

AN14431

PN7160/PN7220 configuration files

Rev. 2.0 — 4 February 2025

Application note

Document information

| Information | Content |
|-------------|---|
| Keywords | PN7160, PN7220, configuration files, Android |
| Abstract | This document provides information about the PN7160/PN7220 configuration files and the settings supported by each IC. |



1 Introduction

This document provides information about the PN7160/PN7220 configuration files and the settings supported by each IC.

2 Configuration files

[Table 1](#) explains which configuration file is valid for the specific IC and where to push it on the target DH.

Table 1. Configuration files used by the IC and their location

| Configuration file | IC | Location | Description |
|--|------------------|-------------|---|
| libnfc-nci.conf | PN7160 PN7220 | system/etc/ | System file from Android. |
| libnfc-nxp.conf | PN7160 PN7220 | vendor/etc/ | General HAL configuration and flags that are common between PN7160 and PN7220. |
| libnfc-nxp-eeprom.conf | PN7220 | vendor/etc/ | A number of EEPROM configurations that are usually changed by the user. |
| libnfc-nxp-rfExt.conf | PN7220 | vendor/etc/ | All settings that users can set in NFC Cockpit. Note: Ensure that there are no mismatches between common settings in <i>libnfc-nxp-eeprom.conf</i> and <i>libnfc-nxp-rfExt.conf</i> . |
| libemvco-nxp.conf | PN7220 | vendor/etc/ | Debugging purposes for EMVCo mode. |

libnfc-nci.conf, *libnfc-nxp.conf*, *libnfc-nxp-eeprom.conf*, and *libnfc-nxp-rfExt.conf* are used on NFC MW stack bring-up (Android host boot, NFC stack ON, etc.).

MW uses the configuration files in the following sequence:

1. *libnfc-nci.conf*
2. *libnfc-nxp.conf*
3. *libnfc-nxp-eeprom.conf*
4. *libnfc-nxp-rfExt.conf*

The *libemvco-nxp.conf* configuration file is only used when the user switches from NFC Forum mode to EMVCo mode. It is used for debugging purposes.

3 *libnfc-nci.conf*

[Table 2](#) shows all the settings in *libnfc-nci.conf*. The column *When is it used?* describes when the flag is used.

Table 2. Settings in *libnfc-nci.conf*

| Setting name | Support | When is it used? |
|---|--|--|
| APPL_TRACE_LEVEL | Not supported | On every Android boot or NFC service bring-up. |
| PROTOCOL_TRACE_LEVEL | Not supported | On every Android boot or NFC service bring-up. |
| NFC_DEBUG_ENABLED | Supported | On every Android boot or NFC service bring-up. |
| NFA_STORAGE | Supported | On every Android boot or NFC service bring-up. |
| HOST_LISTEN_TECH_MASK | Supported | On every Android boot or NFC service bring-up. |
| SCREEN_OFF_POWER_STATE | Not supported Note: Do not change the default value (0x01). Otherwise, it can break the NFC stack on the PN7160. | On every Android boot or NFC service bring-up. |
| NCI_HAL_MODULE | Not supported | On every Android boot or NFC service bring-up. |
| POLLING_TECH_MASK | Supported | On every Android boot or NFC service bring-up. |
| P2P_LISTEN_TECH_MASK | Not supported Note: Supported by PN7160 up until Android 13. | On every Android boot or NFC service bring-up. |
| PRESERVE_STORAGE | Supported | On every Android boot or NFC service bring-up. |
| AID_MATCHING_MODE | Supported | On every Android boot or NFC service bring-up. |
| NFA_MAX_EE_SUPPORTED | Supported | On every Android boot or NFC service bring-up. |
| OFFHOST_AID_ROUTE_PWR_STATE | Not supported | On every Android boot or NFC service bring-up. |
| NCI_RESET_TYPE | Supported on PN7160 | On every Android boot or NFC service bring-up. |
| NFA_DM_DISC_DURATION_POLL | Supported | On every Android boot or NFC service bring-up. |

3.1 NFC_DEBUG_ENABLED

With this flag users can enable extended logs in MW. It is used for debugging.

Table 3. NFC_DEBUG_ENABLED

| Value | Description |
|----------------|------------------------|
| 0x00 | Extended logs disabled |
| 0x01 (Default) | Extended logs enabled |

3.2 NFA_STORAGE

This flag defines the storage location in which NFC service stores persistent data.

By default, the storage location is set to `/data/vendor/nfc`.

3.3 HOST_LISTEN_TECH_MASK

With this flag users can set which technology the controller should use for card emulation.

Note: PN7160 supports card emulation for Type A and Type B, while PN7220 only supports Type A. Both ICs supports T4T.

3.4 POLLING_TECH_MASK

With this flag users can set which technology the controller uses for polling.

3.5 PRESERVE_STORAGE

Verify the content of all nonvolatile stores.

3.6 AID_MATCHING_MODE

With this flag users can set how the system matches the AID.

3.7 NFA_MAX_EE_SUPPORTED

Maximum EE supported number.

3.8 NCI_RESET_TYPE

With this flag users can select how often the configuration shall be reset.

Table 4. NCI_RESET_TYPE

| Value | Reset type |
|----------------|--|
| 0x00 (default) | Reset the configuration every time |
| 0x01 | Reset the configuration only once every boot |
| 0x02 | Keep configuration |

3.9 NFA_DM_DISC_DURATION_POLL

This flag defines the TOTAL_DURATION that is specified in the NCI specification ([\[1\]](#)) and NFC controller user manual (PN7160 [\[2\]](#), PN7220 [\[3\]](#)).

There is a difference between the definition of TOTAL_DURATION in NCI specification and NFC controller user manuals.

In the NCI specification, TOTAL_DURATION specifies the duration which includes polling mode + listen mode:

- TOTAL_DURATION = Polling mode duration + Listen mode duration

In the NFC controller user manuals, TOTAL_DURATION specifies the duration of listen mode only:

- TOTAL_DURATION = Listen mode duration

PN7160/PN7220 support only TOTAL_DURATION that is specified in the user manual.

Default value: 500 (500 ms)

4 libnfc-nxp.conf

Table 5 shows all available settings in *libnfc-nxp.conf*. The column *When is it used?* describes when the flag is used. Some flags from this configuration file have additional checks to avoid multiple writes into EEPROM.

Table 5. Settings in *libnfc-nxp.conf*

| Setting name | PN7160 | PN7220 | When is it used? |
|--|---------------|---------------|---|
| NXPLOG_EXTNS_LOGLEVEL (LOGGING) | Supported | Supported | On every Android boot or NFC service bring-up |
| NXPLOG_NCIHAL_LOGLEVEL (LOGGING) | Supported | Supported | On every Android boot or NFC service bring-up |
| NXPLOG_NCIX_LOGLEVEL (LOGGING) | Supported | Supported | On every Android boot or NFC service bring-up |
| NXPLOG_NCIR_LOGLEVEL (LOGGING) | Supported | Supported | On every Android boot or NFC service bring-up |
| NXPLOG_FWDNLD_LOGLEVEL (LOGGING) | Supported | Supported | On every Android boot or NFC service bring-up |
| NXPLOG_TML_LOGLEVEL (LOGGING) | Supported | Supported | On every Android boot or NFC service bring-up |
| NXP_NFC_DEV_NODE | Supported | Supported | On every Android boot or NFC service bring-up |
| MIFARE_READER_ENABLED | Supported | Supported | On every Android boot or NFC service bring-up |
| NXP_FW_TYPE | Supported | Supported | On every Android boot or NFC service bring-up |
| NXP_AGC_DEBUG_ENABLE | Not supported | Not supported | Not supported |
| NXP_ACT_PROP_EXTN | Supported | Supported | On every Android boot or NFC service bring-up |
| NXP_NFC_PROFILE_EXTN | Supported | Not supported | On every Android boot or NFC service bring-up |
| NXP_I2C_FRAGMENTATION_ENABLED | Supported | Supported | On every Android boot or NFC service bring-up |
| NFA_PROPRIETARY_CFG | Supported | Supported | On every Android boot or NFC service bring-up |
| NXP_EXT_TVDD_CFG | Supported | Not supported | If the config file was modified or if a FW download is performed |
| NXP_EXT_TVDD_CFG_X | Supported | Not supported | If the config file was modified or if a FW download is performed |
| NXP_CORE_CONF | Supported | Supported | PN7160: On every Android boot or NFC service bring-up PN7220: If the config file was modified or if an FW update happens or if the NXP_SET_CONFIG_ALWAYS flag is set to 0x01 |
| NXP_SET_CONFIG_ALWAYS | Not supported | Supported | On every Android boot or NFC service bring-up |
| NXP_RF_CONF_BLK_X | Supported | Not supported | If the config file was modified or if a FW download is performed |
| ISO_DEP_MAX_TRANSCEIVE | Not supported | Not supported | Not supported |

