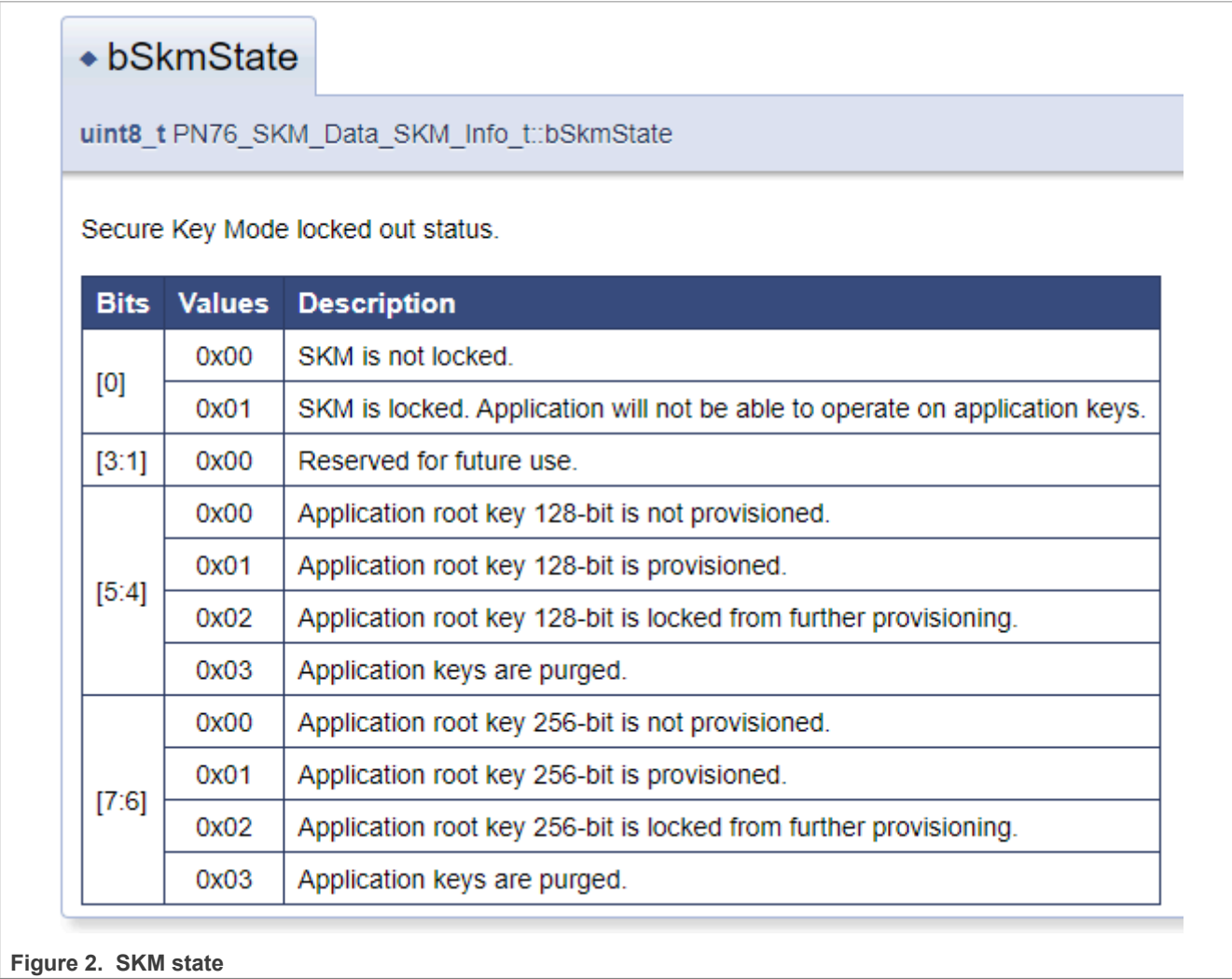
**Note:** In early versions of the SDK, the TPT\_KEYS have not been set correctly. Running the MIFARE DESFire example without changing the TPT\_KEY results in a failed authentication.

