AN13892

PN7160 frequently asked questions

Rev. 1.2 — 6 September 2024 | Application note

Document information

Information	Content
Keywords	PN7160, PMU, CFG1, CFG2, TXLDO Check, DPC, NCI, ECP,ETSI, FCC, symmetrical matching, asymmetrical matching, output power, RX Gain, Card Emulation
Abstract	This document covers frequently asked questions about the PN7160 in a question-answer style.

1 Introduction

This document is a collection of frequently asked question about the PN7160.Most of the questions are covered in greater detail in other documents.

Note: *For more information, refer to the PN7160 product page on nxp.com (see [1]).*

2 Which power configuration to choose – CFG1 or CFG2?

2.1 CFG1

In **CFG 1**, the **Main power supply** (VDD(UP) and VBAT) is taken from **a battery** (e.g., a cell phone battery). So this configuration is optimized for a user case when a battery power supply is used.

In this configuration TXLDO voltage possible settings are 2.7 V, 3 V, 3.3 V, 3.6 V.

VDD(PAD) is supplied by 1.8 V or 3.3 V.

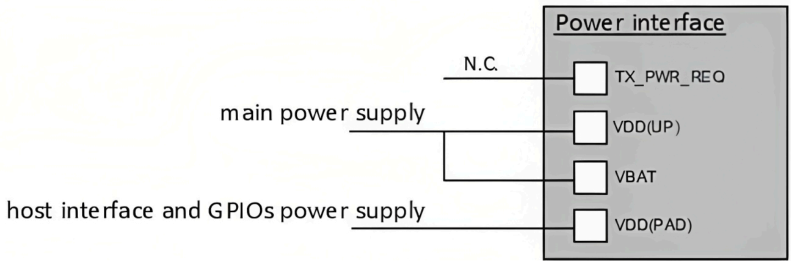


Figure 1. PMU Configuration 1 (CFG1)

Note: The details of the power management configuration are described in the dedicated documents → AN12988 and UM11495.

2.2 CFG2 - DC-DC converter is not used

In **CFG 2**, the VDD(UP) pin is connected to an external power supply. VBAT pin can be connected to the same PMU/Regulator, which supplies VDD(UP). See the example in Figure 3.

In this configuration, TXLDO voltage possible settings are 2.7 V, 3 V, 3.3 V, 3.6 V, 3.9 V, 4.2 V, 4.5 V, 4.7 V, 4.75 V, 5 V, and 5.25 V.

VDD(PAD) is supplied by 1.8 V or 3.3 V.

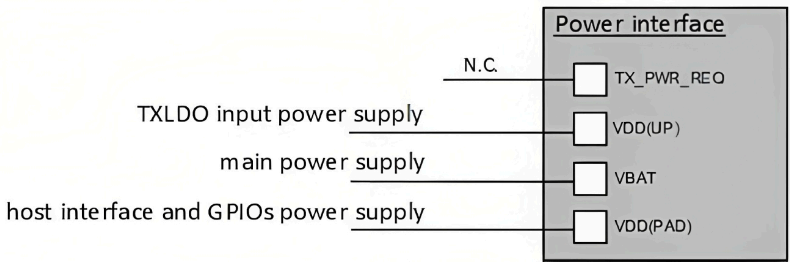


Figure 2. PMU Configuration 2 (CFG2) - DC-DC is not used

Figure 3 shows a common example of the PN7160 power supply using the same external 5 V supply source for VDD(UP) and VBAT.

For this example, the VDD(PAD) pin is supplied by 3.3 V.

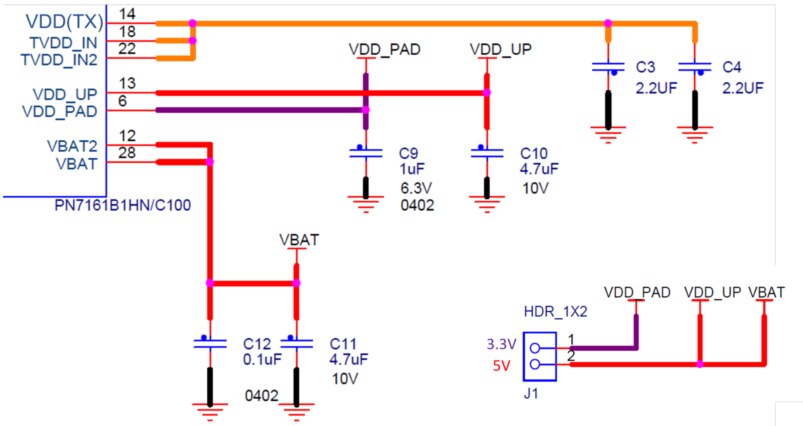


Figure 3. Example of the PMU settings according to CFG2

Note: The details of the power management configuration are described in the dedicated documents → AN12988 and UM11495.

2.3 CFG2 - DC-DC converter is used

In **CFG 2**, a DC-DC converter can be used to increase the VDD(UP) voltage of the main supply voltage. VBAT pin is typically connected to a PMU/regulator, which also supplies the DC-DC converter input. See the example in Figure 4

In this configuration, TXLDO voltage possible settings are 2.7 V, 3 V, 3.3 V, 3.6 V, 3.9 V, 4.2 V, 4.5 V, 4.7 V, 4.75 V, 5 V, and 5.25 V.

VDD(PAD) is supplied by 1.8 V or 3.3 V.

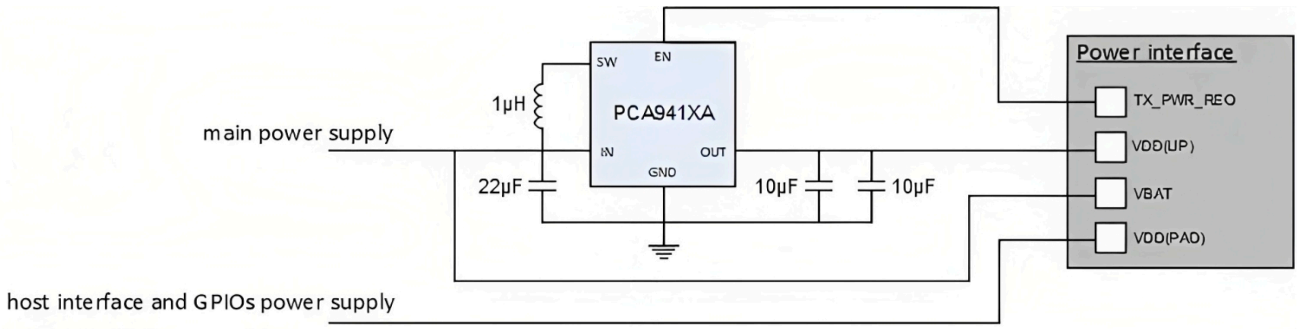


Figure 4. PMU Configuration 2 (CFG2) - DC-DC used

This configuration is useful if you want to use TXLDO **5 V** or **5.25 V** voltage output (Then the VDD(UP) requires to be **5.25 V** or **5.4 V**)

Note: The details of the power management configuration are described in the dedicated application note → AN12988 and UM11495.

2.4 PMU configurator

For easier PMU configuration the "*PMU_CONFIG_PN7160.xlsx*" can be used. This excel sheet is available in the **Design Resources** section on the PN7160 product page.

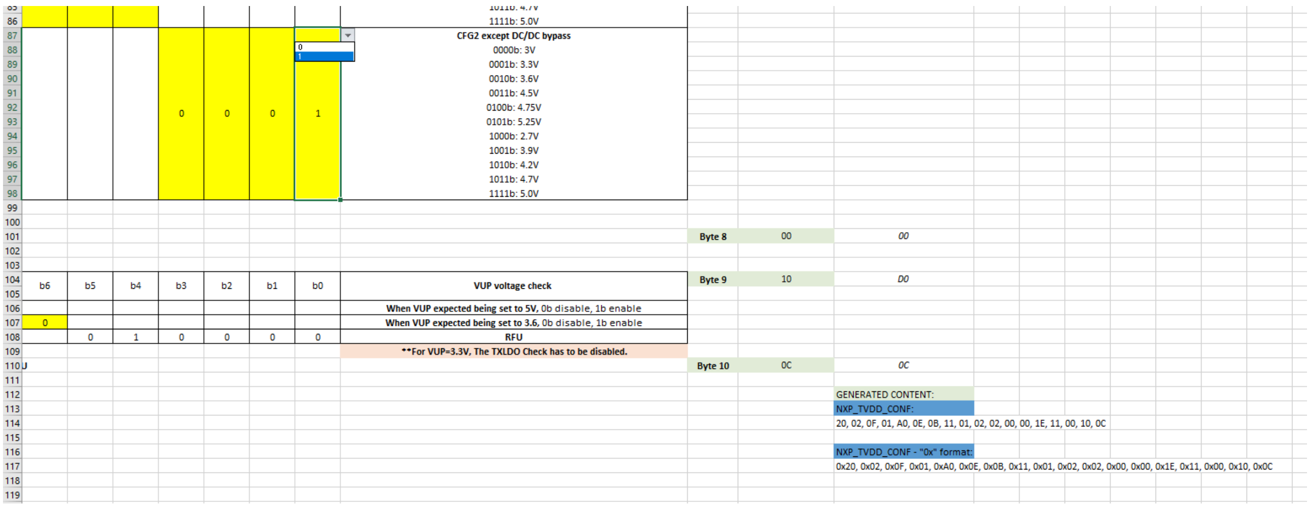


Figure 5. PMU configurator

3 Which power matching to choose – symmetrical or asymmetrical?

The main criterium for considering symmetrical or asymmetrical matching is a **D**ynamic **P**ower **C**ontrol (**DPC**).

If the design is to be as simple as possible, and a sufficiently large antenna is used. (e.g. 40 mm vs 40 mm => 1600 mm^2) + The maximum output power of the IC is not required for the target application. Then the asymmetrical matching is a good choice.

Once the maximum output power is the main criterium and/or small antenna (e.g. 40 mm vs 20 mm => 800 mm^2) is used in the design, then the symmetrical matching + DPC feature has to be used.

More details about the DPC can be found in the dedicated Application note → AN13224.

Don´t use the symmetrical tuning without Dynamic Power Control. This matching is more detuning and loading sensitive → It may lead to TXLDO overcurrent.

3.1 Asymmetrical matching

Asymmetrical tuning

- More robust against detuning and loading → tuning increases under detuning and loading conditions
- Cut off frequency ≈ **20 MHz - 22 MHz**
- No need to use DPC function
- Potentially lower operating volume
- Typically lower output power (Due to the higher target impedance, typically 20-25 Ω)

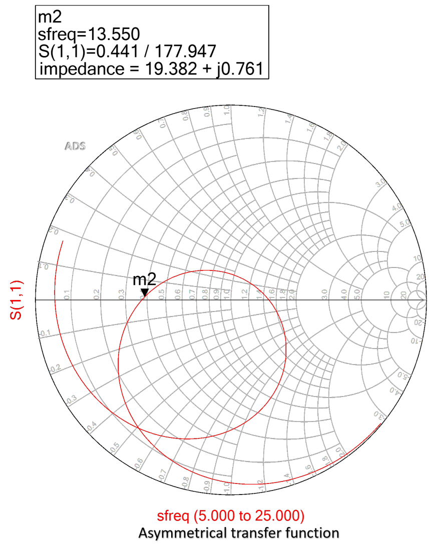


Figure 6. Asymmetrical tuning example

3.2 Symmetrical matching

Symmetrical tuning

- More detuning and loading sensitive → tuning decreases under detuning and loading conditions
- Cut off frequency ≈ **14.4 MHz -14.7 MHz**
- DPC function is required
- Increases the operating volume
- Allows using smaller antennas
- Typically higher output power (Due to the lower target impedance, typically 16-17 Ω)

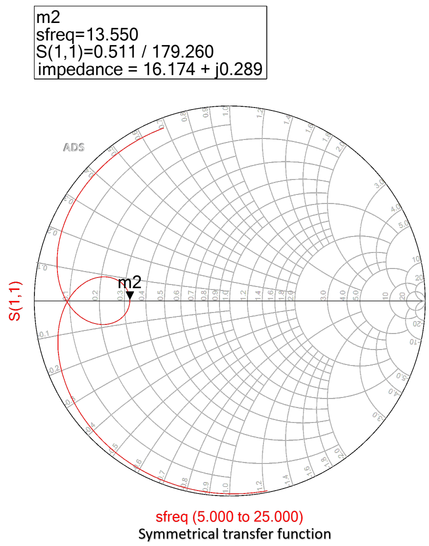


Figure 7. Symmetrical tuning example

4 Dynamic power control - configuration

> The DPC is **disabled** by default on the new ICs. If users want to use symmetrical tuning, the DPC has to be set accordingly.

