AN13996

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

Rev. 1.0 — 5 July 2023 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.

Revision history

Rev	Date	Description
v.1.0	20230705	Initial version

1 Introduction

The PN76 family has many options to save power. This application note highlights some of the most used functions and features. The goal is that the user understands the purpose of the different lower power modes, knows how to configure and run them. To explain different modes and its usage, the PN76 SDK example 'nfc_low_power_modes' is used.

As not every function is highlighted within this application note, it is highly recommended to read the [1] data sheet, [2] user manual, and [3] API reference documentation.

1.1 Environment

Following tools, hardware and software have been used:

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

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 evaluation board quick start guide [5].

1.2 Debugging

When the chip enters ultra-low power modes the debugger loses connection. Most probably without any immediate warning.

The SDK offers the option to use UART as debug output instead of Semihost (console within IDE). For the low-power examples, it is recommended to use UART and the J-Link Section 4 "RTT Viewer". The RTT viewer is easy to reconnect after the connection has been lost due to going to low power and waking up later on.

2 Low-power modes

2.1 Ultra low-power card detection (ULPCD)

The ultra low-power card detection (ULPCD) offers the highest current saving. In this mode, the only wake-up sources to escape from the card detection loop are either a detected antenna detuning, a signal on GPIO3 or a reset (RESET_N) of the PN7642.

NOTE: The ULPCD **cannot** be used together with the DC-DC function.

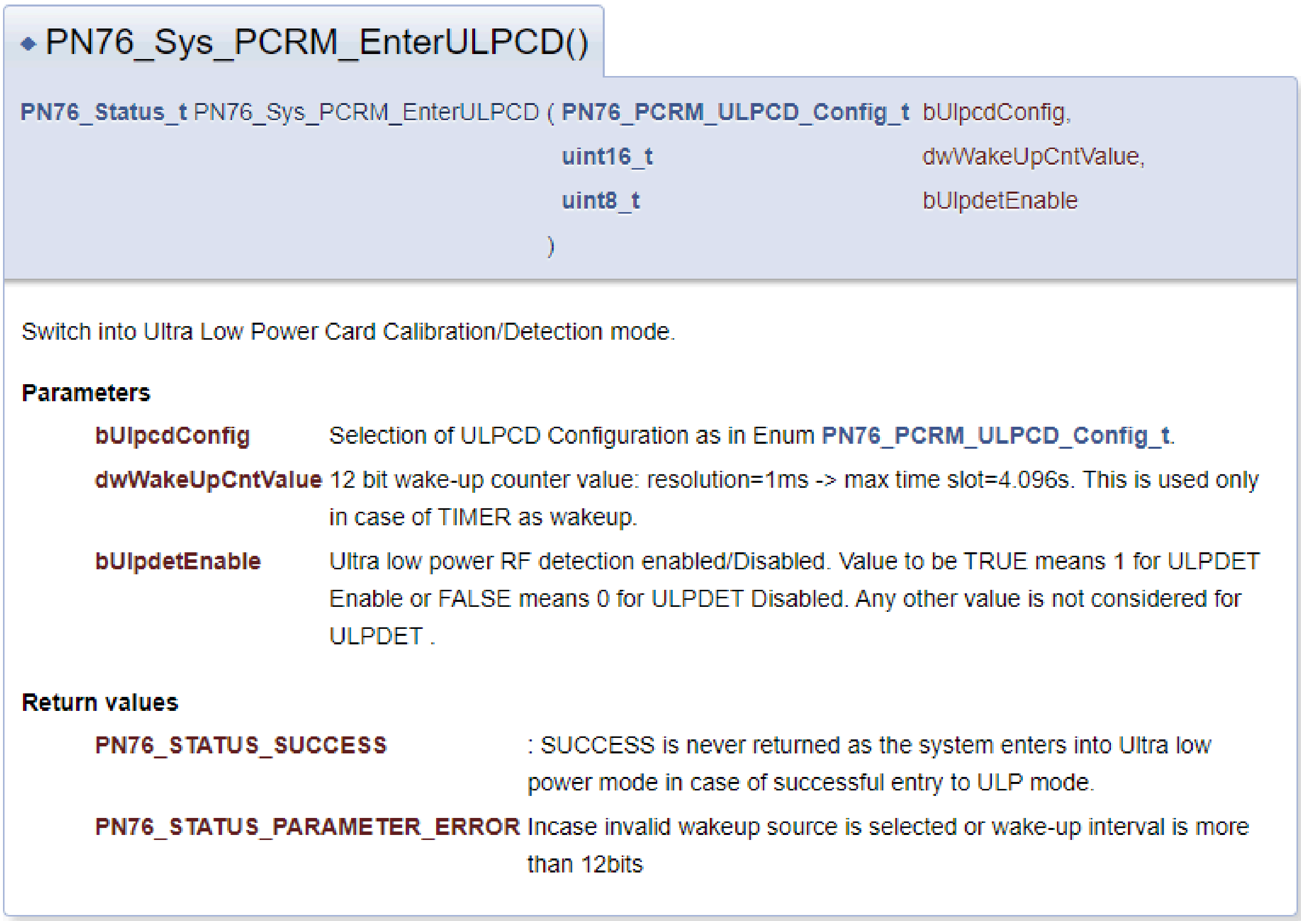
For more information about the ULPCD capabilities and description, see the [1] data sheet or [2] user manual.