3.4 When can I provision a APP\_MASTER\_KEY?

Provisioning of the secure key store is only possible if the secure key store is not locked and you must provision the APP\_ROOT\_KEY first.

- If the key store is locked, no further provisioning is possible. Keys within the key store are still valid and can be used, but no new key can be provisioned.
- If the APP\_ROOT\_KEY is not provisioned, the default TPT\_KEYS (transport keys available in PN7642 data sheet) are still active. Provision both 128-bit and 256-bit APP\_ROOT\_KEYs before you provision any other key.

To obtain the SKM state, use the API "PN76\_Sys\_SKM\_Get\_SKM\_Info()" which returns the SKM state (Figure 2).



For how to work with the secure key mode (SKM) to provision the key store, use *pnev7642fama\_pn\_skm* example available in the SDK. And read the application note [2].



### 3.5 Why does the chip reset by calling EdDSA\_Sign?

Before using any cryptographic operations and APIs on the PN7642, the module must be initialized. In addition, if keys or key derivation operations are used, the KeyStore must be initialized as well.

See example *pn7642fama\_pn\_mbedtls\_demo* for a detailed how-to.

```
213
214  /* Initialize the crypto modules */
215  InitStatus = (PN76_Status_t)phmbedcrypto_Init();
216  if (InitStatus != PN76_STATUS_SUCCESS)
217  {
218      PRINTF("Crypto initialization failure\r\n");
219      while (1)
220          ;
221  }
222
223  /* Initialize the PN7642 KeyStore module */
224  eKeyStoreStatus = PN76_Sys_KeyStore_Init(&bKeyStoreStatus);
225
226  /* bKeyStoreStatus 6th bit means fatal error. */
227  if ((eKeyStoreStatus != PN76_STATUS_SUCCESS) || ((bKeyStoreStatus & 0x40U) != 0U))
228  {
229      PRINTF("Crypto initialization error\r\n");
230      while (1)
231          ; /* if Failed Do not go further */
232  }
233
```

Figure 3. Initialization of Crypto Unit and KeyStore

4 Host interface

4.1 What can I do if PN7642 does not answer on HIF I2C?

In the bootloader, PN7642 uses the values of ATX\_C (I<sup>2</sup>C Adr Bit 0) and ATX\_D (I<sup>2</sup>C Adr Bit 1) to set the I<sup>2</sup>C address.

For more information about I<sup>2</sup>C addressing and the I<sup>2</sup>C host interface functionality, refer to [\[1\]](#).

Table 1. I<sup>2</sup>C interface addressing

I <sup>2</sup> C_ADR1	I <sup>2</sup> C_ADR0	I <sup>2</sup> C address (R/W = 0. write)	I <sup>2</sup> C address (R/W = 0, read)
0	0	0x28	0x28
0	1	0x29	0x29
1	0	0x2A	0x2A
1	1	0x2B	0x2B

## 5 Contact Interface

### 5.1 Why do I get a ATR parsing error?

The CT library can be initialized to be compliant with either ISO7816 or EMVCo. Especially at the ATR there are differences, which are incompatible with each other. Make sure to set the compliance to your required specification:

```
120
121 static void phExCcid_CtTask_Init(phhalCt_SlotType_t slotType)
122 {
123     g_CtTaskData.halCt.phhalCt_Params[0].pTDAPins = &g_CtTaskData.tda[0];
124     g_CtTaskData.halCt.phhalCt_Params[1].pTDAPins = &g_CtTaskData.tda[1];
125     g_CtTaskData.palCt.phalDataParams = &g_CtTaskData.halCt;
126     phpalCt_Init(&g_CtTaskData.palCt);
127     g_CtTaskData.palCt.sAtrParams.pbAtrBuffer = gbphCT7816_ApduRespBuffer;
128
129     phhalCt_Init(&g_CtTaskData.halCt, slotType);
130     phpalCt_SetConfig(&g_CtTaskData.palCt, E_CONF_COMPLIANCE, (uint8_t)E_ISO7816_ENABLE);
131 }
132
```

Figure 4. CT compliance configuration

## 6 Software

### 6.1 Difference between *libintfs.a* and *intfs.a* in the SDK?

Both *libintfs.a* and *intfs.a* libraries are available in the SDK.