The dynamic power control description is available in the following application note → AN13224. The **AN13224** application note describes the DPC configuration using the NFC Factory Test Application (see Figure 8). There is also a training video describing DPC settings available.

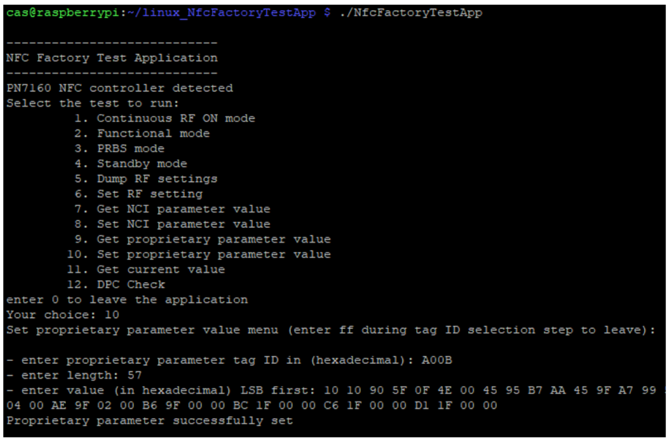


Figure 8. NFC Factory Test Application - DPC settings

The configuration can also be done using the configuration file (*libnfc-nxp.conf*) as well as the MCUXpresso. See the examples below.

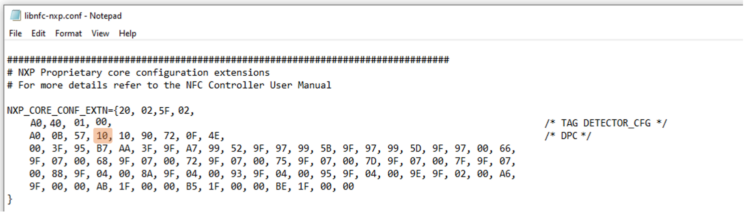


Figure 9. DPC Configuration in configuration file (libnfc-nxp.conf)

The DPC function can be enabled/disabled using the **orange marked byte**. For this example → **0x10** DPC is Disabled, **0xF0** DPC is Enabled. The dedicated NCI command is given by *PN7160_DPC_configuration_table.xlsx*, which is part of the PN7160 dynamic power control guide.

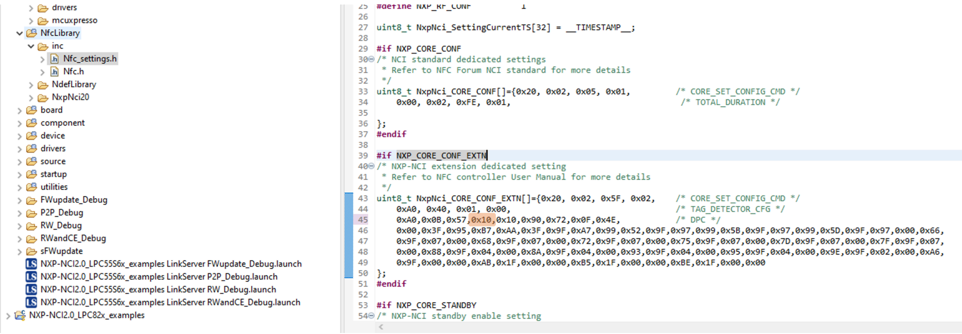


Figure 10. DPC configuration in the MCUXpresso

The DPC entries are different for a different antenna, VDD(UP) value, and antenna tuning.

The DPC Check function (Described in AN13224) is supported from the FW version **12.50.06**.

5 DPC in continuous RF on mode and PRBS mode

When entering Continuous RF on mode or PRBS mode, all interrupts are disabled. That means that DPC will not work. The DPC is supported in functional modes (polling mode, writing mode).

6 How to adjust the output power?

The target impedance of the circuit connected between TX1 and TX2 nodes defines the output power of the PN7160 IC RF transmitter.

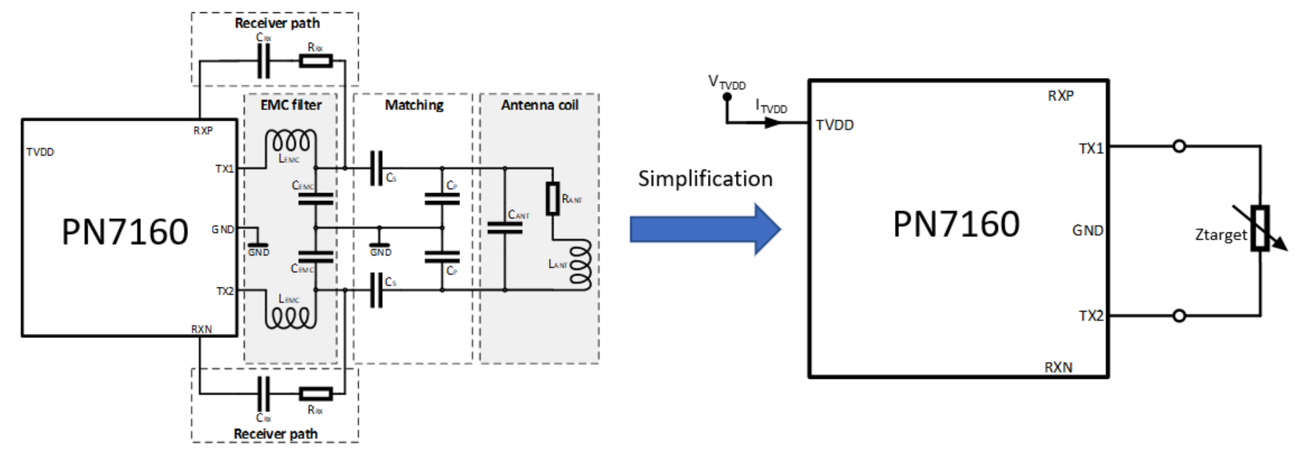


Figure 11. Output power adjustment - Simplification

The graph below shows the TX driver current I_{TVDD} versus the target impedance.

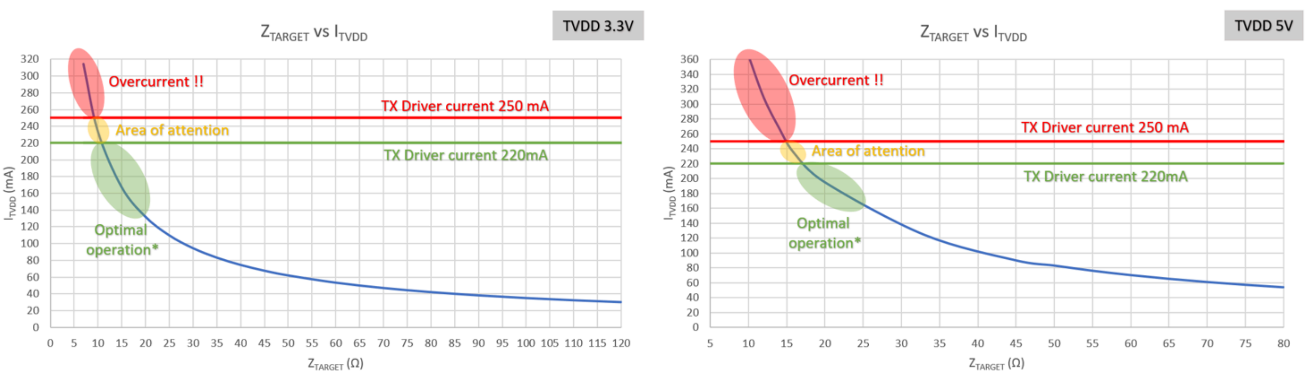


Figure 12. Ztarget versus ITVDD current

Note: For PN7160 → *TVDD=>VDD(UP)*

- The green area shows the "optimal" operation of the NFC Transmitter.
  - For symmetrical tuning, it is recommended to choose the target impedance corresponding to the TX Driver current **210 mA - 230 mA**
  - For asymmetrical tuning, the target current of the TX driver is typically **160 mA - 180 mA**

The antenna shall be tuned to never exceed the 250 mA maximum current.

I_{TVDD} is a general naming for the TX Driver current which NXP typically uses. This current can be physically measured on the VDD(UP) or VDD(TX) pins or measured using the NCI command "Antenna self-test 1 command".

7 How to support/pass ETSI and FCC tests?

The device having CE or/and FCC marking indicates that the electromagnetic radiation from the device is below the limits specified by European Telecommunications Standards Institute (ETSI) and/or Federal Communications Commission (FCC).

For the ETSI test, The EUT (**E**quipment **U**nder **T**est) e.g. an NFC reader is tested according to the applicable standards as referenced below:

Table 1. ETSI test list

EMISSION		
Description of Test Item	**Standard**	**Limits**
Conducted Disturbance at the Mains Terminal	EN 55032: 2015+A11: 2020	Class A
Conducted Common Mode Disturbance at telecommunication Port	EN 55032: 2015+A11: 2020	Class A
Radiated Disturbance	EN 55032: 2015+A11: 2020	Class A
IMMUNITY (EN 55024: 2010+A1:2015, EN 55035: 2017+A11:2020)		
Description of Test Item	**Basic Standard**	**Performance Criteria**
Radio-frequency, Radiated Immunity	IEC 61000-4-3:2006+A1:2007+A2: 2010	A
Intentional Radiator (ETSI EN 300 330: V2.1.1)		
Description of Test Item	**Basic Standard**	**Clause**
Permitted range of operating frequencies	ETSI 300 330: V2.1.1	4.3.1
Operating frequency ranges	ETSI 300 330: V2.1.1	4.3.2
Modulation bandwidth	ETSI 300 330: V2.1.1	4.3.3
Transmitter H-field requirements	ETSI 300 330: V2.1.1	4.3.4
Transmitter radiated spurious domain emission limits < 30 MHz	ETSI 300 330: V2.1.1	4.3.8
Transmitter radiated spurious domain emission limits > 30 MHz	ETSI 300 330: V2.1.1	4.3.9

For the FCC test, The EUT (**E**quipment **U**nder **T**est) e.g. an NFC Reader is tested according to the applicable standards as referenced below:

Table 2. FCC test list

Description of Test Item	Standard	Limits
Powerline Conducted Emission Measurement	47 CFR FCC Part 15 Subpart B ANSI C63.4-2014	15.107(b) Class A
Powerline Conducted Emission Measurement	47 CFR FCC Part 15 Subpart C ANSI C63.10-2013	15.207(a)
Radiated Emission Measurement (30-1000MHz)	47 CFR FCC Part 15 Subpart B ANSI C63.4 -2014	15.109(b) Class A
Radiated Emission	47 CFR FCC Part 15 Subpart C	15.209(a)

Table 2. FCC test list...*continued*

Measurement (30-1000MHz)	ANSI C63.10-2013	
Radiated Emission Measurement (Above 1 GHz)	47 CFR FCC Part 15 Subpart C ANSI C63.10-2013	15.109(b) Class A
Radiated Emission Measurement (Above 1 GHz)	47 CFR FCC Part 15 Subpart C ANSI C63.10-2013	15.209(a)
Occupied bandwidth	FCC RULES AND REG ULATIONS PART 2 AND ANSI C63.10-2013	2.1049
In-band and out band Emissions	FCC RULES AND REGULA TIONS PART 15 SUBPART C AND ANSI C63.10-2013	15.225(a)(b)(c) (d)
Frequency Tolerance	FCC RULES AND REGULA TIONS PART 15 SUBPART C AND ANSI C63.10-2013	15.225(e)

Before testing, always check the current version of the standard and limits.

NXP provides the **NFC Factory Test Application** that puts the PN7160 into the correct modes for testing. For more details, see the dedicated application note → AN13287.