Table 5. Settings in *libnfc-nxp.conf*...continued

| Setting name | PN7160 | PN7220 | When is it used? |
|--|-----------|---------------|--|
| PRESENCE_CHECK_ALGORITHM | Supported | Not supported | On every Android boot or NFC service bring-up |
| NXP_FLASH_CONFIG | Supported | Supported | On every Android boot or NFC service bring-up |
| NXP_CHIP_TYPE | Supported | Supported | On every Android boot or NFC service bring-up |
| NXP_SUPPORT_NON_STD_CARD | Supported | Supported | On every Android boot or NFC service bring-up |
| NXP_NON_STD_CARD_TIMEDIFF | Supported | Supported | On every Android boot or NFC service bring-up |
| NXP_SYS_CLOCK_TO_CFG | Supported | Not supported | If there is a mismatch between the settings in the configuration file and what is stored in EEPROM |
| NXP_T4T_NFCEE_ENABLE | Supported | Not supported | On every Android boot or NFC service bring-up |
| DEFAULT_T4TNFCEE_AID_POWER_STATE | Supported | Not supported | On every Android boot or NFC service bring-up |

Some settings from *libnfc-nxp.conf* are common between PN7160 and PN7220 but are not located in different configuration files. [Table 6](#) shows the settings and their locations for the specific IC.

Table 6. Setting and configuration file mapping

| Setting name | PN7160 | PN7220 | When is it used? |
|--------------------------------------|------------------------|--------------------------------|---|
| NXP_SYS_CLK_SRC_SEL | <i>libnfc-nxp.conf</i> | <i>libnfc-nxp-eeeprom.conf</i> | PN7160: If there is a mismatch between the settings in the configuration file and what is stored in EEPROM. PN7220: If the config file was modified or if an FW download happened. |
| NXP_SYS_CLK_FREQ_SEL | <i>libnfc-nxp.conf</i> | <i>libnfc-nxp-eeeprom.conf</i> | PN7160: If there is a mismatch between the settings in the configuration file and what is stored in EEPROM. PN7220: If the config file was modified or if an FW download happened. |
| NXP_CORE_CONF_EXTN | <i>libnfc-nxp.conf</i> | <i>libnfc-nxp-eeeprom.conf</i> | PN7160: On every Android boot or NFC service bring-up. PN7220: If the config file was modified or if an FW update happens or if the NXP_SET_CONFIG_ALWAYS flag is set to 0x01. |

4.1 LOGGING

The following flags can be configured to set extended logs from the MW stack.

- `NXPLOG_EXTNS_LOGLEVEL`
 - Configuration for extns logging level
- `NXPLOG_NCIHAL_LOGLEVEL`
 - Configuration for enabling logging of HAL
- `NXPLOG_NCIX_LOGLEVEL`
 - Configuration for enabling logging of NCI TX packets
- `NXPLOG_NCIR_LOGLEVEL`
 - Configuration for enabling logging of NCI RX packets
- `NXPLOG_FWDNLD_LOGLEVEL`
 - Configuration for enabling logging of FW download functionality
- `NXPLOG_TML_LOGLEVEL`
 - Configuration for enabling logging of TML

All flags are set to 0x03 by default.

Flags can be set to the following values:

Table 7. Possible log levels

| Name | Value |
|--------------------------------------|-------|
| <code>NXPLOG_DEFAULT_LOGLEVEL</code> | 0x01 |
| <code>NXPLOG_DEBUG_LOGLEVEL</code> | 0x03 |
| <code>NXPLOG_WARN_LOGLEVEL</code> | 0x02 |
| <code>NXPLOG_ERROR_LOGLEVEL</code> | 0x01 |
| <code>NXPLOG_SILENT_LOGLEVEL</code> | 0x00 |

4.2 NXP_NFC_DEV_NODE

With this flag users can set the location of the device node and its name.

The default value that is also connected to the driver implementation is `/dev/nxpncf`.

4.3 MIFARE_READER_ENABLED

The flag is used to enable MIFARE extensions in the reader.

4.4 NXP_FW_TYPE

Defines the FW file type.

Table 8. `NXP_FW_TYPE`

| Value | File type |
|-------|-----------|
| 0x01 | .so |
| 0x02 | .bin |

4.5 NXP_ACT_PROP_EXTN

This flag is used to enable NXP proprietary settings for PN7160/PN7220.

To enable proprietary settings, use the following command:

```
NXP_ACT_PROP_EXTN={2F, 02, 00}
```

4.6 NXP_NFC_PROFILE_EXTN

This setting is used in PN7160 to switch from NFC Forum profile to EMVCo profile and back:

```
NXP_NFC_PROFILE_EXTN={20, 02, 05, 01, A0, 44, 01, 00} // (00 = NFC Forum) / (01 = EMVCo)
```

For more information, see [\[2\]](#).

4.7 NXP_I2C_FRAGMENTATION_ENABLED

Use this flag to enable and disable I2C fragmentation.

4.8 NFA_PROPRIETARY_CFG

Set vendor specific proprietary protocol and discovery configuration.

4.9 NXP_EXT_TVDD_CFG

This setting is used only by PN7160. Users can select the TVDD configuration using [Section "NXP_EXT_TVDD_CFG_X"](#).

4.10 NXP_EXT_TVDD_CFG_X

TVDD configuration setting for PN7160. For instructions on how to set the configuration, refer to [\[2\]](#) section "Configuration".

4.11 NXP_CORE_CONF

With this flag users can set the core configuration settings. For PN7160, this data is stored in the EEPROM. For PN7220, this data is stored in the RAM and can be flashed to the EEPROM by enabling the [NXP_FLUSH_SRAM_TO_FLASH_ENABLED](#) flag (located in [libnfc-nxp-EEPROM.conf](#)).

Note: The supported NCI Specification settings depend on the IC in use. Refer to the NCI Specification ([\[1\]](#)) and the dedicated user manual (PN7160 [\[2\]](#), PN7220 [\[3\]](#)).

4.12 NXP_CORE_CONF_EXTN

NXP Proprietary core configuration extensions. Refer to the user manuals (PN7160 [\[2\]](#), PN7220 [\[3\]](#)).

Note: For PN7220, this flag is used in [libnfc-nxp-EEPROM.conf](#) (previously in [libnfc-nxp.conf](#)), For PN7160, this flag is used in [libnfc-nxp.conf](#).