The two libraries serve the same purpose, have the same content, but comply with different toolchain requirements. If you must use one library, pick any of the two available in the SDK.

### 6.2 NFC Cockpit is not working. What can I do?

Refer to the section *Firmware overview* in [\[3\]](#).

The NFC Cockpit application (*.bin*) is compiled for a particular firmware version of PN7642. Major firmware versions are not compatible because of API address changes. For example, the NFC Cockpit application compiled for PN7642 FW v1.0 does not run on PN7642 with firmware v2.0.

### 6.3 Why is the SDK not working with VSC?

Even though the MCUXpresso Visual Studio Code (VSC) extension is available, not all SDK versions are compatible. For example, PN7642 SDK version vxx.\*12\*.xx is not compatible. To be supported by the VSC extension, the SDK minimum version must be 14 (xx.14.xx).

Updates are planned to make the PN7642 SDK VSC compliant.

### 6.4 Why can't I see any output in the IDE console?

When importing an example from the SDK, you can choose the debug console. If the *SDK Debug Console* is 'Semihost', the debug output is in the IDE console. If *SDK Debug Console* is 'UART', the debug output can be grabbed on the debug UART.

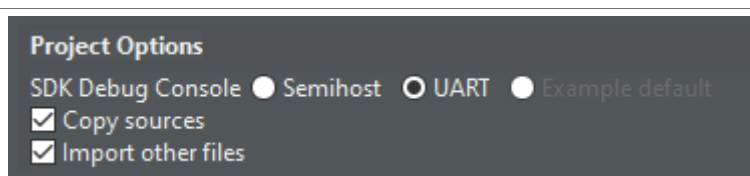


Figure 5. Example of project configuration for debug console

If you do not see any output in your MCUXpresso IDE console, cross-check the settings. You can change the settings for an example you have already imported.

- Go to the *Quickstart Panel*.
- Select *Quick Settings*.
- Select *SDK Debug Console*.
- Select *Semihost console* or *UART console*.

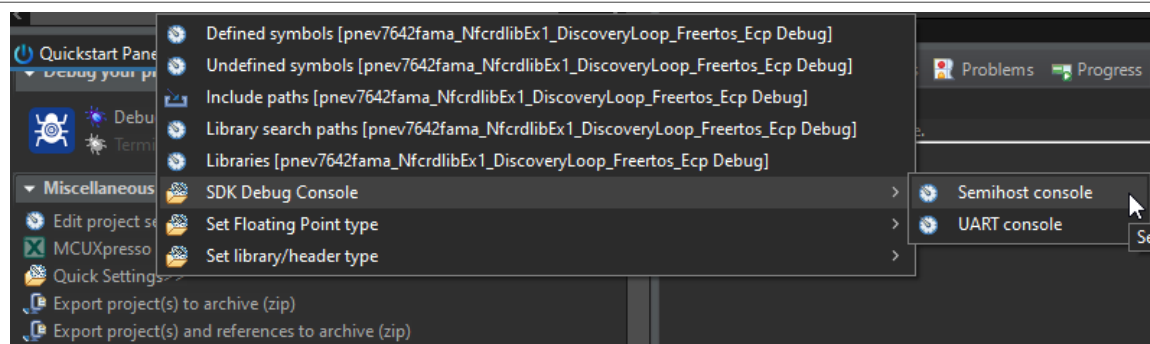


Figure 6. Quick settings for debug console

## 6.5 My examples are crashing. What to do?

Most examples rely on the underlying libraries. The libraries are built for a specific PN7642 firmware version. If the firmware version of PN7642 does not match the firmware version used by the SDK, all kinds of issues and unexpected behavior can occur.