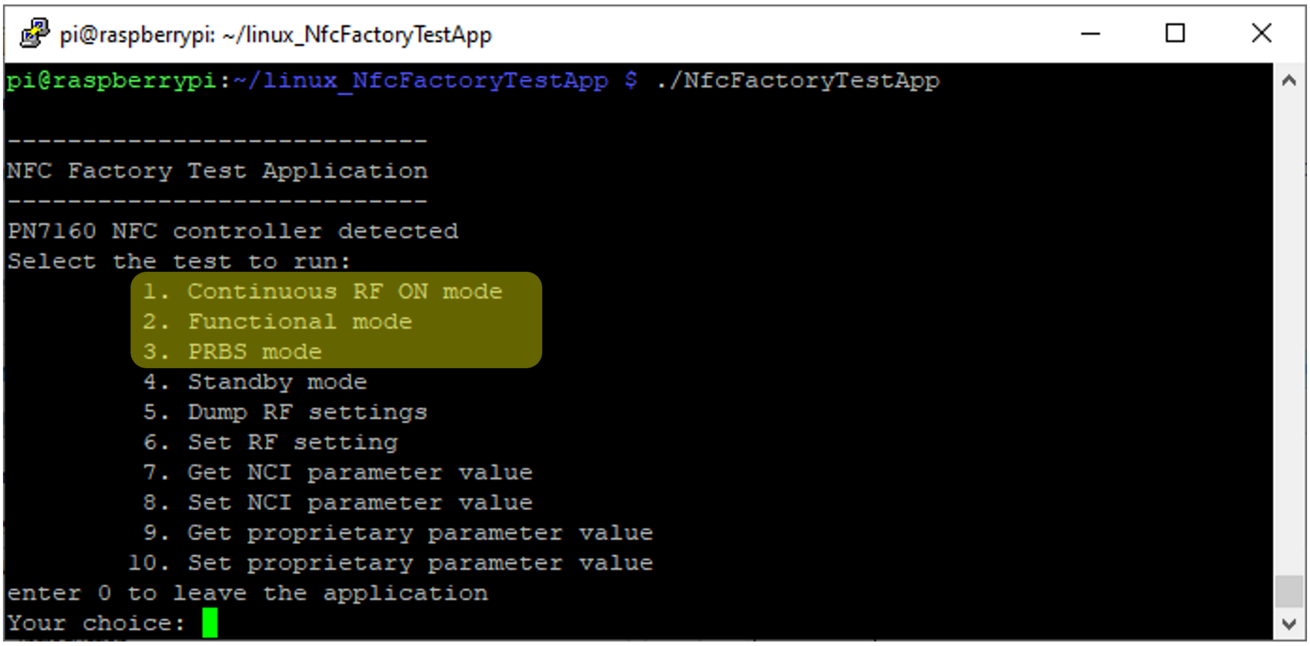


Figure 13. NFC Factory Test Application

7.1 EMI recommendations

Typically the EMI issues (overshooting of given limits) are caused by:

- Incorrect matching network layout and wrong components placement
  - EMI filter (L0 and C0) is too far from the NFCC
  - Generally long RF traces
- Antenna detuning → too high radiated power
- NFC higher harmonics are radiated by the power line.
  - The supply line has to be filtered, for example by using a ferrite filter
- Strong coupling between the NFC Reader and NFC Tag
  - On the other hand, it is recommended to match the tag and the reader antenna geometries for maximum coupling. It increases the read range but it can also impact the EMI behavior negatively (especially in a radiated emissions test). This is typically due to the higher harmonics emitted by some NFC Cards/Tags under high coupling conditions.

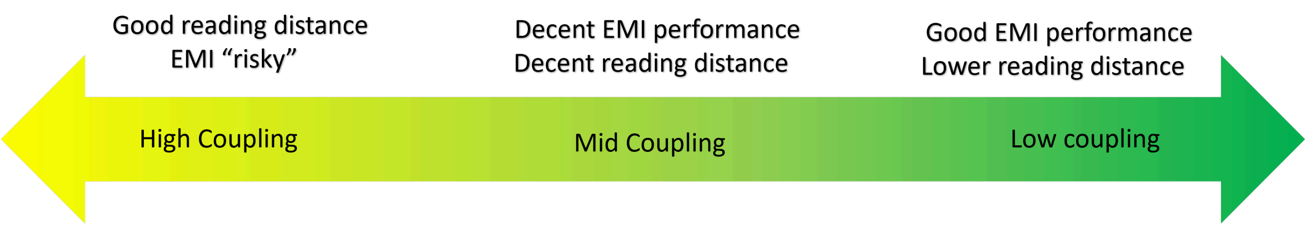


Figure 14. Coupling, EMI, and reading distance

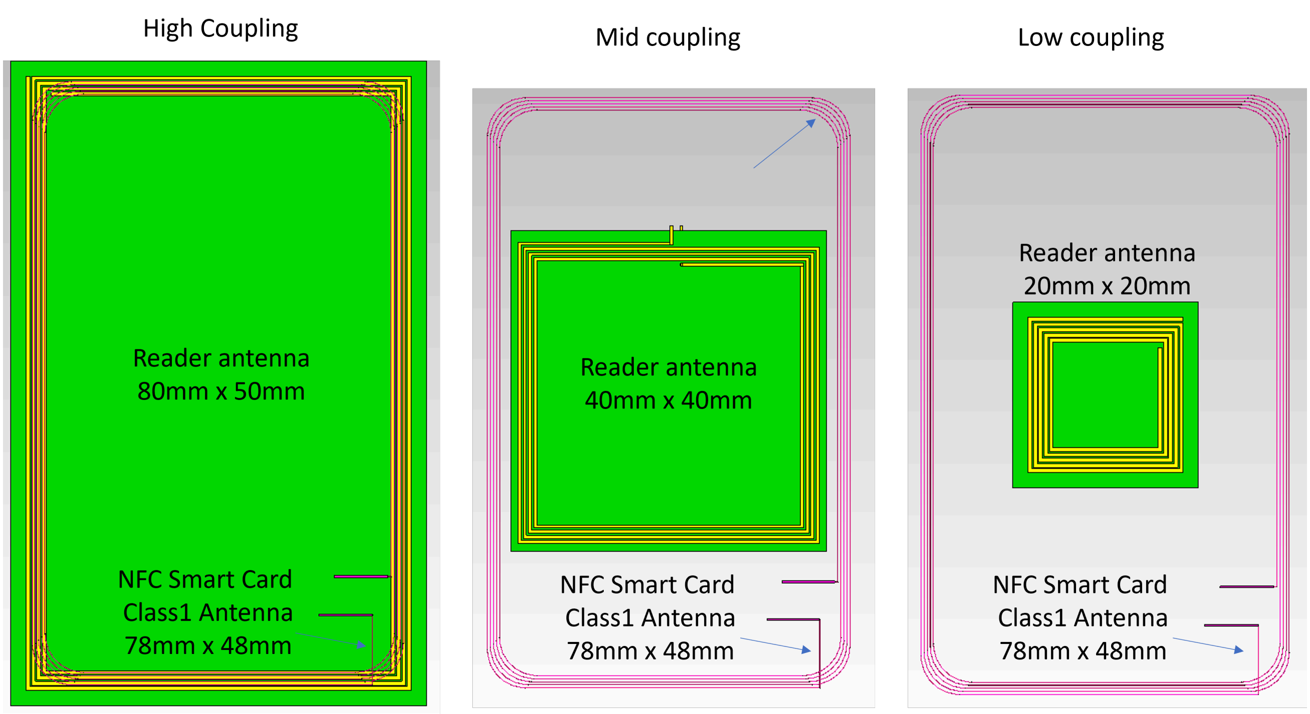


Figure 15. An example of the antenna designs, Top View.

See also general design recommendations in Figure 16.

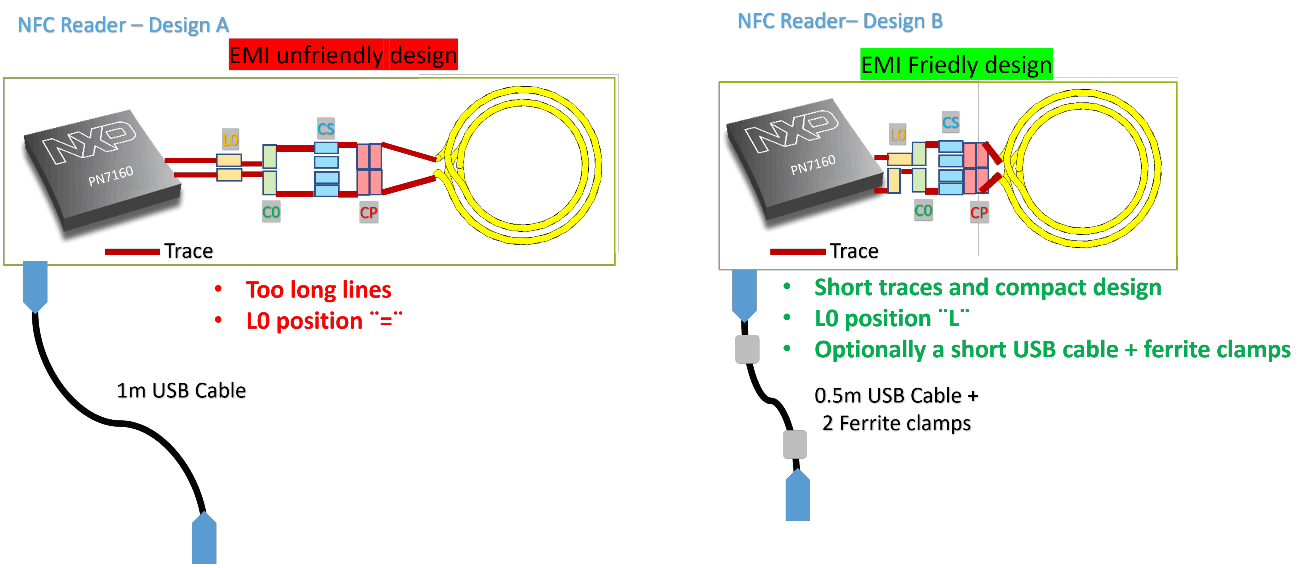


Figure 16. EMI recommendations for the PN7160 RF part

Note: More details about the EMI optimization are described in the dedicated application note → AN12988.

8 How to use an external antenna?

If you decide to go for an external antenna. There are several rules to consider. The electrical antenna parameters are changed by adding a feeding line (wires or coax cable). Typically, the inductance and resistance is getting higher. This needs to be considered during the antenna + the feeding line selection. The external antenna + the feeding line should meet requirements described in the AN13219.

Also, adding a long feeding line can reduce EMI immunity. If your design requires the feeding line between the PCB and the antenna, consider the recommendations described below.

8.1 Antenna with a twisted-pair feeding

The most cheapes/effective solution is to use wires. For EMI robustness, it is highly recommended to use a twisted pair as shown below.

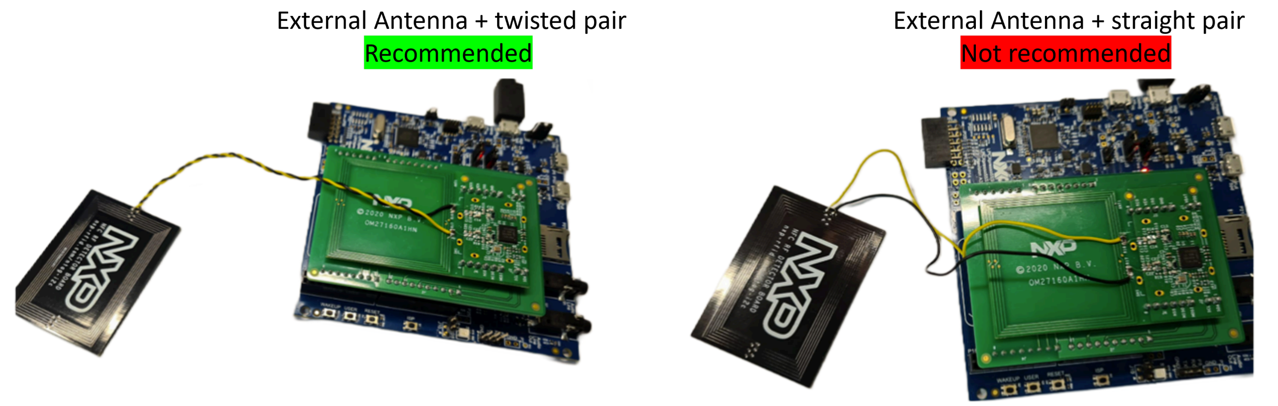


Figure 17. External antenna connected using wires

The matching procedure is the same as described in the AN13219.

The wire length must be considered during the antenna desing/selection. The length of the wires has a direct effect on overall Antenna inductance.