The usage of ULPCD is explained in following Section 3.4 "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: [3]

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 used only in case of TIMER as wakeup.
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

ATTENTION: To this date, the description of the API is wrong (to be fixed in the next release).

The second parameters *dwWakeUpCntValue* purpose is to configure the time between the consecutive rf-pings, and so the name *dwWakeUpCntValue* is misleading, since there is no wake-up source timer.

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 to 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.

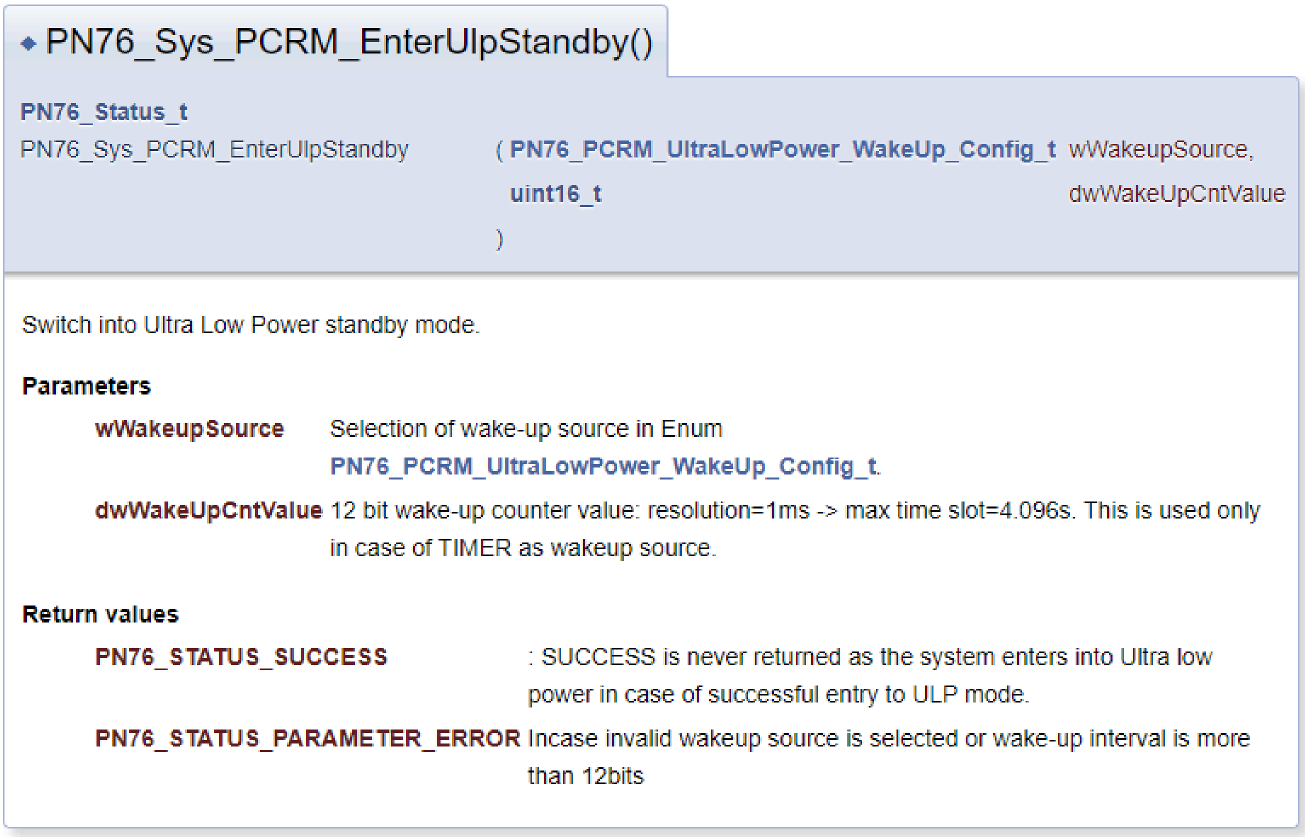
This mode is typically used if you want to save as much power possible and waking up in intervals, or triggered by an external RF-Field, is good enough.