Note: The settings stored in this flag are written to EEPROM. Beware of limited read and write cycles of the EEPROM.

4.13 NXP_SET_CONFIG_ALWAYS

Note: For debugging only.

With this flag users can enable the MW stack to set the `NXP_CORE_CONF` on each NFC stack bring-up (every Android boot, NFC stack restart, etc.).

Table 9. NXP_SET_CONFIG_ALWAYS

| Value | Description |
|-------|---------------------|
| 0x00 | ON MODIFY (Default) |
| 0x01 | ALWAYS |

By default, the Android stack checks the configuration file and send the settings to PN7220 only when the configuration file was modified. Using the `NXP_SET_CONFIG_ALWAYS`, the settings can be sent to PN7220 on every NFC stack bring-up.

4.14 NXP_RF_CONF_BLK_X

This flag is used only by PN7160. User can set (additional) RF configuration settings with this flag.

```

NXP_RF_CONF_BLK_1={ 20, 02, 4C, 09,
    A0, 0D, 03, 78, 0D, 02,
    A0, 0D, 03, 78, 14, 02,
    A0, 0D, 06, 4C, 44, 65, 09, 00, 00,
    A0, 0D, 06, 4C, 2D, 05, 35, 1E, 01,
    A0, 0D, 06, 82, 4A, 55, 07, 00, 07,
    A0, 0D, 06, 44, 44, 03, 04, C4, 00,
    A0, 0D, 06, 46, 30, 50, 00, 18, 00,
    A0, 0D, 06, 48, 30, 50, 00, 18, 00,
    A0, 0D, 06, 4A, 30, 50, 00, 08, 00
}
NXP_RF_CONF_BLK_2={
    20, 02, 1E, 03,
    A0, 38, 04, 24, 10, 0B, 00,
    A0, AA, 04, 38, 04, 98, 08,
    A0, AF, 0C, 83, D5, 78, 80, 00, 83, D5, 78, 80, 00, 77, 08
}
...
NXP_RF_CONF_BLK_X={ ... }

```

Refer to [\[2\]](#) section "Configuration" for instructions on how to retrieve the TAG ID.

4.15 PRESENCE_CHECK_ALGORITHM

This flag is used only by PN7160. Users can define what type of presence check algorithm is used for T4T.

Table 10. PRESENCE_CHECK_ALGORITHM

| Value | Algorithm |
|-------|--|
| 0x00 | NFA_RW_PRESENCE_CHK_DEFAULT (MW stack will select algorithm) |
| 0x01 | NFA_RW_PRESENCE_CHK_I_BLOCK |
| 0x02 | NFA_RW_PRESENCE_CHK_ISO_DEP_NAK |

4.16 NXP_FLASH_CONFIG

This flag is used to select when a FW update is executed.

Table 11. NXP_FLASH_CONFIG

| Value | Description |
|----------------|---|
| 0x01 | FLASH_UPPER_VERSION |
| 0x02 (Default) | FLASH_DIFFERENT_VERSION |
| 0x03 | FLASH_ALWAYS <i>Note: For debugging only, not intended for production.</i> |
| 0x04 | FLASH_BLOCKED |

4.17 NXP_CHIP_TYPE

Flag to select which IC is used.

Table 12. NXP_CHIP_TYPE

| Value | Description |
|-------|-------------|
| 0x01 | PN7160 |
| 0x04 | PN7220 |

4.18 NXP_SUPPORT_NON_STD_CARD

This flag is used to enable or disable non-standard tag reading.

4.19 NXP_NON_STD_CARD_TIMEDIFF

This flag is used to set the valid time difference range for non-standard tag detection from first activation fail to next discovery.

4.20 NXP_SYS_CLK_SRC_SEL

Set the source for the clock.

Table 13. NXP_SYS_CLK_SRC_SEL

| Value | Clock source |
|-------|--------------|
| 0x01 | XTAL |
| 0x02 | PLL |

Refer to [\[2\]](#) section "Configuration" for instructions on how to retrieve the TAG ID.

4.21 NXP_SYS_CLK_FREQ_SEL

Sets the system clock frequency selection in case of PLL setting. Users must check the configuration files for the supported values.

Refer to [\[2\]](#) section "Configuration" for instructions on how to retrieve the TAG ID.

4.22 NXP_SYS_CLOCK_TO_CFG

This flag is only used for PN7160. Sets the timeout value to be used for clock request acknowledgment. Refer to [2] section "Configuration", for instructions on how to retrieve the TAG ID.

4.23 NXP_T4T_NFCEE_ENABLE

This flag is only used for PN7160. Switching between T4T_NFCEE and host card emulation. See [4] for more information.

Table 14. NXP_T4T_NFCEE_ENABLE

| Value | Description |
|-------|-------------------|
| 0x01 | Enable T4T_NFCEE |
| 0x00 | Disable T4T_NFCEE |

The following commands are send to PN7160 when this flag is set to 0x01:

```
220000 // NFCEE_DISCOVER_CMD
42010100 // NFCEE_DISCOVER_RSP
610A0B0200031080040003108104 // NFCEE_DISCOVER_NTF

2201021001 // NFCEE_MODE_SET_CMD
42010100 // NFCEE_MODE_SET_RSP
62010100 // NFCEE_MODE_SET_NTF
610A0B0200031080040003108104 // RF_NFCEE_DISCOVERY_REQ_NTF
```

4.24 DEFAULT_T4TNFCEE_AID_POWER_STATE

The flag is only used for PN7160. It sets the AID power state for T4TNFCEE. It is reflected in RF_SET_LISTEN_MODE_ROUTING_CMD.

Example:

```
21 01 2C 00 07 02 09 10 3B D2 76 00 00 85 01 01 03 04 00 00 FE FE 01 03 00 11 04
01 03 00 01 05 00 03 00 39 00 00 03 00 39 01 00 03 00 39 02
```

Note: 3B is the default value.

5 *libnfc-nxp-EEPROM.conf*

[Table 15](#) shows all settings in *libnfc-nxp-EEPROM.conf*. The column *When is it used?* describes when the flag is used.

Table 15. Settings in *libnfc-nxp-EEPROM.conf*

| Settings | When is it used? |
|---|---|
| NXP_SYS_CLK_SRC_SEL | If the config file was modified or if a FW download is performed. |
| NXP_SYS_CLK_FREQ_SEL | If the config file was modified or if a FW download is performed. |
| NXP_ENABLE_DISABLE_STANBY | On every Android boot or NFC service bring-up. |
| NXP_ENABLE_DISABLE_LPCD | If the config file was modified or if a FW download is performed. |
| NXP_HCE_SENS_RES | If the config file was modified or if a FW download is performed. |
| NXP_HCE_NFC_ID1 | If the config file was modified or if a FW download is performed. |
| NXP_HCE_SEL_RES | If the config file was modified or if a FW download is performed. |
| NXP_HCE_RNDM_UID_ENB | If the config file was modified or if a FW download is performed. |
| NXP_CORE_CONF_EXTN | If the config file was modified or if a FW download is performed. |
| NXP_IS_TDA_CHIP_PRESENT | If the config file was modified or if a FW download is performed. |
| NXP_ENABLE_DISABLE_PPS_EXCHANGE | If the config file was modified or if a FW download is performed. |
| NXP_PCD_SETTINGS | If the config file was modified or if a FW download is performed. |
| NXP_FLUSH_SRAM_TO_FLASH_ENABLE | If the config file was modified or if a FW download is performed. |

5.1 NXP_SYS_CLK_SRC_SEL

Sets the source for the clock.