8.2 Antenna with a coax cable feeding

In case one wants to use a coax cable, the matching circuit requires some changes. The PN7160 has a differential (balanced) output and the coax cable is supposed to be used for unbalanced signals (single ended).

Therefore a BALUN (balanced to unbalanced) must be used. The length of the coax cable should not be more than 1.5 m

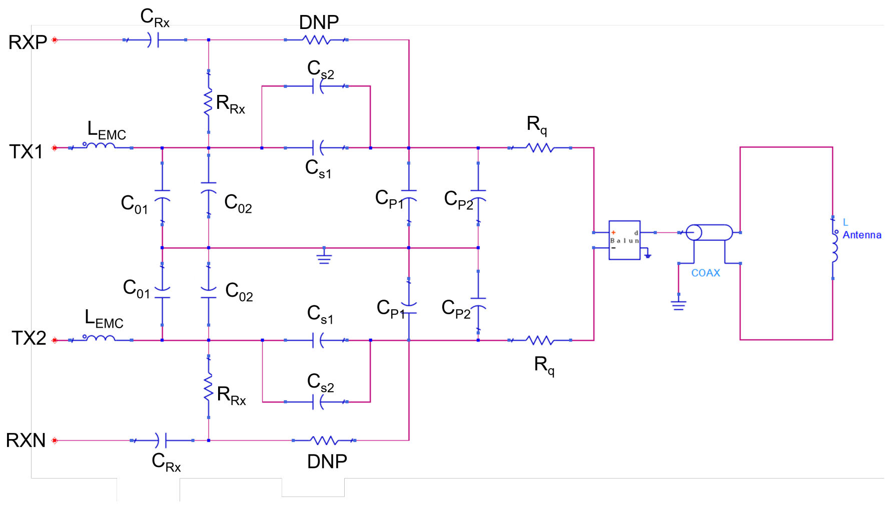


Figure 18. Recommended connection including BALUN and Coax cable

For this example, the following BALUN has been used: **DXW21BN2511NL**

See an example of the external antenna with BALUN + coax cable below.

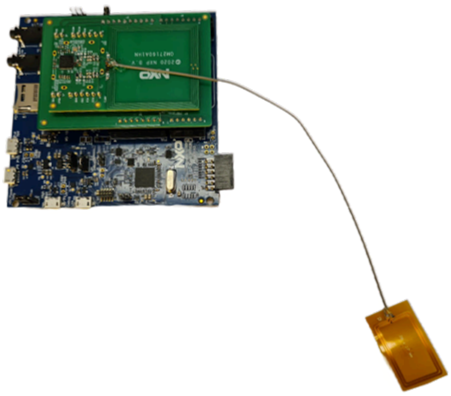


Figure 19. External antenna connected using a coax cable

8.2.1 Antenna + BALUN tuning

The matching procedure is the following.

1. Measure the antenna + coax cable + BALUN by a VNA (disconnect the rest of the tuning circuit → Figure 20)
2. Insert the measured La and Ra to the excel sheet (*PN7160_matching_calculator.xls*) which is available as an attachment of the antenna application note
3. Calculate the tuning components and assemble them
4. Measure the matching with the help of the VNA (It is already the same approach as described in the antenna application note.)
5. Adjust the matching if needed

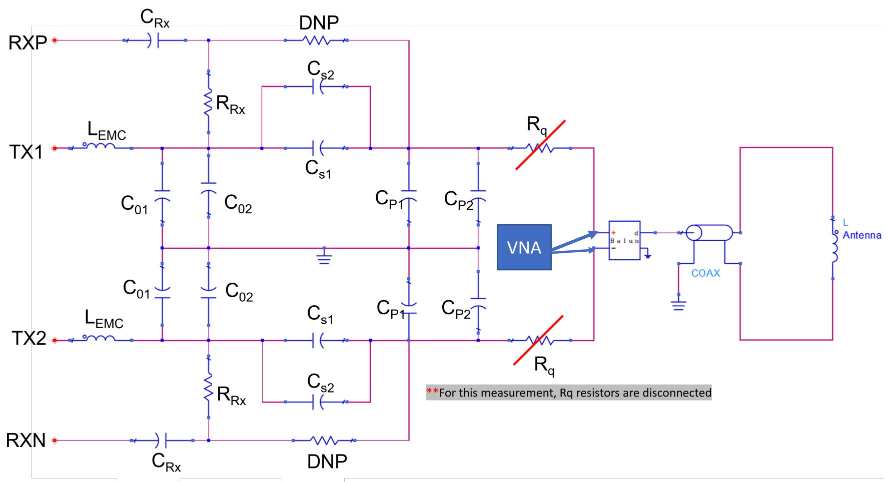


Figure 20. Antenna tuning process - BALUN

After the first step (measurement of the BALUN + coax cable + antenna), the tuning approach is the same as for standard NFC reader antenna tuning described in AN13219.

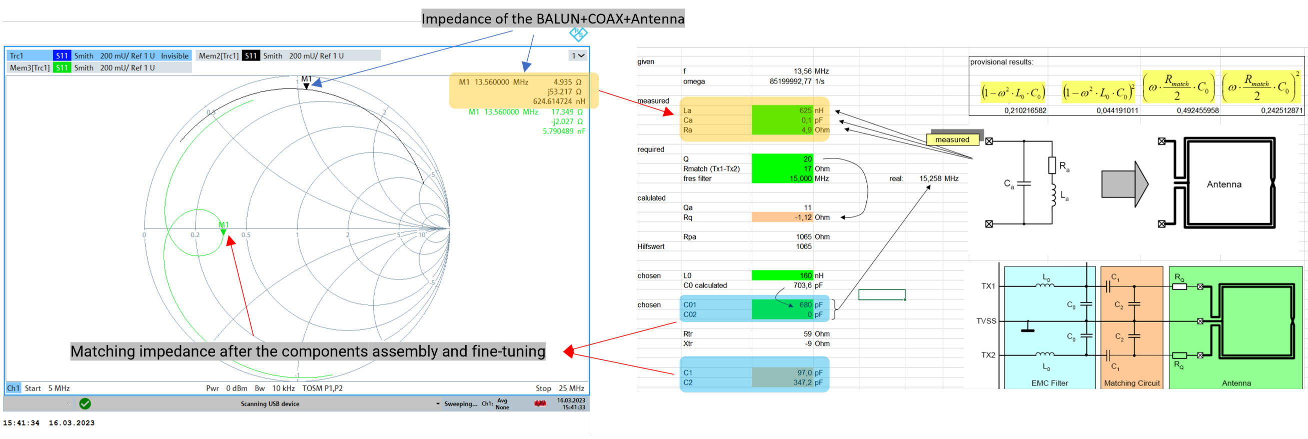


Figure 21. Antenna tuning using BALUN - Calculation and measurement

Since the BALUN and Coax cables have significant resistance, the final quality factor is typically lower than 20. Therefore, the damping resistors **0 Ω** are typically chosen.

9 TXLDO check

It could happen that the PN7160 cannot generate an RF field. This is usually due to a missing or bad power supply on VUP/TVDD or a bad clock/power configuration. The error can be indicated in logs.

```
Waiting for a Tag/Device...

2023:02:06-18:23:35.747  Read _>>>>> Empty packet recieved !!
2023:02:06-18:23:35.747  PN54X - I2C Read successful.....
2023:02:06-18:23:35.748  len =   3 > 612300
2023:02:06-18:23:35.748  PN54X - Posting read message.....
2023:02:06-18:23:35.748  read successful status = 0x0
2023:02:06-18:23:35.748  PN54X - Read requested.....
2023:02:06-18:23:35.748  PN54X - Invoking I2C Read.....
2023:02:06-18:23:36.242  Read _>>>>> Empty packet recieved !!
2023:02:06-18:23:36.242  PN54X - I2C Read successful.....
2023:02:06-18:23:36.242  len =   3 > 612300
2023:02:06-18:23:36.242  PN54X - Posting read message.....
2023:02:06-18:23:36.242  read successful status = 0x0
2023:02:06-18:23:36.243  PN54X - Read requested.....
2023:02:06-18:23:36.243  PN54X - Invoking I2C Read.....
2023:02:06-18:23:36.741  Read _>>>>> Empty packet recieved !!
2023:02:06-18:23:36.741  PN54X - I2C Read successful.....
2023:02:06-18:23:36.741  len =   3 > 612300
2023:02:06-18:23:36.741  PN54X - Posting read message.....
```

Table 112. RF_TXLDO_ERROR_NTF

GID	OID	Numbers of parameter(s)	Description
0001b	0x23	0	Notification used to indicate that TxLdo (RF Transmitter) could not start. This is usually due to a missing or bad power supply on VUP/TVDD or a bad clock/power configuration

Figure 22. TXLDO Error inside of log file

This typically happens once the 3.3 V as the VUP has been used. This issue can be solved by disabling the TXLDO Check register. The register settings are described in UM11495.

Byte 9: TXLDO check

Bit Mask								Description
b7	b6	b5	b4	b3	b2	b1	b0	VUP voltage check
X								When VUP expected being set to 5V 0 disabled, 1 enabled
	X							When VUP expecting being set to 3.6v 0 disabled, 1 enabled
		0	1	0	0	0	0	RFU

NOTE: FW automatically knows whether 5 V or 3.6 V is expected on VUP depending on other configuration bytes.

Figure 23. TXLDO settings in PMU register

10 Reader functions do not work at low temperatures

This is happening due to the high RX sensitivity. The issue can also occur at "normal" temperatures once the NFC design is noisy.

The typical indicator is that the application is returning **60 07 01 a1** (CORE_GENERIC_ERROR_NTF) during the RF Discovery loop as shown below:

```
Select the test to run:
  1. Continuous RF ON mode
  2. Functional mode
  3. PRBS mode
  4. Standby mode
  5. Dump RF settings
  6. Set RF setting
  7. Get NCI parameter value
  8. Set NCI parameter value
  9. Get proprietary parameter value
 10. Set proprietary parameter value
enter 0 to leave the application
Your choice: 2
Functional test mode, starting discovery loop ...
>> 21 03 09 04 00 01 01 01 02 01 06 01
<< 41 03 01 00
NFC Controller is now in functional mode - Press Crtl^Z to stop
<< 60 07 01 a1
<< 60 07 01 a1
<< 60 07 01 a1
<< 60 07 01 a1
<< 60 07 01 a1
<< 60 07 01 a1
<< 60 07 01 a1
```

To fix it, the lower RX gain must be selected as shown in Figure 24. For more details, please look into the PN7160 RF settings guide. A value of 44 dB is a good starting point as such.

Value range:

- High performance and sensitivity for max. reading range: 10b … 11b
  - Strongly depends on the SNR in the system
- Typical: 01b … 10b
- High robustness and stability but low reading range: 00b

Table 15. CLIF_ANA_RX_ REG register

Bit	Symbol	Description
[31:10]	Internal use	Must not be modified
[9:8]	**RX_HPCF**	**Lower Corner Frequency:** **00->45kHz,** **01->85kHz,** **10->150kHz,** **11->250kHz**
[6:4]	**RX_GAIN_Q**	**Gain Adjustment BBA:** **00->18dB,** **01->26dB,** **10->32dB,** **11->39dB** **100->44dB** **101->51dB** **110->53dB** **111->60dB**
[3]		**RFU**
[2:0]	**RX_GAIN_I**	**Gain Adjustment BBA:** **00->18dB,** **01->26dB,** **10->32dB,** **11->39dB** **100->44dB** **101->51dB** **110->53dB** **111->60dB**

Figure 24. RX Gain settings

The RX gain must be adjusted for each technology. See an example in Figure 25.

```
#if NXP_RF_CONF
/* NXP-NCI RF configuration
 * Refer to NFC controller Antenna Design and Tuning Guidelines document for more details
 */
/* Following configuration relates to performance optimization of OM27160 NFC Controller demo kit */
uint8_t NxpNci_RF_CONF[]={0x20, 0x02, 0x67, 0x0C,
  0xA0, 0x0D, 0x03, 0x78, 0x0D, 0x02,
  0xA0, 0x0D, 0x03, 0x78, 0x14, 0x02,
  0xA0, 0x0D, 0x06, 0x4C, 0x44, 0x65, 0x09, 0x00, 0x00,
  0xA0, 0x0D, 0x06, 0x4C, 0x2D, 0x05, 0x35, 0x1E, 0x01,
  0xA0, 0x0D, 0x06, 0x82, 0x4A, 0x55, 0x07, 0x00, 0x07,
  0xA0, 0x0D, 0x06, 0x44, 0x44, 0x03, 0x04, 0xC4, 0x00,
  0xA0, 0x0D, 0x06, 0x46, 0x30, 0x50, 0x00, 0x18, 0x00,
  0xA0, 0x0D, 0x06, 0x48, 0x30, 0x50, 0x00, 0x18, 0x00,
  0xA0, 0x0D, 0x06, 0x4A, 0x30, 0x50, 0x00, 0x08, 0x00,
  0xA0, 0x0D, 0x06, 0x20, 0x44, 0x66, 0x0B, 0x00, 0x00, /* CLIF_ANA_RX_REG - Technology ISO 15693  */
  0xA0, 0x0D, 0x06, 0x3C, 0x44, 0x33, 0x0A, 0x00, 0x00, /* CLIF_ANA_RX_REG - Technology Type A - 106  */
  0xA0, 0x0D, 0x06, 0x06, 0x35, 0xF4, 0x05, 0x70, 0x02  /* AGC_RM_VALUE */
};
#endif
```

Figure 25. RX Gain settings for Technology Type A

11 Card emulation does not work at close range

This is typically due to the high level of load modulation Amplitude. By Default, The PN7160 uses the **Dynamic load modulation Amplitude** mode, which is optimized for the PN7160 Development Kit (**DLMA (1)** curve). This feature ensures that for the longer distance between the PN7160 device and the reader, the LMA amplitude gets higher (See Figure 26).

