AN13996

How to use the low-power features of the PN76 family NFC controller

Rev. 2.0 — 14 October 2024 Application note

Document information

Information	Content
Keywords	PN76, PN7642, ULP, ULPCD, ultra low-power card detection, LPCD, standby, low-power,
Abstract	This application note explains and shows how to use different low-power modes on the PN76 family NFC controllers using the NFC low-power modes example of the SDK.

1 Introduction

The PN76 family has many options to save power. This dodcument highlights some of the most used functions and features. The goal is for the user to understand the purpose of the different lower power modes and knows how to configure and run them in the "nfc_low_power_modes" example.

To explain the different modes and its usage, the PN76 SDK example 'nfc_low_power_modes' is used.

As not every function is discussed within this document, it is highly recommended to read the data sheet, user manual, and PN7642 NFC controller user API documentation.

The LPCD and ULPCD calibration, configuration, parameters, and technical background are explained in detail in the PN7642 design-in recommendations.

1.1 Environment

The following tools, hardware, and software have been used:

- PNEV7642A development board
- MCUXpresso IDE
- MCUXpresso PN7642 SDK
- Debugger (MCU-Link, SEGGER J-Link, ...)
- SEGGER J-Link RTT Viewer

This document does not explain how to install, configure, or set up the PNEV7642 development board and its environment. For getting started with the PNEV7642 development board, read the PN76 family evaluation board quick start guide.

1.2 Debugging

When the chip enters ultra low-power modes, the debugger loses connection, in most cases without any immediate warning.

The SDK offers the option to use UART as debug output instead of Semihost (console within the IDE). For the low-power examples, it is recommended to use UART and the J-Link Section 5 "RTT Viewer". The RTT viewer is easy to reconnect after the connection has been lost due to going to low power and subsequent waking up.

2 Low-power modes

LPCD is currently not covered in this application note and will be added in a future version.

2.1 Ultra low-power card detection (ULPCD)

The ultra low-power card detection (ULPCD) offers the highest current saving. In ULPCD mode, the only wake-up sources to exit from the card detection loop are:

1. a detected antenna detuning
2. a signal on GPIO3
3. the RF level detector (external RF-Field)
4. a reset (RESET_N) of the PN7642

Note: *The ULPCD **cannot** be used together with the DC-DC function. The PNEV7642 development board is designed to operate without DC-DC. The inductor for DC-DC can remain, and VUP is per default set (J1) to be supplied by the 5V board supply.*

For more information about the ULPCD capabilities and description, see the data sheet, user manual, and PN7642 design-in recommendations.

The usage of ULPCD is explained in Section 4.5 "Option 5: Run ULPCD with calibration".

2.1.1 ULPCD API

All APIs and their description can be found in the PN7642 NFC controller user API documentation.

The ULPCD API belongs to the Power Control and Reset Management (PCRM) system services:

◆ PN76_Sys_PCRM_EnterULPCD()

```
PN76_Status_t PN76_Sys_PCRM_EnterULPCD ( PN76_PCRM_ULPCD_Config_t  bUlpcdConfig,
                                         uint16_t                  dwWakeUpCntValue,
                                         uint8_t                   bUlpdetEnable
                                       )
```

Switch into Ultra Low Power Card Calibration/Detection mode.

Parameters

bUlpcdConfig	Selection of ULPCD Configuration as in Enum **PN76_PCRM_ULPCD_Config_t**.
dwWakeUpCntValue	12 bit wake-up counter value: resolution=1ms -> max time slot=4.096s. This is interval defines the wakeup time during calibration phase and time between RF pings during card detection.
bUlpdetEnable	Ultra low power RF detection enabled/Disabled. Value to be TRUE means 1 for ULPDET Enable or FALSE means 0 for ULPDET Disabled. Any other value is not considered for ULPDET .

Return values

PN76_STATUS_SUCCESS	: SUCCESS is never returned as the system enters into Ultra low power mode in case of successful entry to ULP mode.
PN76_STATUS_PARAMETER_ERROR	Incase invalid wakeup source is selected or wake-up interval is more than 12bits

Figure 1. API: Enter ULPCD

2.2 Ultra low-power standby

In the ultra low-power Standby mode, everything is turned off except a timer based on an ultra low frequency oscillator (ULFO). In this mode, only two options are available for wake-up:

1. Wake-up by time-out (resolution: 1 ms, max time-out 4096 ms)
   a. An external RF-Field is ignored and does not cause a wake-up.
2. External RF-Field detected or wake up by time-out.
   a. The wake-up timer is mandatory.

This mode is typically used if users want to save as much power as possible and waking up in intervals, or triggered by an external RF-Field, is sufficient.

The main difference of ULP-Standy to ULPCD (ultra low-power card detection) is that there is no card detection cycle and a wake-up timer is mandatory.

2.2.1 ULP Standby API

All APIs and their description can be found in the PN7642 NFC controller user API documentation.

The ULP Standby API belongs to the PCRM (Power Control and Reset Management) System Services:

◆ PN76_Sys_PCRM_EnterUlpStandby()

```
PN76_Status_t PN76_Sys_PCRM_EnterUlpStandby ( PN76_PCRM_UltraLowPower_WakeUp_Config_t wWakeupSource,
                                              uint16_t                                 dwWakeUpCntValue
                                            )
```

Switch into Ultra Low Power standby mode.

Parameters

wWakeupSource Selection of wake-up source in Enum **PN76_PCRM_UltraLowPower_WakeUp_Config_t**.

dwWakeUpCntValue 12 bit wake-up counter value: resolution=1ms -> max time slot=4.096s. This is used as a wakeup time interval when wakeup source is TIMER.

Note

It is mandatory to have TIMER as wakeup source in all cases.
If external RF is configured as wakeup source, the IC wakes up if there is no external RF within the wakeup counter interval.
Higher priority for wakeup is always the Wakeup counter.

Return values

PN76_STATUS_SUCCESS : SUCCESS is never returned as the system enters into Ultra low power in case of successful entry to ULP mode.

PN76_STATUS_PARAMETER_ERROR In case invalid wakeup source is selected or wake-up interval is more than 12bits

Figure 2. API: Enter ULP Standby

Table 1. EnterUlpStandby wake-up sources

wWakeupSource	Behavior
E_PN76_PCRM_ULP_WAKEUP_SOURCE_TIMER	The chip will enter ULP standby and wake-up after the timer, set by "dwWakeUpCntValue", expires. An external RF-Field is ignored.
E_PN76_PCRM_ULP_WAKEUP_SOURCE_RFFIELD	The chip will enter ULP standby and wake-up either after the timer expires or an external RF-Field is detected.