Table 16. Possible clock sources

| Value | Clock source |
|-------|--------------|
| 0x01 | XTAL |
| 0x02 | PLL |

This setting is not used for any NCI command, but needs to be aligned with NXP_SYS_CLK_FREQ_SEL.

5.2 NXP_SYS_CLK_FREQ_SEL

Sets the system clock frequency selection in case of PLL setting. Users must check the configuration files for the supported values.

NCI Command:

```
20 02 0C 01 A2 02 08 FF 08 F6 01 00 33 64 10 // CORE_SET_CONFIG_CMD
P
```

The system will first perform CORE_GET_CONFIG_CMD with TAG ID "A2 02" to check if the existing setting matches the one set by the user. In case of a mismatch, the system proceeds with the CORE_SET_CONFIG_CMD mentioned above.

```
20 03 03 01 A2 02 // CORE_GET_CONFIG_CMD
40 03 0D 00 01 A2 02 08 FF 08 F6 01 00 33 64 10 // CORE_GET_CONFIG_RS
```

5.3 NXP_ENABLE_DISABLE_STANDBY

For PN7220 there are two possibilities to disable/enable the Standby mode.

- 1. Disable/Enable standby in RAM (CORE_SET_POWER_MODE_CMD in [3])
- 2. Disable/Enable standby in EEPROM

Note: For more information, see [3].

NXP_ENABLE_DISABLE_STANDBY is linked to RAM setting and can be configured also after bootup (in runtime). By default PN7220 is using EEPROM settings, but the user can bypass EEPROM setting with NXP_ENABLE_DISABLE_STANDBY. If the user performs a hard or soft reset of PN7220, the RAM settings are lost and PN7220 will again use the EEPROM setting.

Table 17. Possible settings

| Value | Standby |
|-------|---------|
| 0x00 | Disable |
| 0x01 | Enable |

NCI Command sent to PN7220:

```
2F 00 01 00
```

Based on the value of this flag in the configuration file, the command above is sent. This occurs on every NFC stack bring-up (Android boot).

5.4 NXP_ENABLE_DISABLE_LPCD

This flag is used to enable or disable LPCD. After changing this flag, the EEPROM setting is changed. It is important to understand that this setting should ideally be configured only once, otherwise, the maximum number of EEPROM read/write cycles can be reached and the IC corrupted.

Note: The same setting can be configured with the NFC Cockpit tool. In this case, enable/disable can be configured in the `libnfc-nxp-rfExt.conf` file via NCI Command. It is important that the user configures this flag according to the changes in `libnfc-nxp-rfExt.conf`, otherwise, there will be mismatches between the settings and the one which is used last by the MW stack will be used.

Table 18. Possible settings

| Value | LPCD |
|-------|---------|
| 0x00 | Disable |
| 0x01 | Enable |

NCI Command:

```
20 02 0C 01 A2 7C 08 A3 02 1F BB 00 08 50 00 // CORE_SET_CONFIG_CMD
```

Note: To enable LPCD, standby musts be enabled.

5.5 NXP_HCE_SENS_RES

This flag configures the response to ReqA / ATQA.

Default value: {04, 00}

NCI Command:

```
20 02 0C 01 A2 78 08 37 00 00 00 10 04 00 AA // CORE_SET_CONFIG_CMD
```

The system will first perform `CORE_GET_CONFIG_CMD` with TAG ID "A2 78" to check if the existing setting matches the one set by the user. In case of a mismatch, the system proceeds with the `CORE_SET_CONFIG_CMD` mentioned above.

```
20 03 03 01 A2 78 // CORE_GET_CONFIG_CMD
40 03 0D 00 01 A2 78 08 37 00 00 00 10 04 00 AA // CORE_GET_CONFIG_RSP
```

5.6 NXP_HCE_NFC_ID1

This flag sets the configuration of the UID. The first byte is fixed to 0x08.

Note: If a random UID is selected, this flag is not used.

Default value: {AA, BB, CC}

NCI Command:

```
20 02 0C 01 A2 78 08 37 00 00 00 10 04 00 AA // CORE_SET_CONFIG_CMD
20 02 0C 01 A2 79 08 BB CC 20 01 FE 01 14 01 // CORE_SET_CONFIG_CMD
```


The system will first perform CORE_GET_CONFIG_CMD with TAG ID "A2 78" and "A2 79" to check if the existing setting matches the one set by the user. In case of a mismatch, the system proceeds with the CORE_SET_CONFIG_CMD:

```
20 03 03 01 A2 78 // CORE_GET_CONFIG_CMD
40 03 0D 00 01 A2 78 08 37 00 00 00 10 04 00 AA // CORE_GET_CONFIG_RSP
20 03 03 01 A2 79 // CORE_GET_CONFIG_CMD
20 02 0C 01 A2 79 08 BB CC 20 01 FE 01 14 01 // CORE_GET_CONFIG_CMD
```

5.7 NXP_HCE_SEL_RES

Flag to configure the response to SAK.

Default value: 0x20

NCI Command:

```
20 02 0C 01 A2 79 08 DD CC 20 01 FE 01 14 01 // CORE_SET_CONFIG_CMD
```

The system will first do CORE_GET_CONFIG_CMD with TAG ID "A2 79" to check if the existing setting matches the one set by the user. In case of a mismatch, the system proceeds with the CORE_SET_CONFIG_CMD mentioned above.

```
20 03 03 01 A2 79 // CORE_GET_CONFIG_CMD
40 03 0D 00 01 A2 79 08 DD CC 20 01 FE 01 14 01 // CORE_GET_CONFIG_RSP
```

5.8 NXP_HCE_RNDM_UID_ENB

Flag to disable/enable random UID.

Table 19. Possible settings

| Value | UID |
|----------------|---------------------------|
| 0x00 (default) | Use UID stored in EEPROM |
| 0x01 | Randomly generate the UID |

NCI Command:

```
20 02 0C 01 A2 7B 08 42 84 85 D0 FF 00 06 40 // CORE_SET_CONFIG_CMD
```

The system will first do CORE_GET_CONFIG_CMD with TAG ID "A2 7B" to check if the existing setting matches the one set by the user. In case of a mismatch, the system proceeds with the CORE_SET_CONFIG_CMD mentioned above.

```
20 03 03 01 A2 7B // CORE_GET_CONFIG_CMD
40 03 0D 00 01 A2 7B 08 42 84 85 D0 FF 00 06 40 // CORE_GET_CONFIG_RSP
```

5.9 NXP_IS_TDA_CHIP_PRESENT

This flag indicate if TDA IC is connected on design or not.

Table 20. Possible values

| Value | Description |
|----------------|--------------------------|
| 0x00 (default) | TDA IC is connected. |
| 0x01 | TDA IC is not connected. |

This flag is important in case the user intends to use a contact card or secure element connected to PN7220. In case of a direct SE connection to PN7220, this flag needs be set to 0x01. Otherwise, it can be set to 0x00.

5.10 NXP_ENABLE_DISABLE_PPS_EXCHANGE

This is independent configuration applicable on all slots irrespective of TDA connected or not. This feature (if enabled) is applicable on slots configured as ISO and NOT on EMVCo slot.

Table 21. Possible values

| Value | Description |
|----------------|-----------------------|
| 0x00 (default) | PPS Exchange Disabled |
| 0x01 | PPS Exchange Enabled |

5.11 NXP_PCD_SETTINGS

Configuration to set polling delay between 2 phases (between 5.1 ms and 10 ms) and default value is {EC, 13} (hex) (5100 μ s).