Once your design uses a different antenna and tuning than PN7160 development kit. It can happen that the LMA curve look like **DLMA (2)**. Which might lead to too much LMA power -> Communication issue.

If so, the DLMA must be adjusted, but it requires some effort + using ISO Tower including a Test PCB Assembly 2. To make it more simple, one of the different **modes** can be selected and used instead of DLMA.

The PN7160 uses 4 different "modes" for active load modulation (ALM) generation.

- Mode 1 (only one TX pin generates ALM + ASK is used)
- Mode 2 (both TX pins generate ALM + ASK is used)
- Mode 3 (both TX pins generate ALM + BPSK is used)
- Dynamic load modulation Amplitude (DLMA) - **Used by default**

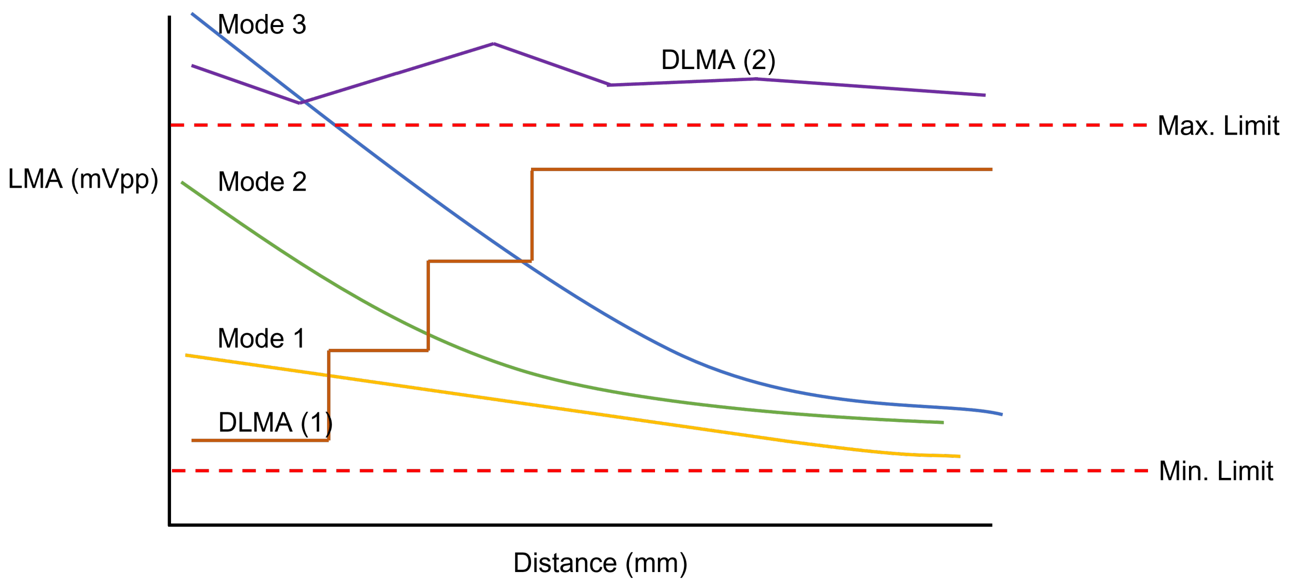


Figure 26. ALM Different modes

If your design does not work properly with default DLMA (especially in close distance). It is recommended to choose **Mode 1**.

11.1 How to switch between card emulation modes?

The DLMA mode is activated by default. This "mode" can be enabled/disabled in "*Core Config*" of PN7160. For **Mode 1** activation, the DLMA must be disabled.

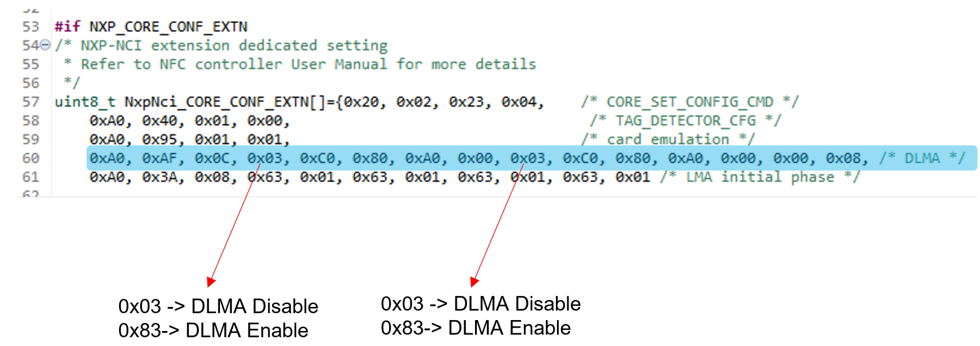


Figure 27. DLMA Disable/Enable

The NCI Command for DLMA Disable

```
0xA0, 0xAF, 0x0C, 0x03, 0xC0, 0x80, 0xA0, 0x00, 0x03, 0xC0, 0x80, 0xA0, 0x00,
 0x00, 0x08, /* DLMA- Disable */
```

The NCI Command for DLMA Enable

```
0xA0, 0xAF, 0x0C, 0x83, 0xC0, 0x80, 0xA0, 0x00, 0x83, 0xC0, 0x80, 0xA0, 0x00,
 0x00, 0x08, /* DLMA- Enable */
```

The **Mode 1** activation is done in RF configuration of PN7160.

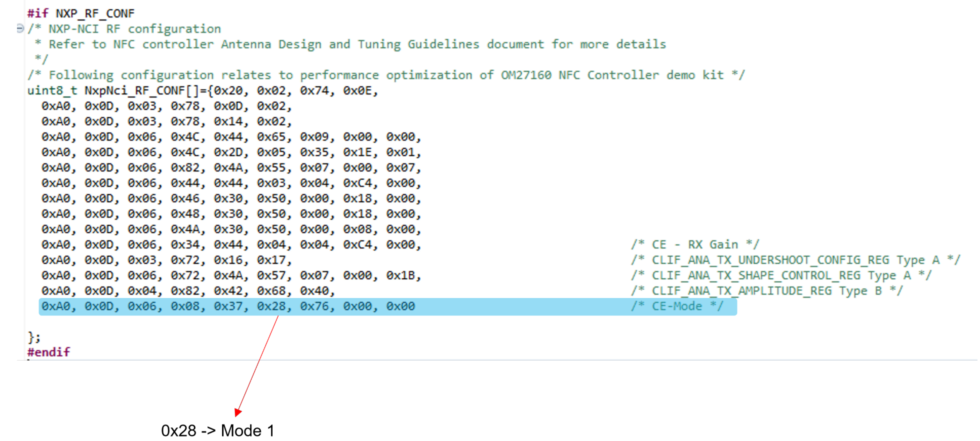


Figure 28. Card Emulation "Mode 1" activation

See the corresponding NCI command:

```
0xA0, 0x0D, 0x06, 0x08, 0x37, 0x28, 0x76, 0x00, 0x00        /* CE-Mode 1 */
```

12 PN7160 development kit - OM27160

NXP offers the OM27160 evaluation board, a flexible and easy-touse NFC controller board featuring PN7160.

It enables the development of an NFC solution based on PN7160 in a Linux or Android environment or even in system based on RTOS or without OS. It exists in 2 configurations, the only difference is then physical host interface exposed:

- OM27160A1HN featuring PN7160A1HN sample offering I2C host interface
- OM27160B1HN featuring PN7160B1HN sample offering SPI host interface

For more information follow these documents: UM11496 and AN12991.

12.1 OM27160 impedance tuning

See the RF tuning network below.

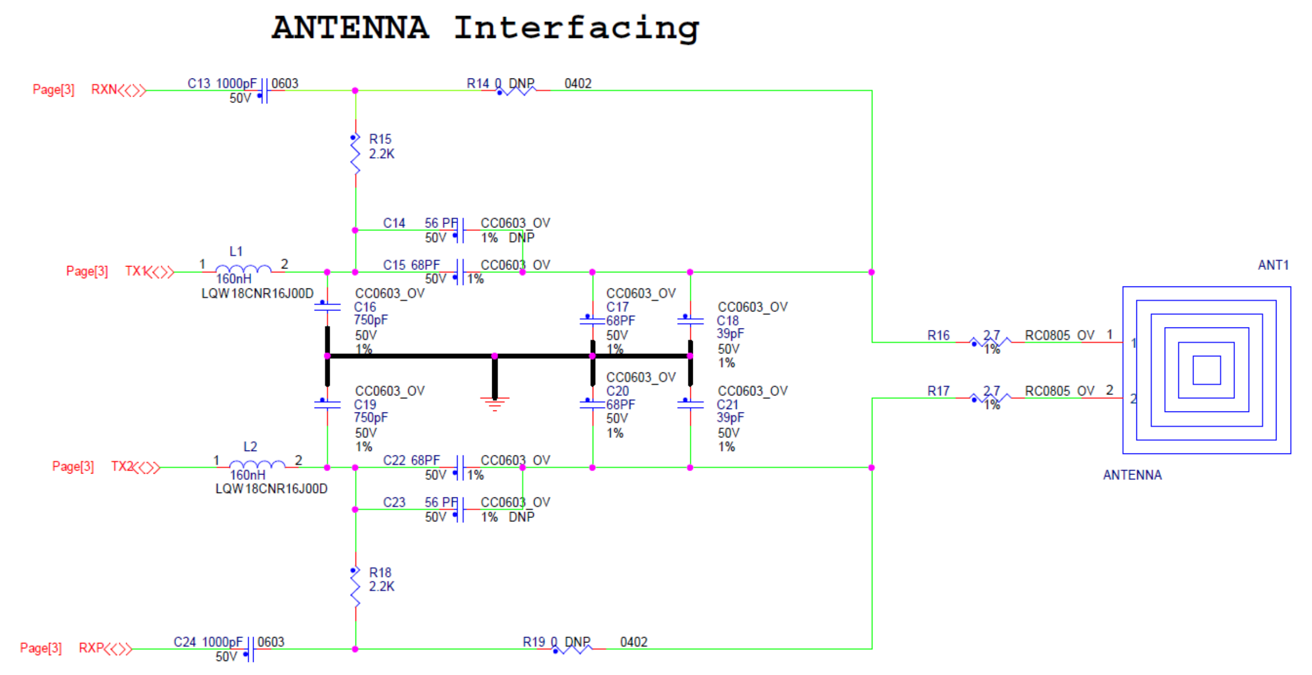


Figure 29. OM27160 - assembled components

The PN7160 Board (OM27160A1 or OM27160B1) can be used with different host devices (LPC, i.MX, K64, Raspberry Pi, Arduino...) and each affects the tuning differently. See the default tuning for **OM27160A1/B1** (Orange). The green curve shows the tuning after connecting the Raspberry Pi interface board + Raspberry Pi to the **OM27160A1/B1**.