3 Wake-up reasons

Depending on the low-power mode, there are various configurable reasons that will cause PN7642 to wake up. The following chapters explain some of them in more detail.

3.1 GPIO 3 Abort

In Ultra-Low-Power-Card-Detection (ULPCD) the GPIO 3 can be used to abort it from an external source (host, power management unit, etc..). An ultralow frequency oscillator drives a wake-up counter, which triggers a periodic activation.

While the GPIO 3 can be raised at any time, the check if it has been raised is only done after the wake-up counter elapsed. This results in inconsistent latency from GPIO 3 edge to actual wake-up. The latency can vary from the *standby time maximum + 1 ms* to a theoretical minimum of 1 ms.

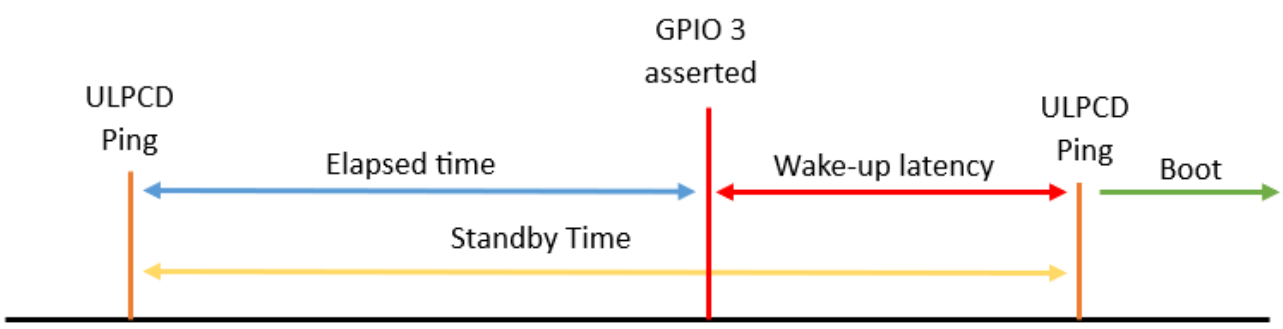

Figure 3. GPIO3 wake-up time variance

ULPCD Ping: A short RF pulse measuring the RSSI value.

Standby Time: The configured time between the wake-ups.

GPIO3: Depending on the configured polarity either a falling or rising edge on GPIO 3 of the PN7642.

Wake-up latency: Time from GPIO 3 trigger to actual PN7642 boot.

The wake-up time is: → **Wake-up latency** = Standby Time - Elapsed time

GPIO 3 is checked every time when the card detection is started. If GPIO 3 is still high, for example, from a previous abort and has not been deasserted, the ULPCD is aborted again. Best practice is to assert GPIO 3 for a short time, long enough to have a stable logical high, and deasserted afterward.

GPIO 3 abort cannot be disabled. If this option shall not be used, the pin can be tied to the ground.

3.2 Card detected

An object is detected to be in the proximity of the reader, when the measured RSSI differs from the reference RSSI by more than a configurable threshold.

In the measurement phase, the card detection activity is performed autonomously by the hardware at configurable time intervals (standby time). This configuration is passed as a parameter in the ULPCD API. The RSSI value is measured and compared against the reference value stored at the calibration phase.

Note: *For ULPCD, only RSSI is considered, not I/Q.*

3.3 External RF detected

When calling the ULPCD API "*PN76_Sys_PCRM_EnterULPCD(…)*" the third parameter "*bUlpdetEnable*" determines if an external RF-Field wakes up the PN7642 wakes up or not.

If the value is not 0, the RF level detector is enabled while in ULPCD. Any detection of an external RF-Field causes the immediate wake-up from ULPCD with the boot reason "BOOT_ULPCD_RX_ULPDET" (see register PCRM_SYS_BOOT1_STS 0xC6).

The low-power mode example prompts the following text: `Boot from ULPCD due to External RF detection`

```
00>  Low Power Mode Example:
00> Boot from ULPCD due to External RF detection
00>  Select the Modes :
```

Figure 4. Boot due to external RF detection

4 NFC low-power mode example

The MCUXpresso SDK (see [5]) contains a low-power mode example called `nfc_low_power_mode`, which includes different low-power modes. The following chapters provide detailed information about the example, and highlights the APIs and their usage.

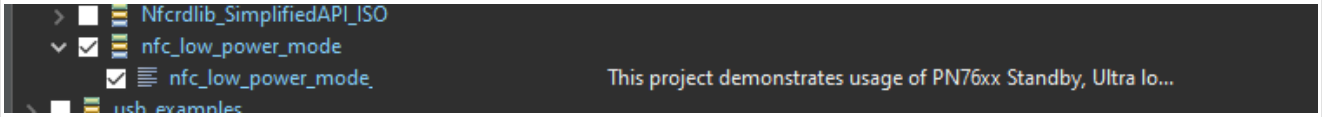

Figure 5. MCUXpresso low-power mode example

The purpose of the example is to showcase certain functionality. For product purposes, it is recommended for users to familiarize themselves with the used APIs and alter the example to their needs. Transfer the necessary additions to the application code.

Instructions for installing MCUXpresso and importing the SDK and its examples can be found in the PN76 family evaluation board quick start guide.

Running the low-power mode example presents the following options:

```
00>  Low Power Mode Example:
00>  Select the Modes :
00>  1 -> Enter Standby with Counter as Wakeup :
00>  2 -> Enter Standby with External RF as Wakeup :
00>  3 -> Enter Ultra Low Power Standby with Counter as Wakeup :
00>  4 -> Enter Ultra Low Power Standby with External RF as Wakeup :
00>  5 -> Run ULPCD with Calibration:
00>  6 -> Run SW LPCD with Calibration:
```

Figure 6. Low-power mode example options

The following chapters explain Section "Option 3: Enter ultra low-power standby with counter as wake-up", Section "Option 4: Enter ultra low-power standby with external RF as wake-up", and Section "Option 5: Run ULPCD with calibration" in detail; which APIs are used, the expected behavior, and how to apply these options to the application code of the user.

Options 1, 2 and 6 are not covered in this document.