For details on compatibility, refer to the section *Firmware overview* in [3]. And either update the firmware of your PN7642 to the SDK firmware, or use the correct SDK version for your PN7642.

## 6.6 Why is LPCD not going to standby?

On NXP Open-Controllers (PN7462 and PN7642), LPCD does not automatically set the chip to standby. By calling the LPCD API immediately after the measurement RF-Ping, the result is returned.

On NXP Closed-Controllers (PN5190, PN7160, PN7220, etc.), LPCD automatically sets the chip to standby.

This differing behavior was chosen on purpose to give the freedom of choice to the developer. As the PN7642 is an SoC, it can control other entities or fulfill other tasks in the user space.

The user can check the result of LPCD, and go to standby by calling one of the PN7642 standby modes (Standby, Deep Sleep, Ultra Low-Power Standby, etc.).

## 6.7 Why can't I connect with my debugger?

If the PN7642 is in ultra low power mode (ULP Standby or ULPCD), in reset, or in download mode (bootloader), the debugger cannot attach. In case a secure firmware update was unsuccessful, the PN7642 will remain in download mode until the firmware has been updated successfully.

The only way to check if the PN7642 is currently in download mode is to connect via the HIF and send a HDLL command (see [UM11905](#)).

## 7 Documentation

### 7.1 Where do I find the API documentation?

The *PN7642 NFC Controller User API Documentation* is part of the SDK package. If you and open

- Unzip the SDK package.
- Open the *docs* directory.
- Unzip *PN76-FW-apiguide.zip*.
- Open the *index.html* file.

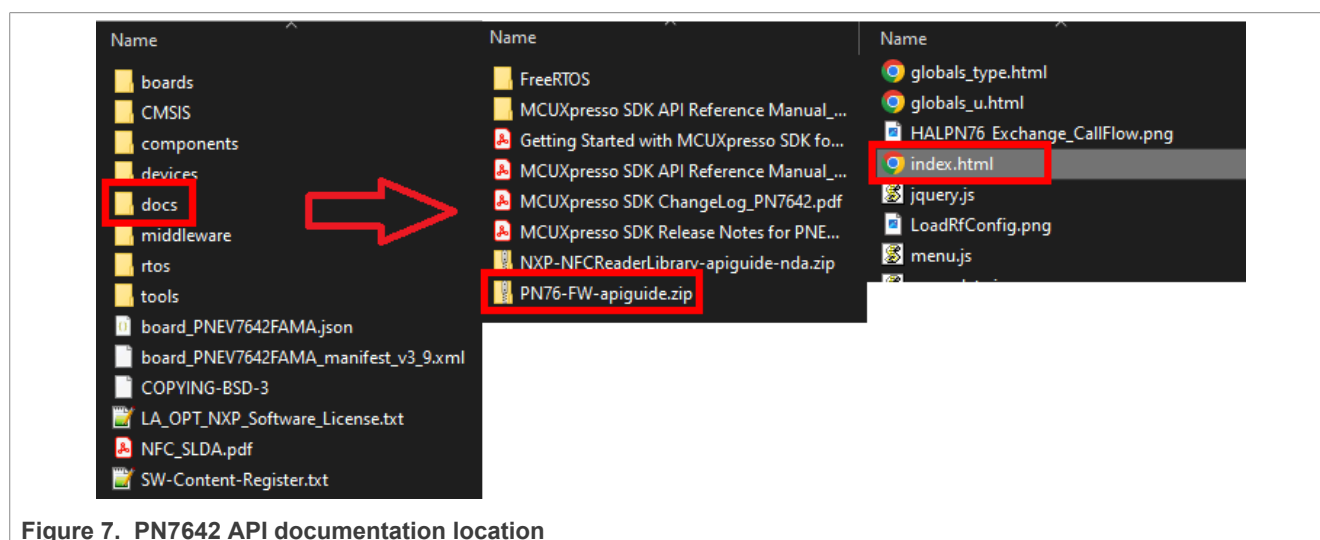


Figure 7. PN7642 API documentation location

The API documentation is generated with DoxyGen out of the actual SDK source code.