5.12 NXP_FLUSH_SRAM_TO_FLASH_ENABLE

Settings configured with [NXP_CORE_CONF](#) flag in [libnfc-nxp.conf](#) are stored in the RAM. Since all settings are deleted from the RAM when a reset is performed, it can occur that settings configured by user are lost. The `NXP_FLUSH_SRAM_TO_FLASH_ENABLED` flag can be enabled when the user is satisfied with the settings in `NXP_CORE_CONF`. If enabled, this flag triggers a flushing of data from RAM to EEPROM, which ensures that data stored with `NXP_CORE_CONF` are not lost even if a reset or a FW update occurs.

6 libnfc-nxp-rfExt.conf

The *libnfc-nxp-rfExt.conf* can be used to configure all settings mentioned in [5]. Users must pay additional attention to other flags mentioned in this document to ensure that there are no mismatches between the settings in *libnfc-nxp-rfExt.conf* and the settings in other flags.

This configuration file should be used with the NFC Cockpit (version 8.1.0 or higher) (see [6]). The following chapters provide instructions for using the NFC Cockpit to build the configuration file.

Note: Configuration files can also be build manually. However, it is not recommended by NXP as mapping between register addresses and NCI tag ID can lead to incorrect configurations.

Flags from this configuration files are used only if the config file was modified or if an FW update is performed.

6.1 How to use NFC Cockpit tool in combination with libnfc-nxp-rfExt.conf

Minimum required version: 08.01.00.

Note: For information on how to run the NFC Cockpit, see the NFC Cockpit.

When NFC Cockpit is running, the user can use the following approach to build the *nxpnfc-nxp-rfExt.conf*:

- 1. Open the NFC Cockpit and click "1." and then "2." in Figure 1 .

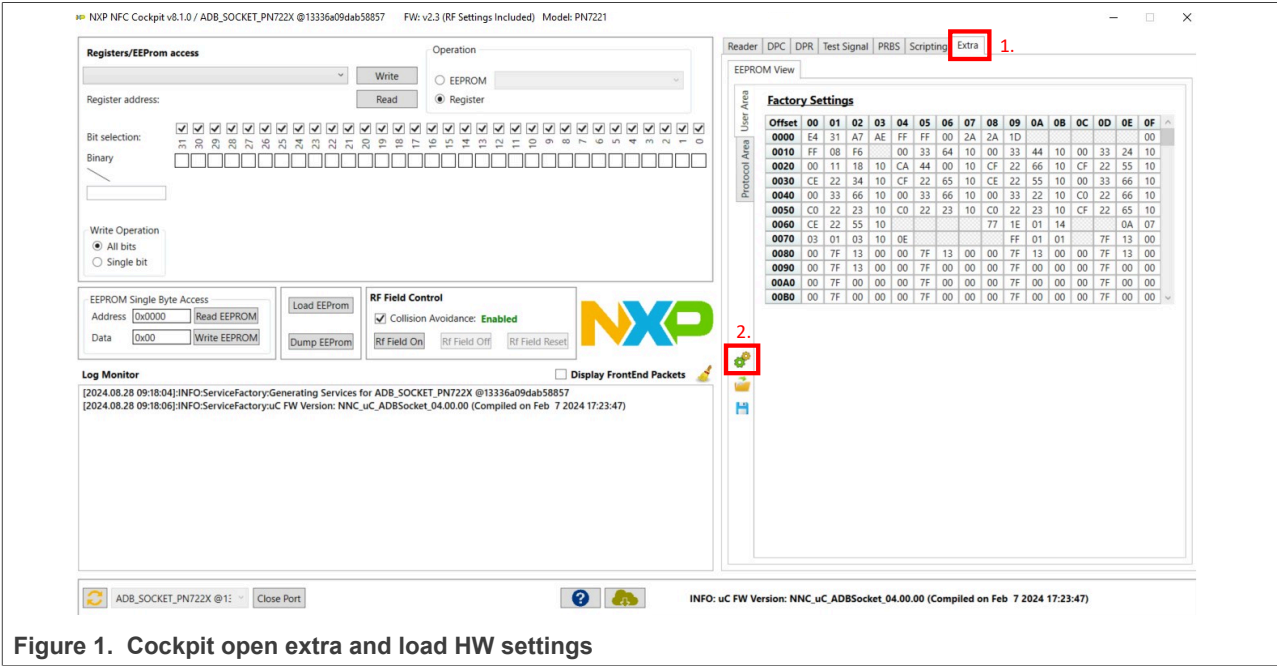


Figure 1. Cockpit open extra and load HW settings

2. The result of clicking the two buttons is:

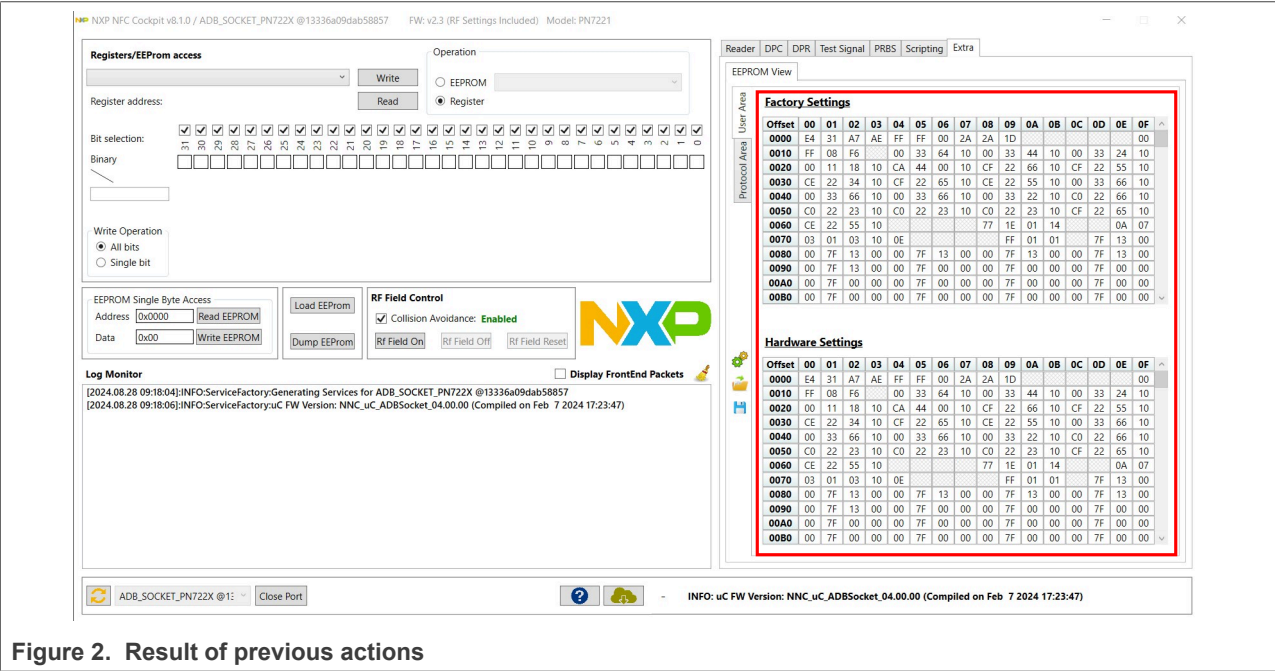


Figure 2. Result of previous actions

Note: Factory settings are default setting from the factory. Hardware settings are current settings stored in EEPROM.

3. To show how to build the configuration file, let's enable the LPCD. This can be done inside "EEPROM Single-Byte Access" with address (0x03E3). First we read it out (see [Figure 3](#)) and then change the bit 7 to 1. After that we write it back to HW ("1.") and click "2." in [Figure 4](#).