4.1 EEPROM read/writes

!Caution! EEPROM writes must be used with extreme care, due to limited write cycles and potential for corruption if power is lost during the operation! A reset or power loss during an EEPROM write can corrupt the currently written EEPROM page. If an EEPROM write is performed make sure that stable power is available and no reset is performed until the write is complete. **!Caution!**

The example *pnev7642fama_nfc_low_power_mode* contains EEPROM reads and writes in the method `ULPCD_Get_HFATT_Value(…)`. It is important to understand that the examples are made to run plug-and-play as easily as possible. For that reason, the the `ULPCD_Get_HFATT_Value()` function is used in the demo code to ensure the correct configuration of the board for these low-power demos. This decision is based on the knowledge that other NFC examples or applications may have been run on the board and set configurations that are not compatible with the Low-Power demo.

In a final application, `ULPCD_Get_HFATT_Value()` and the code contained in it should only be executed once, during the manufacture and configuration phase of the product lifecycle.

These parameters are detailed in subsections Section 4.1.1, Section 4.1.2, and Section 4.1.3.

4.1.1 DC-DC power configuration

For ULPCD operation the DC-DC must be disabled. The example checks if the DC-DC is disabled, and if not, corrects it. A customer building their setup must be aware if they are using DC-DC or not. They would configure the registers only once, during manufacture.

- If the DC-DC is used, ULPCD is not possible.
- If no DC-DC is used, it shall be set once at the configuration phase. Not at every application start.

```
wStatus = PN76_ReadEeprom(&bReadEEPROM_Val, (uint32_t )PN76_USER_PMU->PwrConfig,
0x01U,E_PN76_EEPROM_SECURE_LIB_CONFIG);
if (bReadEEPROM_Val != LOWPOWERMODE_EXAMPLE_ULPCD_VUP_TO_VBAT_CONFIG_VAL) //!= 0x01
{
    /* Update the EEPROM Power Configuration value to VUP connected to VBat. */
    bReadEEPROM_Val = LOWPOWERMODE_EXAMPLE_ULPCD_VUP_TO_VBAT_CONFIG_VAL;
    wStatus = PN76_WriteEeprom(&bReadEEPROM_Val, (uint32_t )PN76_USER_PMU->PwrConfig,
0x01U,E_PN76_EEPROM_SECURE_LIB_CONFIG);
    PN76_CHECK_SUCCESS(wStatus);
}
```

Figure 7. Code - DCDC_PWR_CONFIG

→ ***Conclusion***: No EEPROM write for DCDC configuration is necessary in the final application.

4.1.2 TX_LDO VDDPA setting

This value defines the TX_LDO output during normal operation (that is not ULPCD), and is set to a safe and moderate voltage for the evaluation board. This value must be adapted to the implementers needs at the configuration phase. This value does not need to be changed.

```
wStatus = PN76_ReadEeprom(&bReadEEPROM_Val, (uint32_t )PN76_USER_PMU->TxldoStartVddpa,
0x01U,E_PN76_EEPROM_SECURE_LIB_CONFIG);
    PN76_CHECK_SUCCESS(wStatus);

/* VDDPA High EEPROM Configuration is updated to work with VDDPA value of 2.7V.
    * This can be different based on user configuration to use PN76XX Reader IC. */
if (bReadEEPROM_Val != bVddpa_Val)
{
    wStatus = PN76_WriteEeprom(&bVddpa_Val, (uint32_t)PN76_USER_PMU->TxldoStartVddpa, 0x01U,
E_PN76_EEPROM_SECURE_LIB_CONFIG);
    PN76_CHECK_SUCCESS(wStatus);
}
```

Figure 8. Code - TX_LDO VDDPA setting

→ ***Conclusion:*** The VDDPA EEPROM parameter does not have to be changed in the application code.

4.1.3 HF Attenuator value

A big part of the initial setup is the HF Attenuator value. This value depends on the antenna and hardware setup. Which does not change after production.

The value is read out once, and written into EEPROM as part of a one-time calibration phase.

Note: This value is determined using the same TX_LDO_VDDPA value that is used during normal operation. It is not related to the ULPCD_VDDPA value.

```
/* Read EEPROM ULPCD calibration done status value. */
wStatus = PN76_ReadEeprom(&bUlpcdCalibDone,(uint32_t )PN76_ULPCD_CALIB_STATUS->bCalibDone,
0x01U,E_PN76_EEPROM_USER_AREA);
PN76_CHECK_SUCCESS(wStatus);
if( !bUlpcdCalibDone )
{

/* Read DPC Configuration value. */
wStatus = PN76_ReadEeprom(&bDPC_EEPROM_Val,(uint32_t )PN76_DPC_SETTINGS->bConfig,
0x01U,E_PN76_EEPROM_SECURE_LIB_CONFIG);
PN76_CHECK_SUCCESS(wStatus);

if (bDPC_EEPROM_Val != 0x00U)
{
    bReadEEPROM_Val = 0x00U;
    bDPC_EEPROM_Val = 0x00U;
    /* Disable DPC to get HFATT value by turning ON the RF Field. */
    wStatus = PN76_WriteEeprom(&bReadEEPROM_Val,(uint32_t )PN76_DPC_SETTINGS->bConfig,
0x01U,E_PN76_EEPROM_SECURE_LIB_CONFIG);
    PN76_CHECK_SUCCESS(wStatus);
}

/* Apply Reader Mode Type A settings. */
wStatus = PN76_Sys_LoadRfConfiguration(E_PN76_LOADRF_TX_ISO14443A_106,
E_PN76_LOADRF_RX_ISO14443A_106);
PN76_CHECK_SUCCESS(wStatus);

/* Turn ON the RF Field. */
wStatus = PN76_Sys_FieldON(0);
PN76_CHECK_SUCCESS(wStatus);

/* Wait for sometime to stabilize RF ON before reading HF ATT value. */
LowPowerMode_CommonWait(1500);

/* Read HF ATT value. */
dwReadReg_Val = PN76_Sys_ReadRegister(CLIF_RXCTRL_STATUS);

/* Get the HF ATT Value and store in EEPROM to be used for ULPCD  */
bHFATT_Val = (uint8_t)((dwReadReg_Val & CLIF_RXCTRL_STATUS_RXCTRL_HF_ATT_VAL_MASK) >>
CLIF_RXCTRL_STATUS_RXCTRL_HF_ATT_VAL_POS);
bHFATT_Val |= LOWPOWERMODE_EXAMPLE_HFATT_CTRL_USE_EEPROM_VAL_MASK;
PN76_ReadEeprom(&bHFATT_CurrentVal, (uint32_t)(PN76_ULPCD_CONFIG->Agc_Hfatt_Ctrl), 0x01U,
E_PN76_EEPROM_SECURE_LIB_CONFIG);
PN76_CHECK_SUCCESS(wStatus);
if((bHFATT_Val > (bHFATT_CurrentVal + 5)) || (bHFATT_Val < (bHFATT_CurrentVal + 5)) )
{
    wStatus = PN76_WriteEeprom(&bHFATT_Val, (uint32_t)PN76_ULPCD_CONFIG->Agc_Hfatt_Ctrl, 0x01U,
E_PN76_EEPROM_SECURE_LIB_CONFIG);
    PN76_CHECK_SUCCESS(wStatus);
}
```