### 7.2 Why is the PN5190 mentioned in PN7642 documentation?

The PN5190 and PN7642 share the same hardware platform. Many of the hardware blocks are identical. The PN5190 has been on the market a lot longer and more documentation for it has been created. Some documents are taken from the PN5190 and slightly altered for the PN7642. During this process, it can occur that renaming "PN5190" to "PN7642" has been missed.

## 8 Examples

### 8.1 Which keys to choose in NfcrdlibEx10\_MIFAREDESFire?

At SDK version v2.15.0, the user has to define which TPT\_KEYS are used. Three options are present in the example. The TPT\_KEYS are listed in the data sheet [\[1\]](#).

- PN7642EV\_C100: Define this if you use a PN7642 **C100** chip.
- PN7642EV\_C101: Define this if you use a PN7642 **C101** chip.
- PN7642EV\_INT: These are engineering samples and cannot be officially bought. Triple check the markings on your chip as you most probably do not have this chip version.

Where you set the define (a line above, preprocessor defines, others) does not matter as long as the right keys are enabled. Enabling the wrong key set can lead to locking the key store. A locked key store cannot be unlocked! See [Section 3.3](#).

**Note:** All PNEV7642 development boards are populated with a **C100** chip.

### 8.2 No reset in watchdog example?

The watchdog example does not execute a chip reset on purpose. If the example executes a reset, the example runs in an endless loop of resets and a debugger is continuously losing connection. In the NMI\_Handler(void) a flag is set, representing the triggering of the watchdog.

In this handler, the user can add a reset if they want to. There are multiple ways how a reset can be executed:

- CMSIS **\_\_NVIC\_SystemReset(void)**
- PN7642 API **PN76\_Sys\_Hal\_SoftReset(uint8\_t)**
- PN7642 API **PN76\_Sys\_Hal\_HardReset(void)**

## 9 Miscellaneous

---

### 9.1 What is the error code PCRM\_SYS\_BOOT1\_STS\_RESERVED\_MASK?

This error code is "PCRM\_SYS\_BOOT1\_STS\_REG\_BOOT\_ULPCD\_GPADC\_READY\_TIMEOUT" and has been fixed in firmware version v02.05. See the release notes [RN00257](#) for firmware v02.05 or later.

If you encounter this error please update your firmware to higher than v02.05 and consult [AN14518](#) for further details on the correct crystal setup.

### 9.2 Why is my EEPROM corrupted?

If EEPROM writes are executed and interrupted (power loss, reset, undervoltage, etc...) the page which is currently written might be corrupted. Make sure that EEPROM writes are executed in a power stable and controlled environment.

Up to SDK v02.15.002 EEPROM writes are in some of the NFC examples. This will be fixed, and from v02.15.003 onwards no EEPROM writes are executed within the PN7642 SDK examples.

With the API "PN76\_EDL\_Get\_Integrity\_NXP\_Fw(...)" (see API reference manual) or the HDLL command "DL\_CHECK\_INTEGRITY" (see [UM11905](#)) you can get information about the integrity status of the NXP FW and EEPROM.