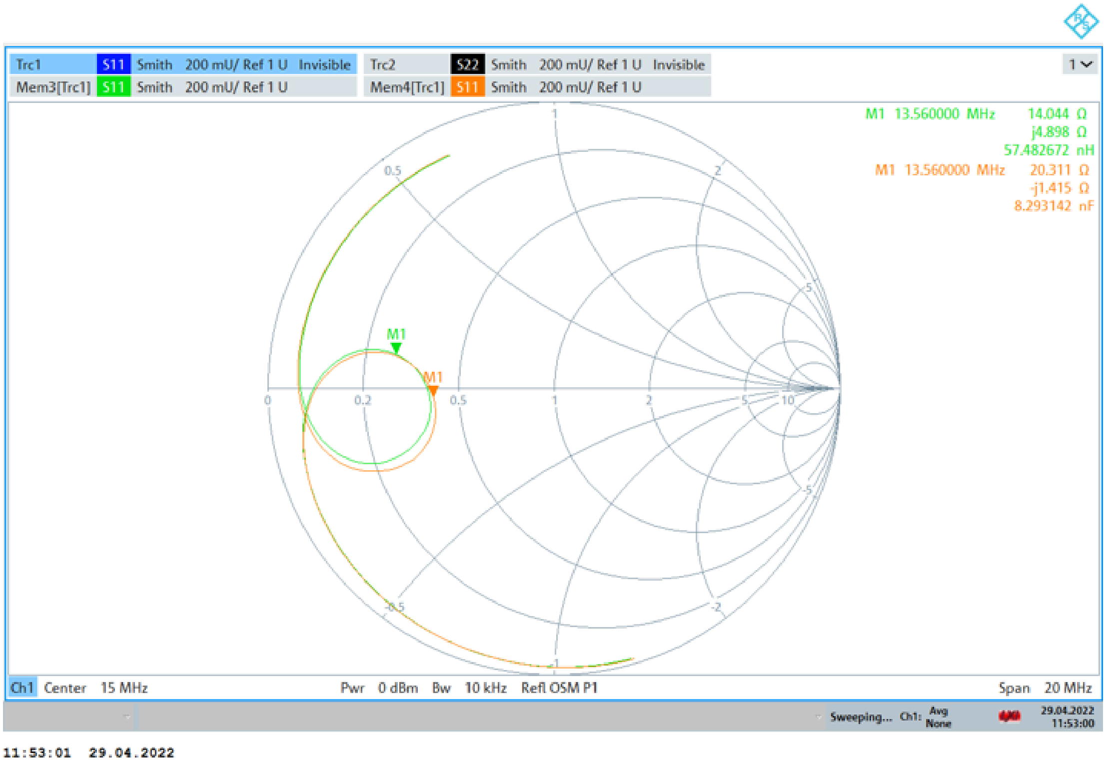


Figure 30. OM27160 - Impedance matching

The following tables show the performance of the OM27160 in reader mode and card mode.

Table 3. Reader mode distances for typical cards (VDD(TX)=4.75V)

Card type	Communication distance (mm)
NTAG 5 Link Demo Kit (Antenna 54 mm vs 27 mm)	62
NFC Sample Card (NTAG 216 – Class 1 Antenna)	77
ICODE SLIX (SL2S2002 - Class 1 Antenna)	78
NFC Sample Card (MIFARE Ultralight - Class 1 Antenna)	72
MIFARE Ultralight C (Round Antenna 35 mm)	32

Table 4. Card mode distances for typical mobile phones (VDD(TX)=4.75V)

Device	Communication distance (mm)
Samsung Galaxy S20	57
Huawei P20 Lite	52
iPhone Xs Max	50

For **VDD(TX)=3.3V**, the reading range is about 10-15% smaller

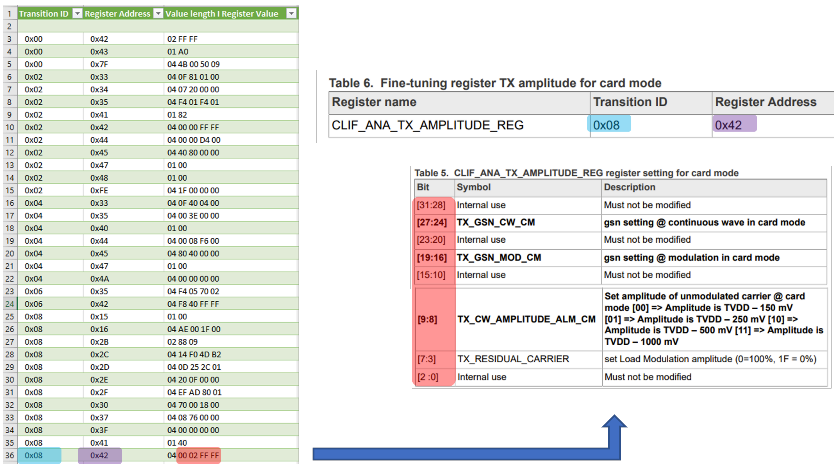
12.2 OM27160 - RF registers setting

The default OM27160 RF configuration can be loaded using the NFC Factory Test Application → Dump RF settings. This function uses the *RF_GET_TRANSITION_CMD* command, which is described in UM11495 → Section 13.3.

```
Select the test to run:
          1. Continuous RF ON mode
          2. Functional mode
          3. PRBS mode
          4. Standby mode
          5. Dump RF settings
          6. Set RF setting
          7. Get NCI parameter value
          8. Set NCI parameter value
          9. Get proprietary parameter value
         10. Set proprietary parameter value
         11. Get current value
         12. DPC Check
enter 0 to leave the application
Your choice: 5
Dumping all RF settings:
 transition 0x00, register 0x42 = 02 FF FF
 transition 0x00, register 0x43 = 01 A0
 transition 0x00, register 0x7F = 04 4B 00 50 09
 transition 0x02, register 0x33 = 04 0F 81 01 00
 transition 0x02, register 0x34 = 04 07 20 00 00
 transition 0x02, register 0x35 = 04 F4 01 F4 01
 transition 0x02, register 0x41 = 01 82
 transition 0x02, register 0x42 = 04 00 00 FF FF
```

Figure 31. How to check the RF settings in the NFC Factory Test Application

Received data can be copied and pasted to the Excel or different tool. Then the content can be decoded with the help of the PN7160 RF settings guide application note → AN13218. See an example:



1	Transition ID	Register Address	Value length I Register Value
2			
3	0x00	0x42	02 FF FF
4	0x00	0x43	01 A0
5	0x00	0x7F	04 4B 00 50 09
6	0x02	0x33	04 0F 81 01 00
7	0x02	0x34	04 07 20 00 00
8	0x02	0x35	04 F4 01 F4 01
9	0x02	0x41	01 82
10	0x02	0x42	04 00 00 FF FF
11	0x02	0x44	04 00 00 D4 00
12	0x02	0x45	04 40 80 00 00
13	0x02	0x47	01 00
14	0x02	0x48	01 00
15	0x02	0xFE	04 1F 00 00 00
16	0x04	0x33	04 0F 40 04 00
17	0x04	0x35	04 00 3E 00 00
18	0x04	0x40	01 00
19	0x04	0x44	04 00 08 F6 00
20	0x04	0x45	04 80 40 00 00
21	0x04	0x47	01 00
22	0x04	0x4A	04 00 00 00 00
23	0x06	0x35	04 F4 05 70 02
24	0x06	0x42	04 F8 40 FF FF
25	0x08	0x15	01 00
26	0x08	0x16	04 AE 00 1F 00
27	0x08	0x2B	02 88 09
28	0x08	0x2C	04 14 F0 4D B2
29	0x08	0x2D	04 0D 25 2C 01
30	0x08	0x2E	04 20 0F 00 00
31	0x08	0x2F	04 EF AD 80 01
32	0x08	0x30	04 70 00 18 00
33	0x08	0x37	04 08 76 00 00
34	0x08	0x3F	04 00 00 00 00
35	0x08	0x41	01 40
36	0x08	0x42	04 00 02 FF FF

Table 6. Fine-tuning register TX amplitude for card mode

Register name	Transition ID	Register Address
CLIF_ANA_TX_AMPLITUDE_REG	0x08	0x42

Table 5. CLIF_ANA_TX_AMPLITUDE_REG register setting for card mode

Bit	Symbol	Description
[31:28]	Internal use	Must not be modified
[27:24]	**TX_GSN_CW_CM**	**gsn setting @ continuous wave in card mode**
[23:20]	Internal use	Must not be modified
[19:16]	**TX_GSN_MOD_CM**	**gsn setting @ modulation in card mode**
[15:10]	Internal use	Must not be modified
[9:8]	**TX_CW_AMPLITUDE_ALM_CM**	**Set amplitude of unmodulated carrier @ card mode [00] => Amplitude is TVDD – 150 mV [01] => Amplitude is TVDD – 250 mV [10] => Amplitude is TVDD – 500 mV [11] => Amplitude is TVDD – 1000 mV**
[7:3]	TX_RESIDUAL_CARRIER	set Load Modulation amplitude (0=100%, 1F = 0%)
[2 :0]	Internal use	Must not be modified

Figure 32. The OM27160 RF settings

The RF parameters can be changed using the NFC Factory Test Application as shown bellow.

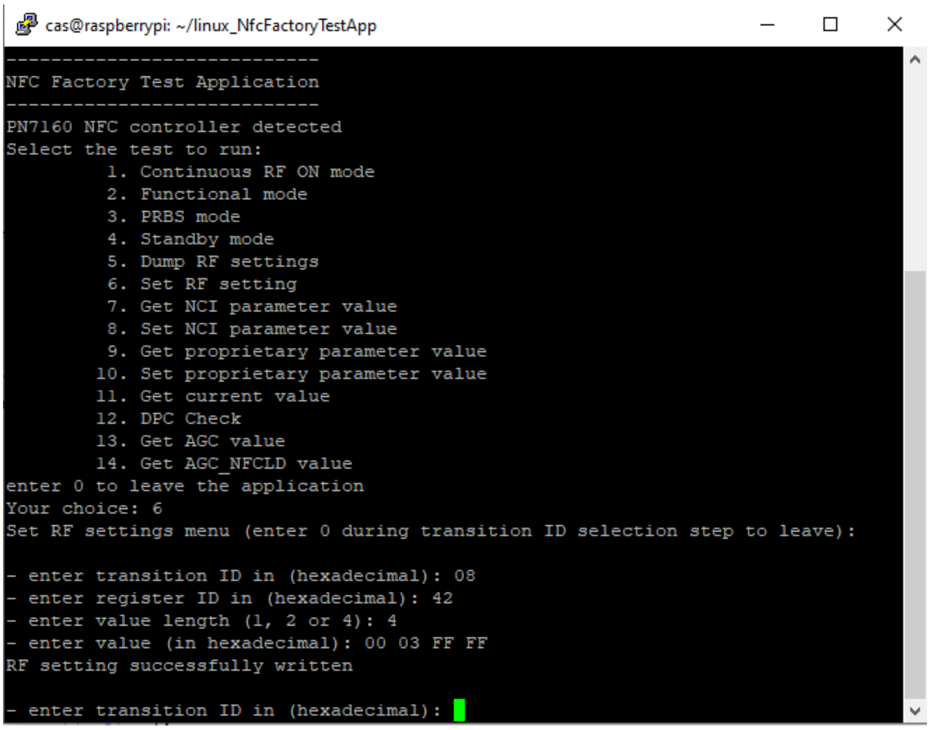


Figure 33. How to set RF settings in the NFC Factory Test Application

Note: *The byte order for the register value is defined as little-endian, meaning LSByte written first (LSB to MSB).*

See an example of how to set an RF parameter correctly.