Figure 9. Code - Get HF_ATT value

→ ***Conclusion:*** The HF ATT value configuration must be done only once. No EEPROM write is necessary in the final application.

4.2 Boot reasons

Before the example prompts the options to exercise different low-power modes, the boot reason of the IC is checked. This check is done in the method `BootLowPowerCheck()` and serves multiple purposes.

First, the boot reason is evaluated to decide which further actions must be taken. An example is going to ULPCD card detection, after the calibration cycle. Second, to display the boot reason in the debug output.

The cause of boot can be retrieved by reading the register `PCRM_SYS_BOOT1_STS`.

Table 2. PCRM_SYS_BOOT1_STS_REG bit fields

Bits	Field	Access	Reset Value	Description
0	BOOT_POR	r	0x0	Bootup Reason
1	BOOT_RXPROT	r	0x0	Bootup Reason
2	BOOT_VUPDET	r	0x0	Bootup Reason
3	BOOT_TEMP	r	0x0	Bootup Reason
4	BOOT_WUC	r	0x0	Bootup Reason
5	BOOT_VDDIO_START	r	0x0	Bootup Reason this is valid if STBY/SUSPEND entered with VDDIO LOSS
6	BOOT_VDDIO_LOSS	r	0x0	Bootup Reason
7	BOOT_PVDDLDO_OVERCURRENT	r	0x0	Bootup Reason
8	RESERVED	–	0x0	reserved
9	RESERVED	–	0x0	reserved
10	BOOT_ULPCD_GPADC_READY_TIMEOUT	r	0x0	ULPCD exit as GPADC READY is not asserted from GPADC Analog
11	BOOT_RX_ULPDET	r	0x0	Bootup Reason
12	BOOT_LPDET	r	0x0	Bootup Reason
13	BOOT_GPIO0	r	0x0	Bootup Reason
14	BOOT_GPIO1	r	0x0	Bootup Reason
15	BOOT_GPIO2	r	0x0	Bootup Reason
16	BOOT_GPIO3	r	0x0	Bootup Reason
17	RESERVED	r	0x0	Reserved
18	RESERVED	r	0x0	Reserved
19	BOOT_I2C	r	0x0	Bootup Reason
20	BOOT_SPI	r	0x0	Bootup Reason
21	WAKEUP_RX_ULP	r	0x0	Indicates if VEN is masked due to wake-up with rx_ulp. This bit is cleared with ULPDET_WKUP_VEN_MASK_CLR
22	BOOT_ULPCD_RX_ULPDET	r	0x0	RX ULPDET resulted in boot in ULPCD mode
23	RESERVED	r	0x0	Reserved
24	BOOT_USB	r	0x0	Bootup Reason
25	BOOT_ULPCD_LDO_VDDPA_ OVERCURRENT	r	0x0	ULPCD LDO VDDPA overcurrent
26	BOOT_ULP_STANDBY	r	0x0	ULPCD Standby
27	BOOT_ULPCD_GPIO_ABORT	r	0x0	ULPCD GPIO Abort
28	BOOT_ULPCD_CALIBRATION_DONE	r	0x0	ULPCD calibration complete
29	BOOT_ULPCD_CARD_DETECT	r	0x0	ULPCD card detect
30	BOOT_ULPCD_CLKDET_ERROR	r	0x0	ULPCD CLK_DET error
31	BOOT_ULPCD_XTAL_TIMEOUT	r	0x0	ULPCD exit due to XTAL timeout

4.3 Option 3: Enter ultra low-power standby with counter as wake-up

Option 3 of the low-power modes example sets the chip into ultra low-power standby. The wake-up counter can be set to predefined options (in Figure 10, the wake-up counter "1" represents 1 second):

```
00>  Low Power Mode Example:
00>  Select the Modes :
00>  1 -> Enter Standby with Counter as Wakeup :
00>  2 -> Enter Standby with External RF as Wakeup :
00>  3 -> Enter Ultra Low Power Standby with Counter as Wakeup :
00>  4 -> Enter Ultra Low Power Standby with External RF as Wakeup :
00>  5 -> Run ULPCD with Calibration:
00>  6 -> Run SW LPCD with Calibration:
  < 3
00> 3
00>  Select the Wake Up interval from below Options :
00>  0 -> 500ms :
00>  1 -> 1000ms :
00>  2 -> 2690ms (Low Power Standby Max Interval) :
00>  3 -> 4095ms (Ultra Low Power Max Interval) :
  < 1
```

Figure 10. Option 3 execution

After the interval selection, it immediately enters low-power mode. After 1 second, the chip wakes up again. If we reconnect the RTT viewer, we can see that the boot reason is wake-up counter expiry (see Figure 11):

```
00>  Low Power Mode Example:
00> Boot after ULP Standby with WakeUp counter expiry
00>  Select the Modes :
00>  1 -> Enter Standby with Counter as Wakeup :
00>  2 -> Enter Standby with External RF as Wakeup :
00>  3 -> Enter Ultra Low Power Standby with Counter as Wakeup :
00>  4 -> Enter Ultra Low Power Standby with External RF as Wakeup :
00>  5 -> Run ULPCD with Calibration:
00>  6 -> Run SW LPCD with Calibration:
```

Figure 11. Boot reason counter expiry

4.3.1 Code flow

The code flow for this option is rather simple. In the *switch block* the case `ULP_STANDBY_WITH_WAKEUP_COUNTER_MODE` is taken, the method `Demo_UltraLowPowerStandby(…)` with the chosen interval is called.