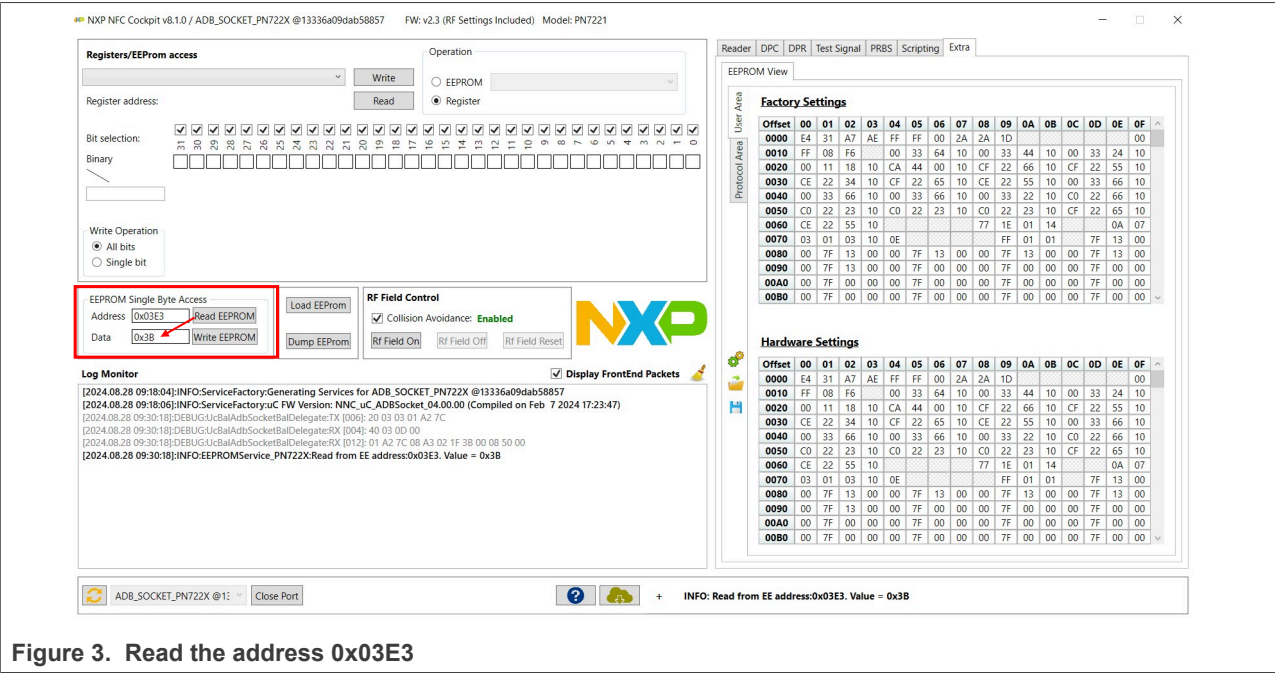


Figure 3. Read the address 0x03E3

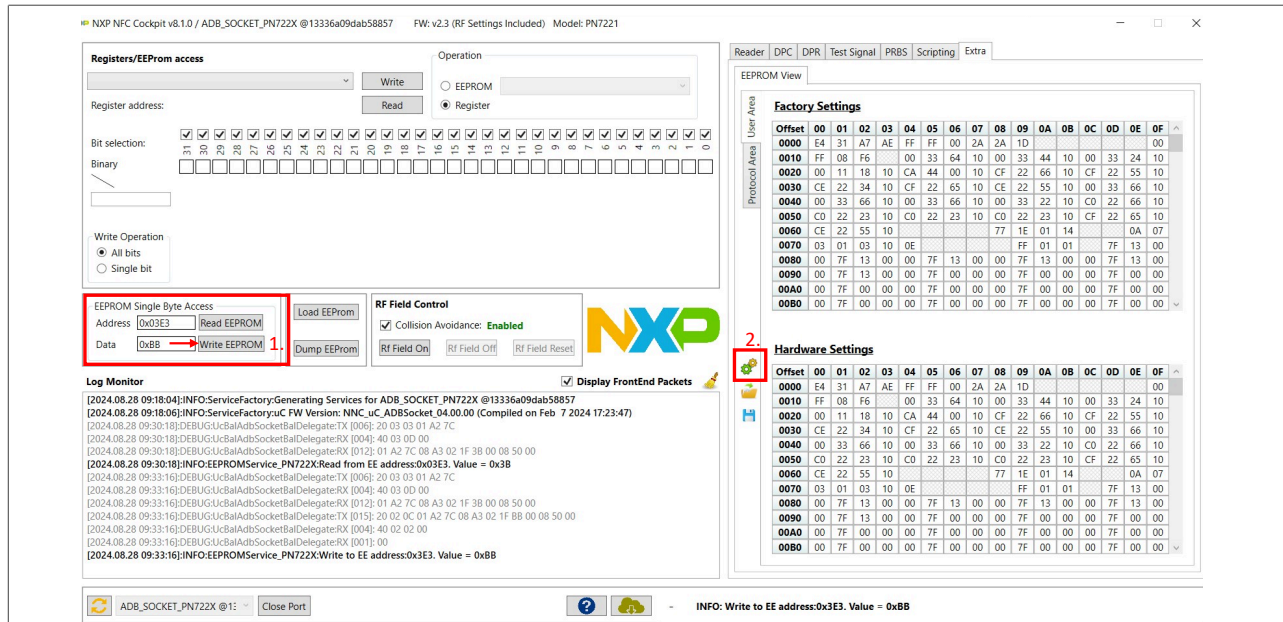


Figure 4. Write to the address 0x03E3 and again load settings from HW ("2.")

4. Users can save the configuration with the "Save" button (see [Figure 5](#)) and name the configuration file as *libnfc-nxp-rfExt.conf*.