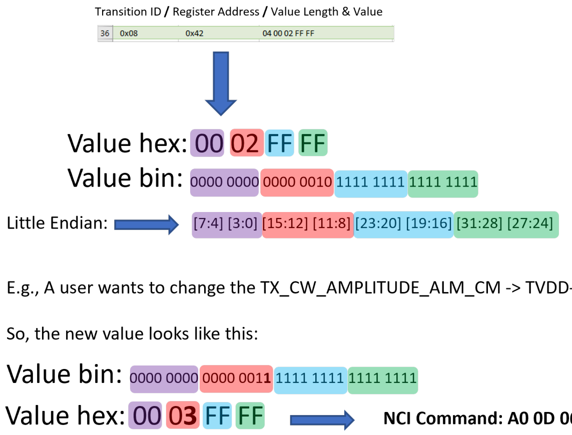


Table 6. Fine-tuning register TX amplitude for card mode

Register name	Transition ID	Register Address
CLIF_ANA_TX_AMPLITUDE_REG	0x08	0x42

Table 5. CLIF_ANA_TX_AMPLITUDE_REG register setting for card mode

Bit	Symbol	Description
[31:28]	Internal use	Must not be modified
[27:24]	**TX_GSN_CW_CM**	**gsn setting @ continuous wave in card mode**
[23:20]	Internal use	Must not be modified
[19:16]	**TX_GSN_MOD_CM**	**gsn setting @ modulation in card mode**
[15:10]	Internal use	Must not be modified
[9:8]	**TX_CW_AMPLITUDE_ALM_CM**	**Set amplitude of unmodulated carrier @ card mode [00] => Amplitude is TVDD – 150 mV [01] => Amplitude is TVDD – 250 mV [10] => Amplitude is TVDD – 500 mV [11] => Amplitude is TVDD – 1000 mV**
[7:3]	TX_RESIDUAL_CARRIER	set Load Modulation amplitude (0=100%, 1F = 0%)
[2 :0]	Internal use	Must not be modified

Figure 34. How to deal with setting RF parameters using little-endian

The full list with default RF settings optimized for OM27160 is available in AN13218.

NXP offers a GUI tool, which may help to create a dedicated NCI command. The tool is available on NXP webpage.

13 How to get ECP (PN7161) support?

To get ECP support, contact local Field Application Engineer.

ECP is supported in the PN7161. PN7161 supports same features as PN7160.

How to get access to documentation and SW commands:

1. For other applications, engage with your Apple Waller representative
2. Get approval from Apple for using PN7161 in the application
3. Sign an NXP NDA for Secure Files access → See the guidance here
4. Create your Secure Files account
5. Request the desired documents on Secure Files

To check if the IC is PN7160 or PN7161 user need to check CORE_RESET_NTF:

```
>> 20 00 01 00                              // CORE_RESET_CMD
<< 40 00 01 00                              // CORE_RESET_RSP
<< 60 00 09 02 00 20 04 04 71 12 50 09 // CORE_RESET_NTF
```

9th bytes of the CORE_RESET_NTF indicates the model ID (0x71 in above example).

- model ID value 0x**6**1 indicates PN716**0**
- model ID value 0x**7**1 indicates PN716**1**

14 PN7150 vs. PN7160

Both products are close controllers but with different characteristics:

Table 5. RF performances

	PN7150	PN7160
Transmitter voltage	2.7 V to 4.75 V	2.7 V to 5.25 V
Max transmitter current	180 mA	250 mA
Output power	0.8 W	1.3 W
Receiver sensitivity	150mVpp	20mVpp

Table 6. Host interface

	PN7150	PN7160
NCI specification	V1.0	V2.0
Host interface	I2C	I2C or SPI

Table 7. Packages

	PN7150	PN7160
Package	HVQFN40	HVQFN40 and VFBGA64

New features in PN7160:

- Autonomous NDEF emulation
- Dynamic power control
- Dynamic load modulation amplitude
- Firmware update

PN7150 and PN7160 are not pin-to-pin compatible.

Table 8. PN7150 supported Android versions

Android version
Android R (Android 11) (last one supported)
Android Q (Android 10)
Android Pie (Android 9)
Android Oreo (Android 8)
Android Nougat (Android 7)
Android Marshmallow (Android 6)
Android Lollipop (Android 5)
Android KitKat (Android 4)

Table 9. PN7160 supported Android versions

Android version
Android 13
Android 12
Android 11

Be aware, that for other integrations (e.g. PN7160 to Android 10), NXP does not provide any support.

15 Configuration files

PN7160 is controlled via configuration files (attached to every Android release and Linux release).

Packages always contain two configuration files:

1. libnfc-nci.conf
2. libnfc-nxp.conf

Explanation of flags can be found in Linux porting guide and Android porting guide.

When changes are performed:

1. In Android, push files with:
   - `adb push libnfc-nxp.conf /vendor/etc/`
   - `adb push libnfc-nci.conf /system/etc/`
   - `adb reboot`
2. In Linux environment:
   - `make`
   - `sudo make install`

16 PN7160 reference design

PN7160 variants:

Table 10. PN7160/61 configurations

Part number	Control interface	Package
PN7160**A1**EV/C100	I2C	VFBGA64
PN7160**A1**HN/C100	I2C	HVQFN40
PN7160**B1**EV/C100	SPI	VFBGA64
PN7160**B1**HN/C100	SPI	HVQFN40
PN7161**A1**EV/C100	I2C	VFBGA64
PN7161**A1**HN/C100	I2C	HVQFN40
PN7161**B1**EV/C100	SPI	VFBGA64
PN7161**B1**HN/C100	SPI	HVQFN40

A1 = I2C control interface

B1 = SPI control interface

NXP provides three different reference designs for PN7160:

1. Android
   - Reference design is done with Hikey960 as Device Host
   - Patches can be found here: Android patches
   - Android porting guide: Android porting guide
2. Linux
   - Reference design is done with Raspberry pi as Device Host
   - Code base can be found here: Linux code base
   - Linux porting guide: Linux porting guide
3. MCUXpresso (BareMetal)
   - Reference design is done with LPC555S6x, LPC82x, and i.MXRT1170
   - Project can be found here: MCUXpresso examples
   - MCUXpresso examples guide: MCUXpresso examples guide

Refer to the application notes for specific device host, to understand how to switch between control interfaces.

17 PN7160 MIFARE examples

PN7160 MIFARE examples for MCUXpresso project and Linux can be found on PN7160 web page under Software and "Secure files". Follow the instructions on the website to get access.

18 How can I disable PN7160 Tag Detector mode?

Tag detector functionality can be enabled/disabled and configured through dedicated PN7161 parameter @0xA040 set and get using NCI standardized CORE_SET_CONFIG and CORE_GET_CONFIG commands.

- In Android or Linux, this is done updating *libnfc-nxp.conf* file:

Table 11. Android/Linux configuration of Tag Detector mode

```
NXP_CORE_CONF_EXTN={20, 02, 11, 04,     //NCI CORE_SET_CONFIG_CMD updated about length and number of parameters
    A0, 5E, 01, 01,
    A0, 40, 01, 00,                     //Tag Detector function disabled (00) or enabled (01)
}
```

- In MCUXpresso example code, this is done updating *Nfc_setting.h* file:

Table 12. MCUXpresso example configuration of discovery loop frequency

```
uint8_t NxpNci_CORE_CONF_EXTN[]={0x20, 0x02, 0x05, 0x01,     /* CORE_SET_CONFIG_CMD */
    0xA0, 0x40,                                              /* TAG_DETECTOR_CFG parameter ID */
    0x01,                                                    /* TAG_DETECTOR_CFG parameter size */
    0x00                                                     /* TAG_DETECTOR_CFG parameter value */
};
```

19 How can I set the discovery loop frequency?

Discovery loop frequency depends on the NCI standard TOTAL_DURATION parameter set and get using NCI standardized CORE_SET_CONFIG and CORE_GET_CONFIG commands. PN7160 default value is 1 Hz (POLL mode phase occurs once a time per second).

- In Android or Linux, the stack defines a default value of 500 ms. However, it can be updated setting the NFA_DM_DISC_DURATION_POLL parameter in *libnfc-nci.conf* file:

Table 13. Android/Linux configuration of discovery loop frequency

`NFA_DM_DISC_DURATION_POLL=256`	TOTAL_DURATION value in milliseconds (decimal)

- In MCUXpresso example code, this is done updating *Nfc_settings.h* file:

Table 14. NXP-NCI example configuration of discovery loop frequency

```
uint8_t NxpNci_CORE_CONF[]={0x20, 0x02, 0x05, 0x01,     /* CORE_SET_CONFIG_CMD */
    0x00,                                               /* TOTAL_DURATION parameter ID */
    0x02,                                               /* TOTAL_DURATION parameter size */
    0x00, 0x01                                          /* TOTAL_DURATION value is ms (LSB first) */
};
```

20 NCI specification

To get the NCI specification, follow the steps on NFC Forum website (NCI Specification).

To get differences between specific version of NCI specification, check "Revision History" section in NCI specification.

21 Troubleshooting

21.1 How to enable logging?

In MCUXpresso:

Enabling NCI communication traces can be done defining NCI_DEBUG compile flag inside the project properties (see Figure 35), or directly in NfcLibrary/NxpNci20/inc/ NxpNci.h file, before building the project. Pay attention that this significantly increases overall memory requirement.

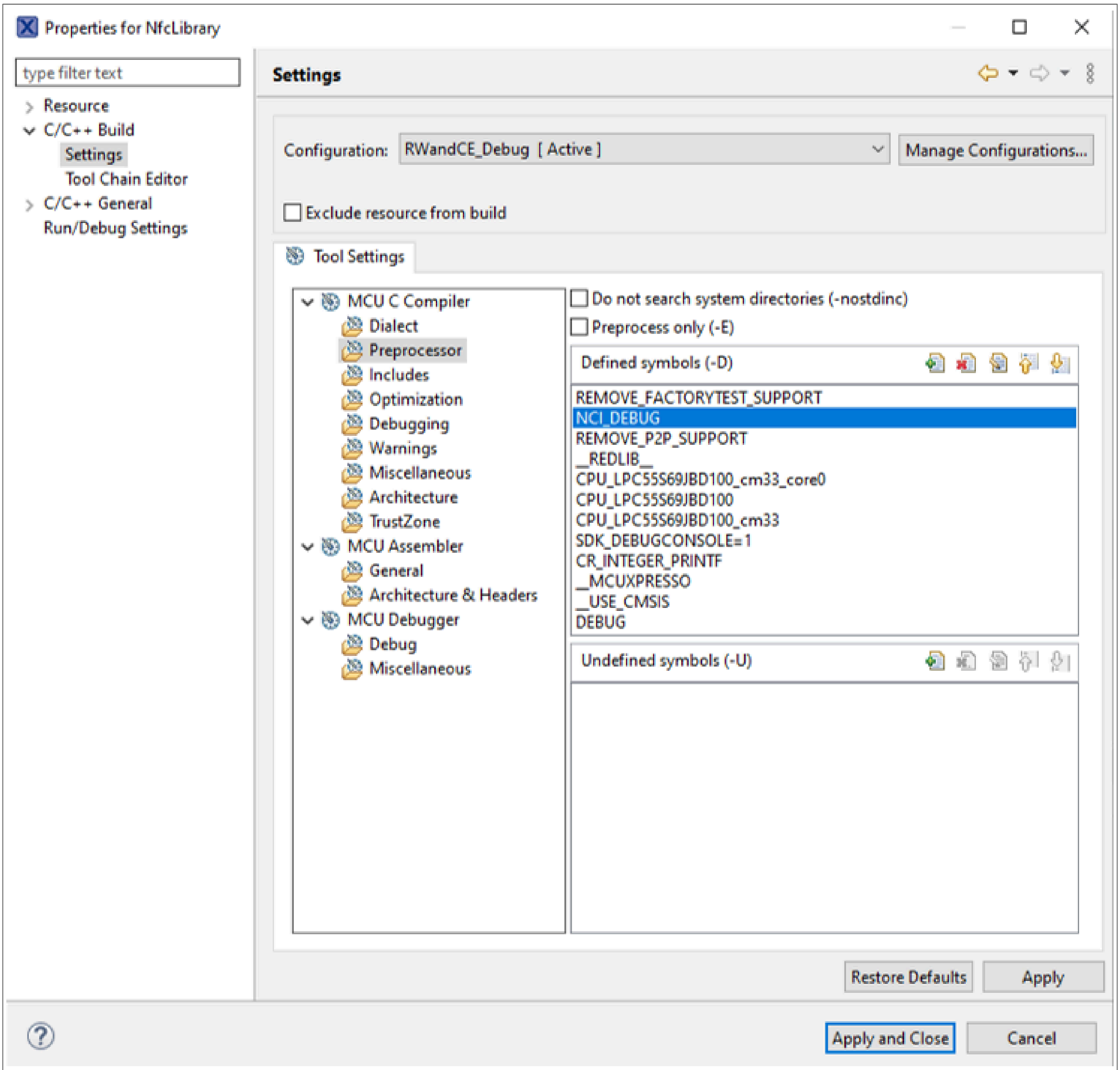


Figure 35. MCUXpresso project preprocessor definition

Linux and Android:

Inside libnfc-nxp.conf, change value to the following flags:

- NXPLOG_EXTNS_LOGLEVEL
- NXPLOG_NCIHAL_LOGLEVEL
- NXPLOG_NCIX_LOGLEVEL
- NXPLOG_NCIR_LOGLEVEL
- NXPLOG_FWDNLD_LOGLEVEL
- NXPLOG_TML_LOGLEVEL

Logging levels:

- NXPLOG_DEFAULT_LOGLEVEL 0x01
- NXPLOG_DEBUG_LOGLEVEL 0x03
- NXPLOG_WARN_LOGLEVEL 0x02
- NXPLOG_ERROR_LOGLEVEL 0x01
- NXPLOG_SILENT_LOGLEVEL 0x00

21.2 Error messages

Error messages coming from PN7160 are encapsulated into NCI messages. When translating those NCI messages, users can see if any error occurred. To find out twhat this error represents, refer to the NCI specification or the user manual (User manual).