## 10 Abbreviations

Table 2. Abbreviations

Acronym	Description
ESFWU	Encrypted secure firmware update
HIF	Host interface
KS	Key store
LPCD	Low Power Card Detection
OTP	One time programmable
POR	Power-on reset
SKM	Secure key mode
SOC	System on Chip
TPT_KEY	Transport key
ULP	Ultra Low Power
ULPCD	Ultra Low Power Card Detection
IAP	In Application Programming
CMSIS	Common Microcontroller Software Interface Standard
API	Application Programming Interface

## 11 References

---

- [1] Data sheet - PN7642 - Single chip solution with high-performance NFC reader, customizable MCU, and security toolbox ([link](#))
- [2] Application note - AN13720 - PN7642 Secure Key Mode demo application ([link](#))
- [3] Application note - AN13134 - PN76 family evaluation board quick start guide ([link](#))
- [4] User manual - UM11905 - PN76 instruction manual ([link](#))
- [5] Release note - RN00257 - PN7642 firmware release note ([link](#))
- [6] Application note - AN14540 - Firmware Update on PN7642 ([link](#))

12 Revision history

Table 3. Revision history

Version	Release date	Description
AN13936 v.3.0	26 March 2025	<ul style="list-style-type: none"><li>• <a href="#">Section 2.6 "What is the difference between PN7642 C100 and C101?"</a> updated.</li><li>• <a href="#">Section 2.7 "How can I set the timeout in Pin-less download mode?"</a> added.</li><li>• <a href="#">Section 2.9 "What happens if a firmware update fails?"</a>: added.</li><li>• <a href="#">Section 6.7 "Why can't I connect with my debugger?"</a> added.</li><li>• <a href="#">Section 8.2 "No reset in watchdog example?"</a>: added.</li><li>• <a href="#">Section 9 "Miscellaneous"</a>: added.</li><li>• <a href="#">Section 9.1 "What is the error code PCRM_SYS_BOOT1_STS_RESERVED_MASK?"</a> added.</li><li>• <a href="#">Section 9.2 "Why is my EEPROM corrupted?"</a>: added.</li></ul>
AN13936 v.2.0	14 October 2024	<p>Editorial changes.</p> <ul style="list-style-type: none"><li>• <a href="#">Section 2.8 "How can I set the application firmware version?"</a>: added.</li><li>• <a href="#">Section 3.5 "Why does the chip reset by calling EdDSA_Sign?"</a>: added.</li><li>• <a href="#">Section 5 "Contact Interface"</a>: added.</li><li>• <a href="#">Section 6.6 "Why is LPCD not going to standby?"</a>: added.</li><li>• <a href="#">Section 7.2 "Why is the PN5190 mentioned in PN7642 documentation?"</a>: added.</li><li>• <a href="#">Section 8 "Examples"</a>: added.</li><li>• <a href="#">Section 10 "Abbreviations"</a>: updated.</li></ul>
AN13936 v.1.0	29 February 2024	<ul style="list-style-type: none"><li>• Initial version</li></ul>

## Legal information

### Definitions

**Draft** — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

### Disclaimers

**Limited warranty and liability** — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

**Right to make changes** — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

**Suitability for use** — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

**Applications** — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

**Terms and conditions of commercial sale** — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <https://www.nxp.com/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

**Export control** — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

**Suitability for use in non-automotive qualified products** — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

**HTML publications** — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

**Translations** — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

**Security** — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at [PSIRT@nxp.com](mailto:PSIRT@nxp.com)) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

**NXP B.V.** — NXP B.V. is not an operating company and it does not distribute or sell products.

## Licenses

**Purchase of NXP ICs with NFC technology** — Purchase of an NXP Semiconductors IC that complies with one of the Near Field Communication (NFC) standards ISO/IEC 18092 and ISO/IEC 21481 does not convey an implied license under any patent right infringed by implementation of any of those standards. Purchase of NXP Semiconductors IC does not include a license to any NXP patent (or other IP right) covering combinations of those products with other products, whether hardware or software.

## Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

**NXP** — wordmark and logo are trademarks of NXP B.V.

**AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile** — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

**Apple** — is a registered trademark of Apple Inc.

**DESFire** — is a trademark of NXP B.V.

**EdgeVerse** — is a trademark of NXP B.V.

**FeliCa** — is a trademark of Sony Corporation.

**ICODE** — is a trademark of NXP B.V.

**I2C-bus** — logo is a trademark of NXP B.V.

**MIFARE** — is a trademark of NXP B.V.

**MIFARE Classic** — is a trademark of NXP B.V.

**MIFARE Plus** — is a trademark of NXP B.V.

**MIFARE Ultralight** — is a trademark of NXP B.V.

Tables

Tab. 1.	I2C interface addressing .....	10	Tab. 3.	Revision history .....	19
Tab. 2.	Abbreviations .....	17			

Figures

Fig. 1.	Configure Pin-less download timeout .....	5	Fig. 5.	Example of project configuration for debug console .....	12
Fig. 2.	SKM state .....	8	Fig. 6.	Quick settings for debug console .....	13
Fig. 3.	Initialization of Crypto Unit and KeyStore .....	9	Fig. 7.	PN7642 API documentation location .....	14
Fig. 4.	CT compliance configuration .....	11			

## Contents

<b>1</b>	<b>Introduction .....</b>	<b>2</b>	9.1	What is the error code PCRM_SYS_	
<b>2</b>	<b>Firmware .....</b>	<b>3</b>		BOOT1_STS_RESERVED_MASK? .....	16
2.1	Can the firmware be downgraded to		9.2	Why is my EEPROM corrupted? .....	16
	another version? .....	3	<b>10</b>	<b>Abbreviations .....</b>	<b>17</b>
2.2	What is the maximum size of an NXP		<b>11</b>	<b>References .....</b>	<b>18</b>
	firmware update file (.esfwu)? .....	3	<b>12</b>	<b>Revision history .....</b>	<b>19</b>
2.3	If an invalid command with a valid CRC is			<b>Legal information .....</b>	<b>20</b>
	received in download mode. Does PN7642				
	stay in download mode? .....	3			
2.4	Do the TPT keys change on a firmware				
	update? .....	3			
2.5	What is the difference between "xx.0x" and				
	"xx.Fx" firmware version? .....	3			
2.6	What is the difference between PN7642				
	C100 and C101? .....	4			
2.7	How can I set the timeout in Pin-less				
	download mode? .....	5			
2.8	How can I set the application firmware				
	version? .....	6			
2.9	What happens if a firmware update fails? .....	6			
<b>3</b>	<b>Secure key store and cryptography .....</b>	<b>7</b>			
3.1	What is the SKM? .....	7			
3.2	When is the SKM authentication counter				
	increased? .....	7			
3.3	Key store is locked, what can I do? .....	7			
3.4	When can I provision a APP_MASTER_				
	KEY? .....	8			
3.5	Why does the chip reset by calling EdDSA_				
	Sign? .....	9			
<b>4</b>	<b>Host interface .....</b>	<b>10</b>			
4.1	What can I do if PN7642 does not answer				
	on HIF I2C? .....	10			
<b>5</b>	<b>Contact Interface .....</b>	<b>11</b>			
5.1	Why do I get a ATR parsing error? .....	11			
<b>6</b>	<b>Software .....</b>	<b>12</b>			
6.1	Difference between libintfs.a and intfs.a in				
	the SDK? .....	12			
6.2	NFC Cockpit is not working. What can I do? ....	12			
6.3	Why is the SDK not working with VSC? .....	12			
6.4	Why can't I see any output in the IDE				
	console? .....	12			
6.5	My examples are crashing. What to do? .....	13			
6.6	Why is LPCD not going to standby? .....	13			
6.7	Why can't I connect with my debugger? .....	13			
<b>7</b>	<b>Documentation .....</b>	<b>14</b>			
7.1	Where do I find the API documentation? .....	14			
7.2	Why is the PN5190 mentioned in PN7642				
	documentation? .....	14			
<b>8</b>	<b>Examples .....</b>	<b>15</b>			
8.1	Which keys to choose in NfcrcLibEx10_				
	MIFAREDESFire? .....	15			
8.2	No reset in watchdog example? .....	15			
<b>9</b>	<b>Miscellaneous .....</b>	<b>16</b>			

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.