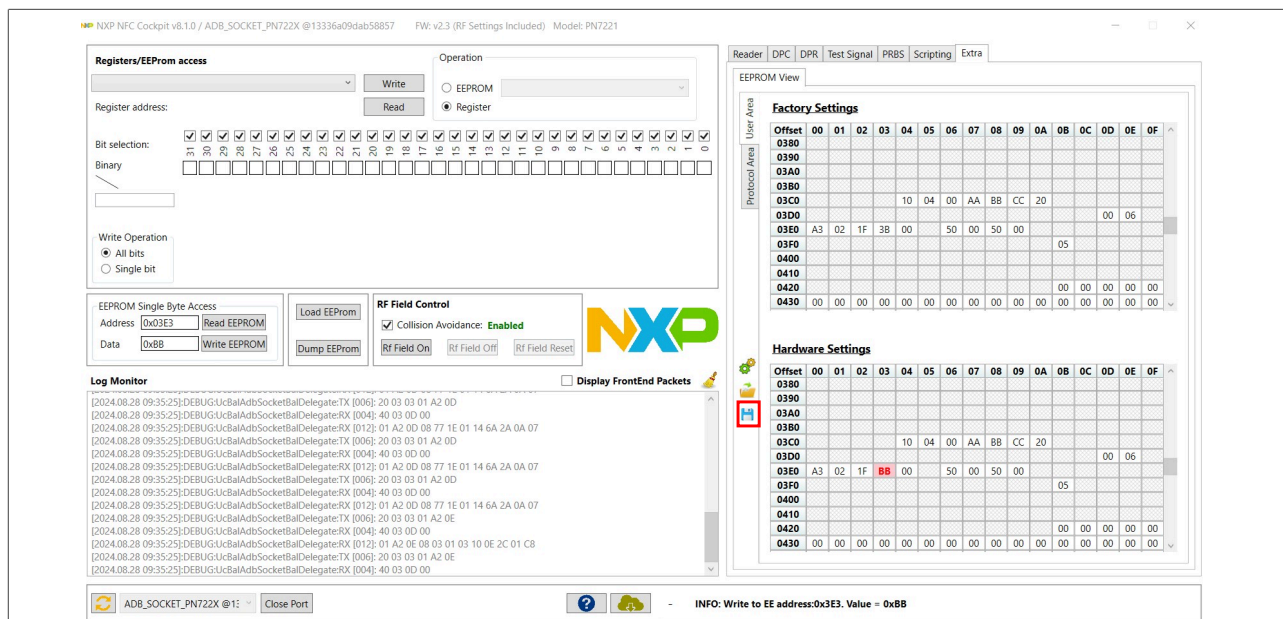


Figure 5. Save changes

5. [Figure 6](#) shows the result. Users can combine multiple NCI commands, or put multiple NXP_RFEXT_CONFIG_X settings inside. It is important that NXP_RFEXT_CONFIG_X matches the number of settings configured in NXP_NUM_OF_RFEXT_CONFIG. (NXP_NUM_OF_RFEXT_CONFIG = X).

```
##### PN7221 libnfc-nxp-rfExt.conf #####

#####
# NXP_NUM_OF_RFEXT_CONFIG : Number of the RF configuration needs to be set.
# MW will read the value of this macro and read the that many config from the
# config file. eg NXP_NUM_OF_RFEXT_CONFIG is 0x02 MW will read NXP_RFEXT_CONFIG_1
# and NXP_RFEXT_CONFIG_2.

NXP_NUM_OF_RFEXT_CONFIG=0x01
#####
# RF-EXT Configuration, Fill the NXP_RFEXT_CONFIG_x with valid NCI pkt.
# eg. NXP_RFEXT_CONFIG_1={20 02 0C 01 A2 00 08 E4 31 A7 AE FF FF 00 2A} => UserArea
# eg. NXP_RFEXT_CONFIG_2={20 02 0A 01 A0 0D 06 00 15 89 22 00 00}      => ProtocolArea

# UserArea Settings
#####
NXP_RFEXT_CONFIG_1={20 02 0C 01 A2 7C 08 A3 02 1F BB 00 08 50 00}

# ProtocolArea Settings
#####
#####
```

Figure 6. Example of *libnfc-nxp-rfExt.conf*

Note: The shown example only presents one of the settings, which can be set in *libnfc-nxp-rfExt.conf* or *libnfc-nxp-EEPROM.conf* (*NXP_ENABLE_DISABLE_LPCD*). The NCI command is the same, which is why it is important that users use only the *libnfc-nxp-EEPROM.conf* or (not recommended by NXP) the change is performed in both configuration options.

7 libemvco-nxp.conf

This configuration file is only valid for PN7220. It needs to be pushed to `/vendor/etc/`. The column *When is it used?* describes when the flag is used.

Table 22. Settings in libnfc-nxp-eeeprom.conf

| Settings | When is it used? |
|---|------------------------------|
| NXP_LOG_EXTNS_LOGLEVEL (LOGGING) | On every mode switch toggle. |
| NXP_LOG_NCIHAL_LOGLEVEL (LOGGING) | On every mode switch toggle. |
| NXP_LOG_NCIX_LOGLEVEL (LOGGING) | On every mode switch toggle. |
| NXP_LOG_NCIR_LOGLEVEL (LOGGING) | On every mode switch toggle. |
| NXP_LOG_TML_LOGLEVEL (LOGGING) | On every mode switch toggle. |
| NXP_EMVCO_DEBUG_ENABLED (LOGGING) | On every mode switch toggle. |
| NXP_EMVCO_DEV_NODE | On every mode switch toggle. |
| NXP_SET_CONFIG | On every mode switch toggle. |
| NXP_GET_CONFIG | On every mode switch toggle. |
| NXP_CT_MAX_WTX_WAIT_TIME | On every mode switch toggle. |

7.1 LOGGING

Following flags can be configured to set extended logs from the MW stack.

- `NXPLOG_EXTNS_LOGLEVEL`
 - Configuration for extns logging level
- `NXPLOG_NCIHAL_LOGLEVEL`
 - Configuration for enabling logging of HAL
- `NXPLOG_NCIX_LOGLEVEL`
 - Configuration for enabling logging of NCI TX packets
- `NXPLOG_NCIR_LOGLEVEL`
 - Configuration for enabling logging of NCI RX packets
- `NXPLOG_TML_LOGLEVEL`
 - Configuration for enabling logging of TML
- `NXP_EMVCO_DEBUG_ENABLED`
 - Configuration for enabling logging of EMVCO

All flags are set to 0x03 by default.

Flags can be set to one of following values:

Table 23. Possible log levels

| Name | Value |
|--------------------------------------|-------|
| <code>NXPLOG_DEFAULT_LOGLEVEL</code> | 0x01 |
| <code>NXPLOG_DEBUG_LOGLEVEL</code> | 0x03 |
| <code>NXPLOG_WARN_LOGLEVEL</code> | 0x02 |
| <code>NXPLOG_ERROR_LOGLEVEL</code> | 0x01 |

7.2 NXP_EMVCO_DEV_NODE

With this flag users can set the location of the device node and its name.

The default value that is also connected to the driver implementation is `/dev/nxpnfc`.

7.3 NXP_SET_CONFIG

Option to set config command for debugging purpose. Sample command given with PCD SETTING.

7.4 NXP_GET_CONFIG

Option to get config command for debugging purpose. Sample command given with PCD SETTING.

7.5 NXP_CT_MAX_WTX_WAIT_TIME

Option to configure the maximum wait time extension for the contact card feature. The maximum wait time should be 107 seconds to pass the CT compliance test.

By default, 2 seconds is the WTX time-out value, if this property is not found or set.

8 Abbreviations and acronyms

Table 24. Abbreviations

| Acronym | Description |
|---------|--------------------------|
| DH | Device Host |
| MW | MiddleWare |
| P2P | Peer-to-Peer |
| NFC | Near Field Communication |
| AID | Application ID |
| FW | FirmWare |
| T4T | Type 4 Tag |
| NCI | NFC Controller Interface |

9 References

- [1] NFC Forum - NFC Controller Interface
- [2] User manual – UM11495 – PN7160 NFC controller ([link](#))
- [3] User manual – UM11810 – PN722X NFC controller ([link](#))
- [4] Application note – PN7160 card emulation ([link](#))
- [5] Datasheet - PN7220 ([link](#))
- [6] Software – NFC Cockpit Configuration Tool for NFC ICs ([link](#))

10 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2024-2025 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

11 Revision history

Table 25. Revision history

| Document ID | Release date | Description |
|---------------|------------------|---|
| AN14431 v.2.0 | 4 February 2025 | <ul style="list-style-type: none"> • Section 2 "Configuration files" updated • Section 3 "libnfc-nci.conf" updated <ul style="list-style-type: none"> – Section 3.2 "NFA_STORAGE" updated – Section 3.9 "NFA_DM_DISC_DURATION_POLL" added • Section 4 "libnfc-nxp.conf" updated <ul style="list-style-type: none"> – Section 4.5 "NXP_ACT_PROP_EXTN" updated – Section 4.6 "NXP_NFC_PROFILE_EXTN" updated – Section 4.11 "NXP_CORE_CONF" updated – Section 4.12 "NXP_CORE_CONF_EXTN" updated – Section 4.13 "NXP_SET_CONFIG_ALWAYS" updated – NXP_PCD_SETTINGS moved to Section 5 "libnfc-nxp-eeprom.conf" • Section 5 "libnfc-nxp-eeprom.conf" updated <ul style="list-style-type: none"> – Section 5.3 "NXP_ENABLE_DISABLE_STANDBY" updated – Section 5.4 "NXP_ENABLE_DISABLE_LPCD" updated – Section 5.9 "NXP_IS_TDA_CHIP_PRESENT" added – Section 5.10 "NXP_ENABLE_DISABLE_PPS_EXCHANGE" added – Section 5.11 "NXP_PCD_SETTINGS" added – Section 5.12 "NXP_FLUSH_SRAM_TO_FLASH_ENABLE" added • Section 6 "libnfc-nxp-rfExt.conf" updated • Section 7 "libemvco-nxp.conf" updated. • Section 9 "References" updated • Section 10 "Note about the source code in the document" updated |
| AN14431 v.1.0 | 2 September 2024 | <ul style="list-style-type: none"> • Initial version |

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) 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.