The method `Demo_UltraLowPowerStandby(…)` is simple as well; only calling the system service API from the PN76 to enter ultra low-power standby: `PN76_Sys_PCRM_EnterUlpStandby(…)`

```
PN76_Status_t Demo_UltraLowPowerStandby(uint16_t wWakeUpSource, uint16_t wWakeUpIntervalInMs)
{
   PN76_Status_t wStatus = PN76_STATUS_SUCCESS;

   do{
      PN76_Sys_PCRM_EnterUlpStandby((PN76_PCRM_UltraLowPower_WakeUp_Config_t)wWakeUpSource, wWakeUpIntervalInMs);

        APP_PRINTF("\r\n Ultra Low Power Standby Entry FAILED!!!!! : ");

      wStatus = PN76_STATUS_INTERNAL_ERROR;
    } while (0);

    return wStatus;
}
```

Figure 12. API PCRM EnterUlpStandby

4.4 Option 4: Enter ultra low-power standby with external RF as wake-up

This option is calling the ultra low-power standby method with the configuration parameter set to `E_PN76_PCRM_ULP_WAKEUP_SOURCE_RFFIELD`. This sets the chip into *ultra low-power standby* with wake-up either by timer expiry or external RF-Field.

See Section 2.2.1 for more details of the API.

Selecting option 4 and interval "2" (which provides more time to manually place something on the antenna that emits an RF-Field and would therefore wake the chip) will put the chip into ULP standby (see Figure 13):

```
00>  Low Power Mode Example:
00>  Select the Modes :
00>  1 -> Enter Standby with Counter as Wakeup :
00>  2 -> Enter Standby with External RF as Wakeup :
00>  3 -> Enter Ultra Low Power Standby with Counter as Wakeup :
00>  4 -> Enter Ultra Low Power Standby with External RF as Wakeup :
00>  5 -> Run ULPCD with Calibration:
00>  6 -> Run SW LPCD with Calibration:
  < 4
00> 4
00>  Select the Wake Up interval from below Options :
00>  0 -> 500ms :
00>  1 -> 1000ms :
00>  2 -> 2690ms (Low Power Standby Max Interval) :
00>  3 -> 4095ms (Ultra Low Power Max Interval) :
  < 2
```

Figure 13. Option 4 execution

Bringing a device, which emits an RF-Field (for example, cell phone), into the antennas area wakes up the chip with the boot reason of external RF detected (see Figure 14, do not forget to reconnect the RTT viewer as connection is lost when entering ULP):

```
00> Boot from ULP Standby due to External RF detection
00>  1 -> Enter Standby with Counter as Wakeup :
00>  2 -> Enter Standby with External RF as Wakeup :
00>  3 -> Enter Ultra Low Power Standby with Counter as Wakeup :
00>  4 -> Enter Ultra Low Power Standby with External RF as Wakeup :
00>  5 -> Run ULPCD with Calibration:
00>  6 -> Run SW LPCD with Calibration:
```

Figure 14. Boot due to external RF

4.5 Option 5: Run ULPCD with calibration

By entering option 5 and some configuration options, the ULPCD is first calibrated and then restarted for the actual detection (see Figure 15.

```
00>  Low Power Mode Example:
00>  Select the Modes :
00>  1 -> Enter Standby with Counter as Wakeup :
00>  2 -> Enter Standby with External RF as Wakeup :
00>  3 -> Enter Ultra Low Power Standby with Counter as Wakeup :
00>  4 -> Enter Ultra Low Power Standby with External RF as Wakeup :
00>  5 -> Run ULPCD with Calibration:
00>  6 -> Run SW LPCD with Calibration:
  < 5
00> 5
00>  Select the Wake Up interval from below Options :
00>  0 -> 500ms :
00>  1 -> 1000ms :
00>  2 -> 2690ms (Low Power Standby Max Interval) :
00>  3 -> 4095ms (Ultra Low Power Max Interval) :
  < 0
00> 0
00>  Select the RF ON duration during ULPCD from below Options :
00>  0 -> 5.20 usecs :
00>  1 -> 13.20 usecs (Default) :
00>  2 -> 23.6 usecs :
00>  3 -> 34.1 usecs :
00>  4 -> 45 usecs :
  < 2
00> 2
```

Figure 15. Option 5 execution

After entering all the options, the chip goes into ULPCD. On the eval board, you can see the blue LED pulse on every ~320 ms (ULPCD RF-Field cycle). During this time, the RF-Field is also turned on for a short period (RF-On duration from above) to check for any potential detuning.

Inspecting the RF-Field with an oscilloscope shows a short pulse every ~320 ms (depending on the setting above and the LFO settings). See Figure 16:

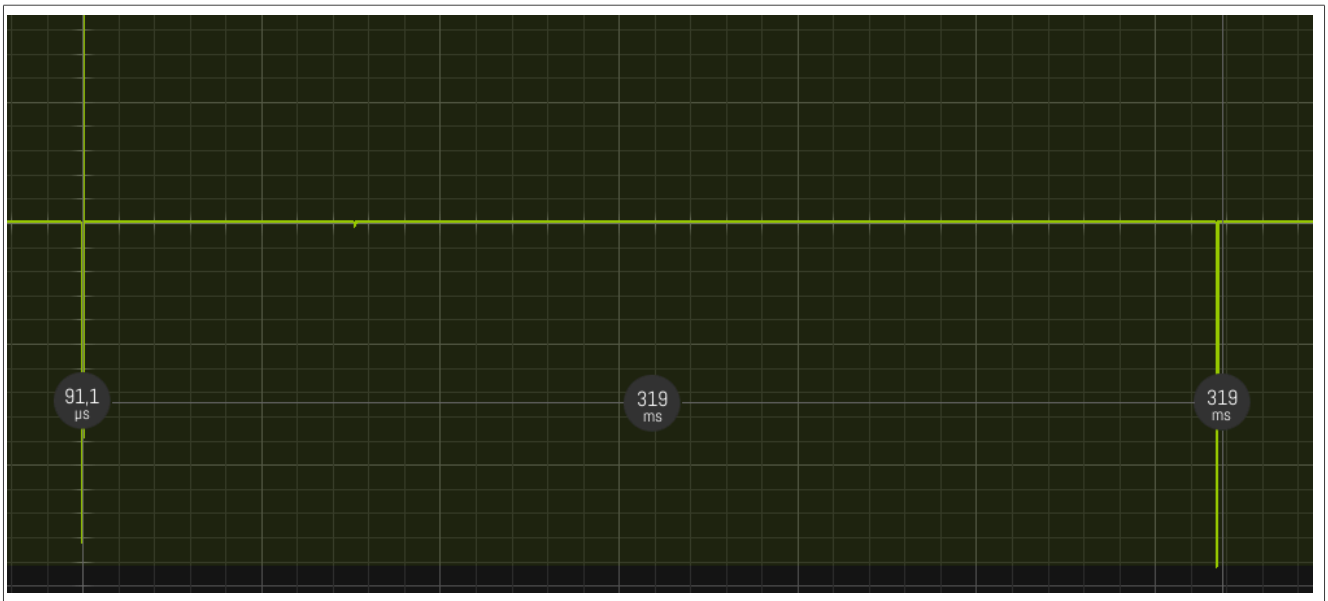

Figure 16. ULPCD oscilloscope inspection

If you bring a card into the RF-field, the chip wakes up and the boot reason is card detection (see Figure 17). Reconnect the RTT viewer to see the output, as it looses connection when entering ULPCD:

```
00> Low Power Mode Example:
00> Boot after ULPCD exit with CARD Detect
00> Select the Modes :
00> 1 -> Enter Standby with Counter as Wakeup :
00> 2 -> Enter Standby with External RF as Wakeup :
00> 3 -> Enter Ultra Low Power Standby with Counter as Wakeup :
00> 4 -> Enter Ultra Low Power Standby with External RF as Wakeup :
00> 5 -> Run ULPCD with Calibration:
00> 6 -> Run SW LPCD with Calibration:
```

Figure 17. ULPCD boot reason card detection

4.5.1 Code flow

Following the flow of the code, we see that the API "*PN76_Sys_PCRM_EnterULPCD()*" is called with "*E_PN76_PCRM_ULPCD_CALIBRATION*" as the first parameter:

```
342     /* Perform ULPCD Calibration Cycle */
343     wStatus = PN76_Sys_PCRM_EnterULPCD(E_PN76_PCRM_ULPCD_CALIBRATION, wULPCDWakeUpTime, 0);
344     /* Control never reaches here, adding for warning removal */
345     return  PN76_STATUS_SUCCESS;
```

Figure 18. Example: Enter ULPCD calibration mode

As the comment at line 344 implies, the debugger is never supposed to reach this line. This is due to the behavior of the ultra low-power modes. Everything is shut down except the bare minimum to exercise ULPCD. This shutdown includes the core itself and the debugger loses connection.

Note: Ultra-low power modes cannot be live debugged!

For the ULPCD calibration, make sure that nothing is placed onto the antenna. The measured calibration values are intended to reflect an unloaded condition.

After the calibration is done, the chip wakes up automatically. Following the code flow the function "BootLowPowerCheck()" shall be investigated closer:

```
99      /* Check of for the BOOT reason after wake up */
100     BootLowPowerCheck();
```

Figure 19. Example: Method BootLowPowerCheck()

This method distinguishes between the different boot reasons. As we are coming from ULPCD calibration, file lpm.c line 387 should evaluate as true and call "ULPCD_Demo(…)":

```
387     else if (dwBootRegVal & PCRM_SYS_BOOT1_STS_BOOT_ULPCD_CALIBRATION_DONE_MASK)
388     {
389         ULPCD_Demo(wULPCDWakeUpTime);
390     }
```

Figure 20. Example: Checking wake-up reason

The initialization of parameter "*wULPCDWakeUpTime*" is done by using define. This define can be adjusted to the users liking:

```
40  /* User can use suitable PowerDown Time between successive Field ON cycle used for measurement and comparison with
41   * reference values */
42  #define ULPCD_POWERDOWN_TIME 330U
```

Figure 21. Example: Power down Time define

Opening the method "*ULPCD_Demo(…)*" shows that now the API to enter into ULPCD is called with the option "*E_PN76_PCRM_ULPCD_CARD_DETECTION*" rather than "*E_PN76_PCRM_ULPCD_CALIBRATION*" as before:

```
220 static PN76_Status_t ULPCD_Demo(uint16_t wLPCDWakeUpTime)
221 {
222     phStatus_t wStatus;
223
224     /* Perform ULPCD Detection Cycle */
225     wStatus = PN76_Sys_PCRM_EnterULPCD(E_PN76_PCRM_ULPCD_CARD_DETECTION, wLPCDWakeUpTime, 0);
226     PN76_CHECK_SUCCESS(wStatus);
227     /* Control never reaches here, adding for warning removal */
228     return  PN76_STATUS_SUCCESS;
229 }
```

Figure 22. Example: Enter ULPCD

The parameter "*wLPCDWakeUpTime*" is the before mentioned define. The description of it is misleading. See chapter Section 2.1.1 for more details.

After calling this API, the PN76 enters ULPCD with the purpose of card detection. It will not wake up from it until one of the following occurs:

- Antenna detuning (can be due to an external card or any conductive metal)
- Chip reset (power cycle, RESET_N, VEN, …)
- GPIO3
- RFLD

4.5.2 Abort with GPIO3

Another option to exit ULPCD mode is abort via GPIO3. The development board can have multiple configurations. The option shown below is not the most elegant, but is one that works without regard for the board configuration.

To abort via GPIO3, we have to apply +3.3 V to GPIO3.

Use a jumper wire to connect J47.4 (P3V3_BRD) to pin 30 on the module board:

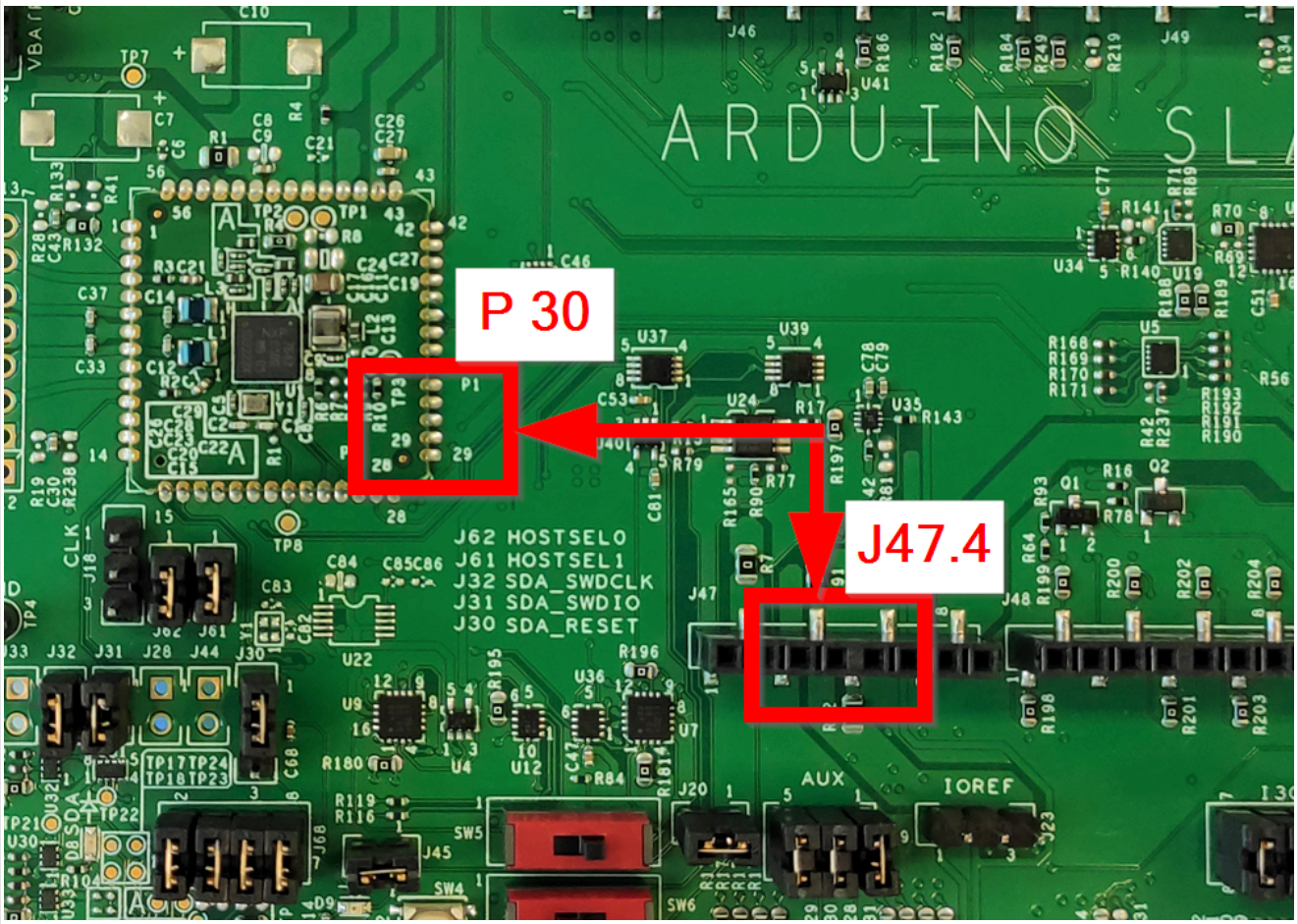

Figure 23. GPIO3 abort wiring

P.30 is the second pin of the right row (above 29).

If the chip is woken up, the green LED is turned on.

5 RTT Viewer

The J-Link RTT Viewer is a GUI tool from SEGGER, which attaches to either an existing J-Link connection of a debugger or opening a connection on its own. Instead of having the IDE console as input and output, the J-Link RTT Viewer is used. This option is more lightweight, easier to reattach.

At the import of the example, you can choose "UART" or "Semihost". UART represents the RTT Viewer option and Semihost the output in the IDE console. You can switch the debug console at any time, using the quick settings: Quick Settings → SDK Debug Console → Semihost/UART

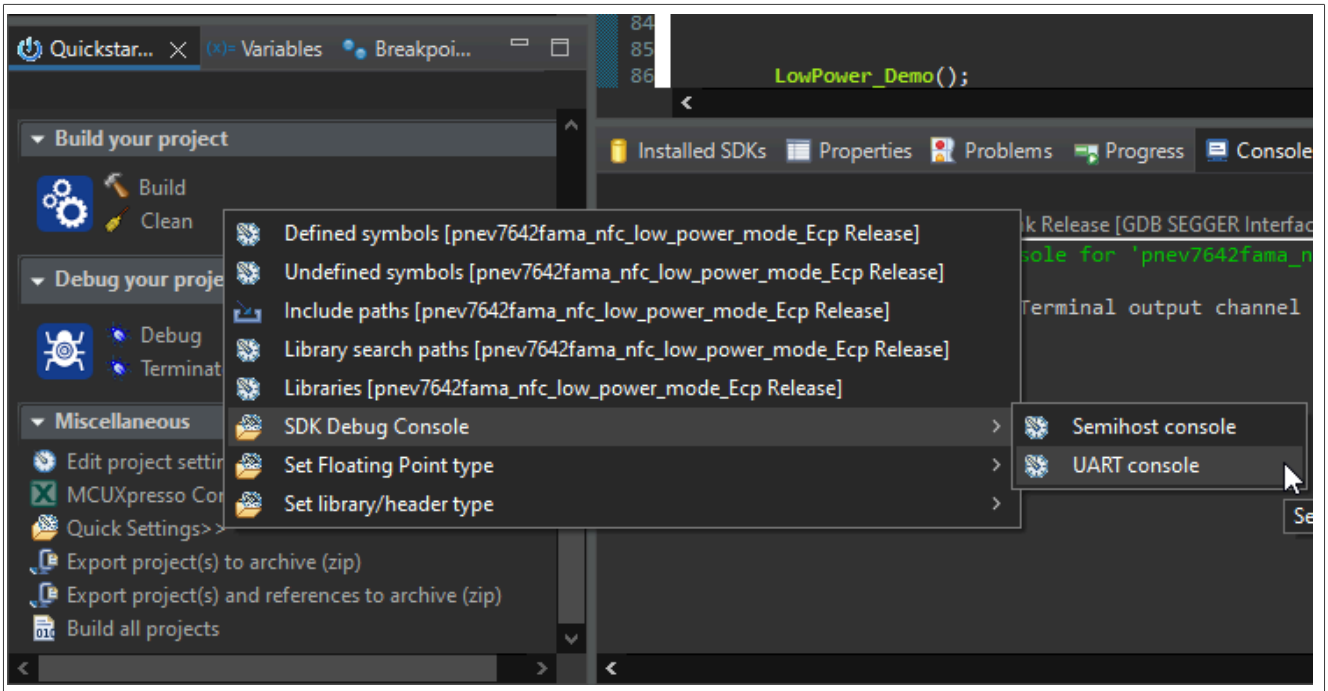

Figure 24. MCUXpresso Debug Console Option

The configuration to attach to the PN7642 is as below:

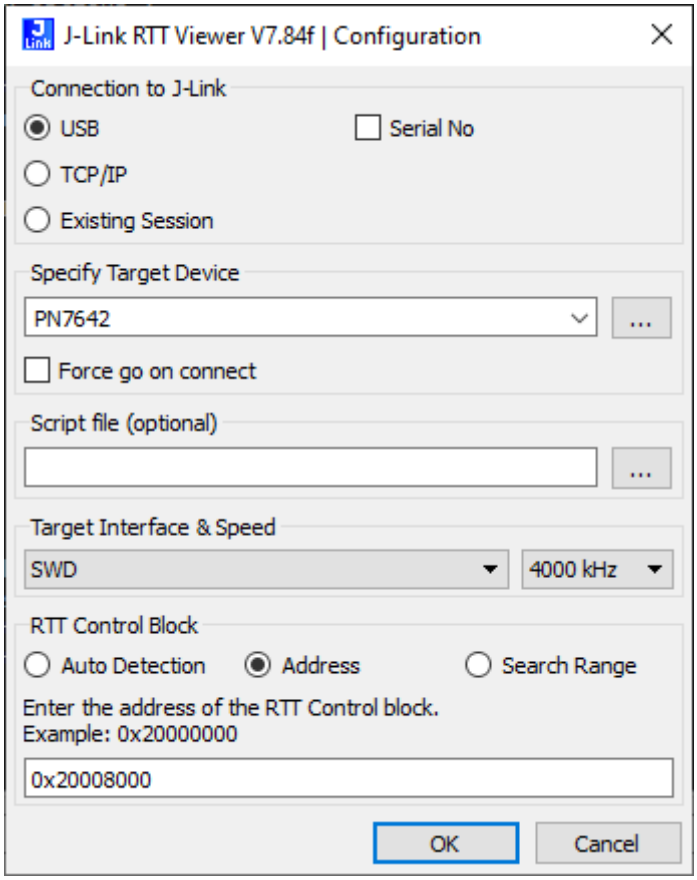

Figure 25. J-Link RTT Viewer configuration

The address of the RTT control block is: **0x20008000**

6 Tips, tricks, and FAQ

Q: What can I do if the chip is cycling through ULP Standby and I cannot connect with a debugger anymore?

A: Usually this happens if your application has no reason to stay active and directly goes into ULP Standby again. The easiest solution, on the development board, would be to bring the chip into USB Mass-Storage and replace the application with some other, which is not using any low-power modes (for example, LED Blinky).

Alternatively you can try to catch the chip in its active state. Hold reset, then release the reset and press connect of the debugger at the same time.

Q: I cannot use the J-Link RTT viewer because I do not have a SEGGER J-Link, what can I do?

A: If you have a MCU-Link Pro or LPC-Link2 instead, you can flash J-Link Lite on them. With J-Link Lite, you can use the RTT viewer. Another option can be to use semihost and indicate the boot reason with LEDs.

Q: How to visualize the RF-Field?

A: You can use the probe of the oscilloscope and connect the Ground clamp to the tip of the probe itself. By placing this loop on the antenna, you should be able to trigger easily on a rising edge to capture the RF-Field. Depending on your chosen interval time, I would adjust the time axis to your chosen interval x 4.

Q: How to retain data over ULPCD?

The PN76 has no RAM section that is not deleted during ULP/ULPCD. This decision has been made to minimize the current consumption as much as possible. The ALV (always live) RAM section is also turned off in these ultra-low-power modes. Although it can be used for other standby and sleep modes.

The PN7642 flash and EEPROM write endurance are minimum 100k cycles. As long as your code writes infrequently and you can ensure you do not exceed 100k writes, EEPROM, and flash should be acceptable.

Q: Why does the ULPCD example not wake-up by an external RF-Field?

The example calls the EnterULPCD API per default without enabling the RFLD option. To change this go to the method "ULPCD_Demo" and change the third parameter to "1".

```
static PN76_Status_t ULPCD_Demo(uint16_t wLPCDWakeUpTime)
{
    phStatus_t wStatus;

    /* Perform ULPCD Detection Cycle */
    wStatus = PN76_Sys_PCRM_EnterULPCD(E_PN76_PCRM_ULPCD_CARD_DETECTION, wLPCDWakeUpTime, 1)
    PN76_CHECK_SUCCESS(wStatus);
    /* Control never reaches here, adding for warning removal */
    return  PN76_STATUS_SUCCESS;
}
```

Figure 26. Enable RFLD at the ULPCD example

7 Abbreviations

Table 3. Abbreviations

Acronym	Description
LPCD	Low power card detection
NFC	Near Field Communication
PCRM	Power clock reset management
RFLD	Radio frequency level detector
SDK	Software development kit
ULFO	Ultra low frequency oscillator
ULP	Ultra low power
ULPCD	Ultra low power card detection

8 References

[1] Data sheet – PN7642 product data sheet (link)
[2] User manual – UM11566 – PN76 family NFC open controller (link)
[3] Resource – PN7642 NFC controller user API documentation (part of [5]→ *docs*)
[4] Application note – AN14138 – PN7642 design-in recommendations
[5] Software – PN7642 MCUXpresso SDK (link)
[6] Application note – AN13134 – PN76 family evaluation board quick start guide (link)

9 Note about the source code in the document

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

Copyright 2023-2024 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.

10 Revision history

Table 4. Revision history

Document ID	Release date	Description
AN13996 v.2.0	14 October 2024	Editorial changes. • Section 1 "Introduction": updated. • Section 2 "Low-power modes": updated. • Section 2.1 "Ultra low-power card detection (ULPCD)": updated. • Section 2.1.1 "ULPCD API": updated. • Section 2.2.1 "ULP Standby API": updated. • Section 3 "Wake-up reasons": added. • Section 4.1 "EEPROM read/writes": added. • Section 6 "Tips, tricks, and FAQ": updated. • Section 7 "Abbreviations": added. • Section 8 "References": updated.
AN13996 v.1.0	05 July 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.

J-Link — is a trademark of SEGGER Microcontroller GmbH.

SEGGER Embedded Studio — is a trademark of SEGGER Microcontroller GmbH.

Tables

Figures

Contents

Date of release: 14 October 2024
Document identifier: AN13996