22 Card Emulation support

PN7160 supports Type A and Type B Card emulation. Both types are supported also in both scenarios of CE:

1. Card emulation by the DH-NFCEE
2. Card emulation over NFCC

Explanation of scenarios is provided in User manual and PN7160 Card emulation.

23 How can user set the discovery loop sequence?

PN7160 discovery loop sequence must be set to define which technologies should be polled for. In Android or Linux, this is done updating *libnfc-nxp.conf file*:

```
###############################################################
# Force tag polling for the following technology(s).
# The bits are defined as tNFA_TECHNOLOGY_MASK in nfa_api.h
#
# Notable bits:
# NFA_TECHNOLOGY_MASK_A 0x01 /* NFC-A */
# NFA_TECHNOLOGY_MASK_B 0x02 /* NFC-B */
# NFA_TECHNOLOGY_MASK_F 0x04 /* NFC-F */
# NFA_TECHNOLOGY_MASK_ISO15693 0x08 /* Proprietary */
# NFA_TECHNOLOGY_MASK_KOVIO 0x20 /* Proprietary */
# NFA_TECHNOLOGY_MASK_A_ACTIVE 0x40 /* NFC-A active mode */
# NFA_TECHNOLOGY_MASK_F_ACTIVE 0x80 /* NFC-F active mode */
# This flag when set to zero will disable Reader mode
POLLING_TECH_MASK=0xEF
```

POLLING_TECH_MASK defines enabled polling technologies in the discovery polling loop sequence as bit field.

In MCUXpresso example code, this is done updating "unsigned char DiscoveryTechnologies":

```
unsigned char DiscoveryTechnologies[] = {
  MODE_POLL | TECH_PASSIVE_NFCA,
  MODE_POLL | TECH_PASSIVE_NFCB,
  MODE_POLL | TECH_PASSIVE_NFCF,
  MODE_POLL | TECH_PASSIVE_15693,
  MODE_POLL | TECH_ACTIVE_NFCA,
  MODE_POLL | TECH_ACTIVE_NFCF,
};
```

24 FW update

Android:

To update FW on Android, follow the instructions below: (adb tools must be installed on the system.)

1. Download the FW from GitHub: PN7160 FW
2. Open terminal at the FW location
3. Run the the following commands:
   - `adb push libpn7160_fw.so vendor/lib64/libpn7160_fw.so` (for 64-bit version) and `adb push libpn7160_fw.so vendor/lib/libpn7160_fw.so` (for 32-bit version)
   - `adb shell svc nfc disable`
   - `adb shell svc nfc enable`

MCUXpresso examples:

Follow the instructions in the application note: PN7160 - MCUXpresso examples guide.

Linux:

When getting source code for Linux (Linux code base) change phDnldNfc_UpdateSeq.c inside / firmware/pn7160/ folder. Inside phDnldNfc_UpdateSeq_Old.c, there is FW version 12.50.05. Inside phDnldNfc_UpdateSeq.c, there is FW version 12.50.09. If we want to switch between versions, simply copy content from phDnldNfc_UpdateSeq_Old.c to phDnldNfc_UpdateSeq.c.

24.1 How to check the current FW version

There are multiple ways how to check the current FW version. For example, it can be done using *nfcDemoApp*. The NCI communication traces must be enabled as shown in Figure 36.

```
###############################################################################
# Logging Levels. Suggested value for debugging is 0x03.
#  NXPLOG_GLOBAL_LOGLEVEL - Configuration for Global logging level
#  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 functionali
#  NXPLOG_TML_LOGLEVEL - Configuration for enabling logging of TML
# Logging Levels
#  NXPLOG_DEFAULT_LOGLEVEL    0x01
#  NXPLOG_DEBUG_LOGLEVEL      0x03
#  NXPLOG_WARN_LOGLEVEL       0x02
#  NXPLOG_ERROR_LOGLEVEL      0x01
#  NXPLOG_SILENT_LOGLEVEL     0x00
NXPLOG_EXTNS_LOGLEVEL=0x03
NXPLOG_NCIHAL_LOGLEVEL=0x03
NXPLOG_NCIX_LOGLEVEL=0x03
NXPLOG_NCIR_LOGLEVEL=0x03
NXPLOG_FWDNLD_LOGLEVEL=0x03
NXPLOG_TML_LOGLEVEL=0x03
###############################################################################
```

Figure 36. NCI communication traces enabled in libnfc-nxp.conf

Then the *nfcDemoApp poll* can be started and the FW version can be found in the logs as shown below. In the example below, the FW version is **12.50.09.**

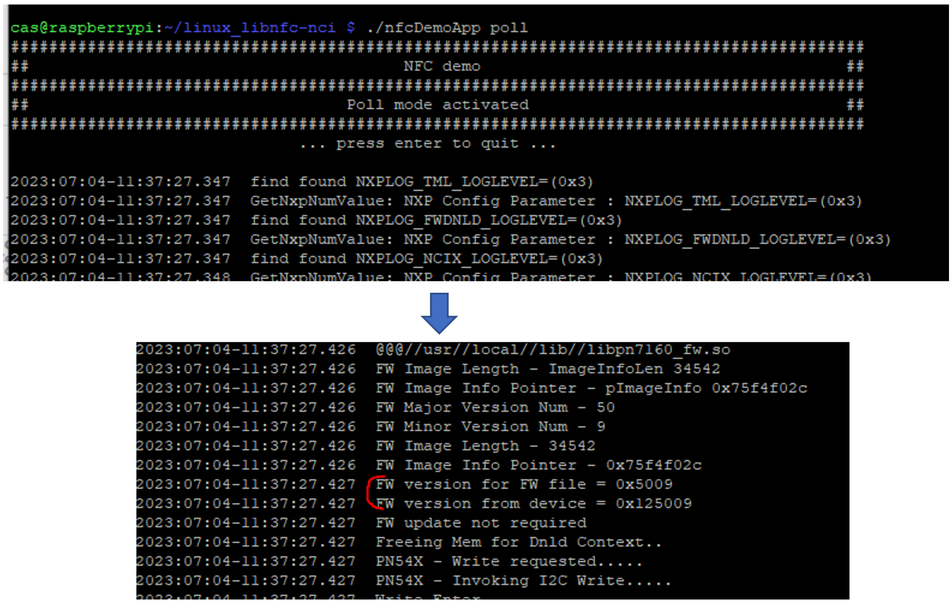


Figure 37. PN7160 FW version check

Another possibility is also to check CORE_RESET_NTF. Check following example:

```
>> 20 00 01 00                             // CORE_RESET_CMD
<< 40 00 01 00                             // CORE_RESET_RSP
<< 60 00 09 02 00 20 04 04 71 12 50 09 // CORE_RESET_NTF
```

10th, 11th and 12th bytes od the CORE_RESET_NTF indicates the FW version (0x12, 0x50, 0x09 point to FW version 12.50.09).

25 Android build errors

Libnfc-nci.conf previously defined at... error:

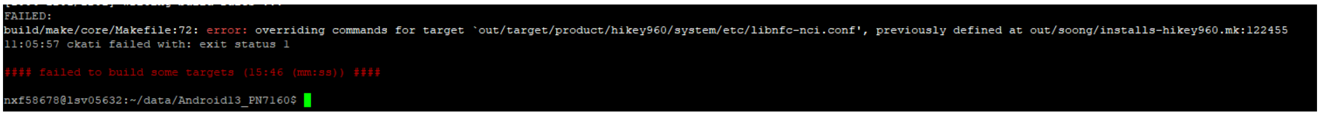


Figure 38. libnfc-nci.conf previously defined here error

If observed error like in picture (instead of hikey960 some other device), just comment out line number listed in error and run build command. In our example this is out/soong/installs-hikey960.mk:122455.

26 Abbreviations

Table 15. Abbreviations

Acronym	Description
ALM	active load emiulation
ASK	amplitude-shift keying
BPSK	binary phase-shift keying
CE	Conformité Européenne - European conformity
CE	card emulation
CFG1	Configuration 1
CFG2	Configuration 2
DC-DC	direct current to direct current
DH	device host
DPC	dynamic power control
EMC	electromagnetic compatibility
EMI	electromagnetic interference
ETSI	European Telecommunications Standards Institute
EUT	equipment under test
FCC	Federal Communications Commission
mA	milliampere
MHz	megahertz
NCI	NFC controller interface
NFC	near-field communication
NFCC	near-field communication controller (e.g. PN7160)
NFCEE	NFC execution environment
PCB	printed-circuit board
PMU	power management unit
RF	radio frequency
Q	Quality factor
TXLDO	transmitter low-dropout regulator
V	voltage
VNA	vector network analyzer

27 References

[1] Webpage – PN7160: NFC Plug and Play Controller with Integrated Firmware and NCI Interface (link)
[2] Application note – AN12988 – PN7160 hardware design guide (link)
[3] User manual – UM11495 – PN7160 NFC controller (link)
[4] Application note – AN13219 – PN7160 antenna design and matching guide (link)
[5] Application note – AN13287 – PN7160 Linux porting guide (link)
[6] Application note – AN13224 – PN7160 dynamic power control guide (link)
[7] Application note – AN13287 – PN7160 Android porting guide (link)
[8] Application note – AN13288 – MCUXpresso examples guide (link)
[9] Resources – Linux code base (link)
[10] Resources – Android patches (link)
[11] Resources – MCUXpresso code (link)
[12] Specification – NCI Specifications (link)
[13] Application note – AN13218 – PN7160 RF settings guide (link)
[14] Application note – AN12991 – PN7160 evaluation kit quick start guide (link)
[15] User manual – UM11496 – PN7160 evaluation board (link)
[16] Application note – AN13861 – PN7160 card emulation (link)
[17] Firmware – PN7160 FW (link)

28 Note about the source code in the document

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

29 Revision history

Table 16. Revision history

Document ID	Release date	Description
AN13892 v.1.2	06 September 2024	Editorial changes • Section 1 "Introduction": updated. • Section 2.4 "PMU configurator": updated, removed the attachment, *PMU_CONFIG_PN7160.xlsx* available under **Design Resources** on the PN7160 product page. • Section 4 "Dynamic power control - configuration": added default configuration and link to DPC video. • Section 7.1 "EMI recommendations": updated. • Section 12.2 "OM27160 - RF registers setting": updated, table with default values moved to AN13218, added link to video training. • Section 18 "How can I disable PN7160 Tag Detector mode?": added. • Section 24 "FW update": updated (12.50.06 → 12.50.0**5**). • Section "Annex 1: RF settings optimized for OM27160" removed. • Section 27 "References": updated.
AN13892 v.1.1	12 October 2023	• Section 2.4 "PMU configurator": added *PMU_CONFIG_PN7160.xlsx* to AN13982 as an attachment. • Section 4 "Dynamic power control - configuration": added clarificaiton, DPC check function available from the FW version 12.50.06 onwards. • Section 10 "Reader functions do not work at low temperatures": added. • Section 11 "Card emulation does not work at close range": added. • Section 11.1 "How to switch between card emulation modes?": added. • Section 24.1 "How to check the current FW version ": added. • Section 28 "Note about the source code in the document ": added • Section 26 "Abbreviations": updated
AN13892 v.1.0	05 May 2023	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.

EdgeVerse — is a trademark of NXP B.V.

ICODE — is a trademarks of NXP B.V.

MIFARE — is a trademark of NXP B.V.

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

NTAG — is a trademark of NXP B.V.

Tables

Figures

Contents

Date of release: 6 September 2024
Document identifier: AN13892