Tables

| | | | | | |
|----------|---|----|----------|--|----|
| Tab. 1. | Configuration files used by the IC and their location | 3 | Tab. 13. | NXP_SYS_CLK_SRC_SEL | 12 |
| Tab. 2. | Settings in libnfc-nci.conf | 4 | Tab. 14. | NXP_T4T_NFCEE_ENABLE | 13 |
| Tab. 3. | NFC_DEBUG_ENABLED | 5 | Tab. 15. | Settings in libnfc-nxp-eeprom.conf | 14 |
| Tab. 4. | NCI_RESET_TYPE | 5 | Tab. 16. | Possible clock sources | 15 |
| Tab. 5. | Settings in libnfc-nxp.conf | 7 | Tab. 17. | Possible settings | 15 |
| Tab. 6. | Setting and configuration file mapping | 8 | Tab. 18. | Possible settings | 16 |
| Tab. 7. | Possible log levels | 9 | Tab. 19. | Possible settings | 17 |
| Tab. 8. | NXP_FW_TYPE | 9 | Tab. 20. | Possible values | 18 |
| Tab. 9. | NXP_SET_CONFIG_ALWAYS | 11 | Tab. 21. | Possible values | 18 |
| Tab. 10. | PRESENCE_CHECK_ALGORITHM | 11 | Tab. 22. | Settings in libnfc-nxp-eeprom.conf | 23 |
| Tab. 11. | NXP_FLASH_CONFIG | 12 | Tab. 23. | Possible log levels | 24 |
| Tab. 12. | NXP_CHIP_TYPE | 12 | Tab. 24. | Abbreviations | 25 |
| | | | Tab. 25. | Revision history | 28 |

Figures

| | | | | | |
|---------|---|----|---------|--|----|
| Fig. 1. | Cockpit open extra and load HW settings | 19 | Fig. 4. | Write to the address 0x03E3 and again | |
| Fig. 2. | Result of previous actions | 20 | | load settings from HW ("2.") | 21 |
| Fig. 3. | Read the address 0x03E3 | 20 | Fig. 5. | Save changes | 21 |
| | | | Fig. 6. | Example of livnfc-nxp-rfExt.conf | 22 |

Contents

| | | | | | |
|----------|--|-----------|------------|---|-----------|
| 1 | Introduction | 2 | 6.1 | How to use NFC Cockpit tool in combination with libnfc-nxp-rfExt.conf | 19 |
| 2 | Configuration files | 3 | 7 | libemvco-nxp.conf | 23 |
| 3 | libnfc-nci.conf | 4 | 7.1 | LOGGING | 24 |
| 3.1 | NFC_DEBUG_ENABLED | 5 | 7.2 | NXP_EMVCO_DEV_NODE | 24 |
| 3.2 | NFA_STORAGE | 5 | 7.3 | NXP_SET_CONFIG | 24 |
| 3.3 | HOST_LISTEN_TECH_MASK | 5 | 7.4 | NXP_GET_CONFIG | 24 |
| 3.4 | POLLING_TECH_MASK | 5 | 7.5 | NXP_CT_MAX_WTX_WAIT_TIME | 24 |
| 3.5 | PRESERVE_STORAGE | 5 | 8 | Abbreviations and acronyms | 25 |
| 3.6 | AID_MATCHING_MODE | 5 | 9 | References | 26 |
| 3.7 | NFA_MAX_EE_SUPPORTED | 5 | 10 | Note about the source code in the document | 27 |
| 3.8 | NCI_RESET_TYPE | 5 | 11 | Revision history | 28 |
| 3.9 | NFA_DM_DISC_DURATION_POLL | 6 | | Legal information | 29 |
| 4 | libnfc-nxp.conf | 7 | | | |
| 4.1 | LOGGING | 9 | | | |
| 4.2 | NXP_NFC_DEV_NODE | 9 | | | |
| 4.3 | MIFARE_READER_ENABLED | 9 | | | |
| 4.4 | NXP_FW_TYPE | 9 | | | |
| 4.5 | NXP_ACT_PROP_EXTN | 10 | | | |
| 4.6 | NXP_NFC_PROFILE_EXTN | 10 | | | |
| 4.7 | NXP_I2C_FRAGMENTATION_ENABLED | 10 | | | |
| 4.8 | NFA_PROPRIETARY_CFG | 10 | | | |
| 4.9 | NXP_EXT_TVDD_CFG | 10 | | | |
| 4.10 | NXP_EXT_TVDD_CFG_X | 10 | | | |
| 4.11 | NXP_CORE_CONF | 10 | | | |
| 4.12 | NXP_CORE_CONF_EXTN | 10 | | | |
| 4.13 | NXP_SET_CONFIG_ALWAYS | 11 | | | |
| 4.14 | NXP_RF_CONF_BLK_X | 11 | | | |
| 4.15 | PRESENCE_CHECK_ALGORITHM | 11 | | | |
| 4.16 | NXP_FLASH_CONFIG | 12 | | | |
| 4.17 | NXP_CHIP_TYPE | 12 | | | |
| 4.18 | NXP_SUPPORT_NON_STD_CARD | 12 | | | |
| 4.19 | NXP_NON_STD_CARD_TIMEDIFF | 12 | | | |
| 4.20 | NXP_SYS_CLK_SRC_SEL | 12 | | | |
| 4.21 | NXP_SYS_CLK_FREQ_SEL | 12 | | | |
| 4.22 | NXP_SYS_CLOCK_TO_CFG | 13 | | | |
| 4.23 | NXP_T4T_NFCEE_ENABLE | 13 | | | |
| 4.24 | DEFAULT_T4TNFCEE_AID_POWER_STATE | 13 | | | |
| 5 | libnfc-nxp-eeeprom.conf | 14 | | | |
| 5.1 | NXP_SYS_CLK_SRC_SEL | 15 | | | |
| 5.2 | NXP_SYS_CLK_FREQ_SEL | 15 | | | |
| 5.3 | NXP_ENABLE_DISABLE_STANDBY | 15 | | | |
| 5.4 | NXP_ENABLE_DISABLE_LPCD | 16 | | | |
| 5.5 | NXP_HCE_SENS_RES | 16 | | | |
| 5.6 | NXP_HCE_NFC_ID1 | 16 | | | |
| 5.7 | NXP_HCE_SEL_RES | 17 | | | |
| 5.8 | NXP_HCE_RNDM_UID_ENB | 17 | | | |
| 5.9 | NXP_IS_TDA_CHIP_PRESENT | 18 | | | |
| 5.10 | NXP_ENABLE_DISABLE_PPS_EXCHANGE ... | 18 | | | |
| 5.11 | NXP_PCD_SETTINGS | 18 | | | |
| 5.12 | NXP_FLUSH_SRAM_TO_FLASH_ENABLE | 18 | | | |
| 6 | libnfc-nxp-rfExt.conf | 19 | | | |

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