The main difference 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: [3]

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 only in case of TIMER as wakeup source.

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	Incase invalid wakeup source is selected or wake-up interval is more than 12bits

Figure 2. API: Enter ULP Standby

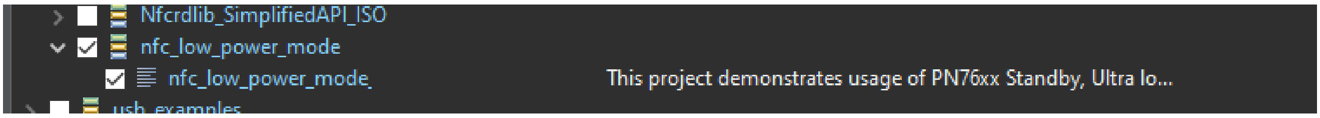
ATTENTION: To this date, the description of the API is wrong (to be fixed in the next release). The wake-up timer is mandatory in both cases, regardless of the wake-up source being timer or RF-Field!

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 NFC low-power mode example

The MCUXpresso SDK [4] contains a low-power mode example called "nfc_low_power_mode". Exercising different low-power modes. This chapter makes a deep-dive into the example and highlight the used APIs as well as pointing out how to use them.



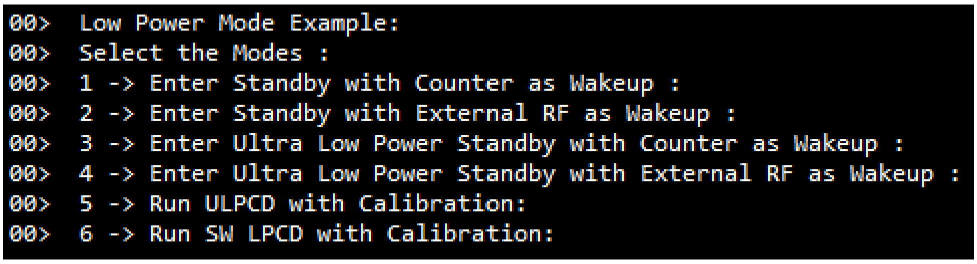
□ Nfcrdlib_SimplifiedAPI_ISO
☑ nfc_low_power_mode
☑ nfc_low_power_mode_ This project demonstrates usage of PN76xx Standby, Ultra lo...
□ usb_examples

Figure 3. MCUXpresso low-power mode example

The examples purpose is to showcase certain functionality. For product purposes, it is recommended to get yourself familiar with the used APIs and alter the example to your needs or transfer the necessities to your application code.

How to install MCUXpresso, import the SDK and its examples, read the evaluation board quick start guide [5]. This is assumed to be known for this application note.

Running the low-power mode example prompts 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 4. Low-power mode example options

Following chapters explain option 3, 4 and 5 in more detail. Which APIs are used. What is the expected behavior and how to apply this to the users application code.

Options 1, 2 and 6 are not explained in this version of the document.

3.1 Boot reason

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

First, the boot reason is evaluated to see if further actions have to be taken. Like going to ULPCD card detection, coming from a calibration cycle. Second, to prompt the boot reason in the debug output.

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

Table 2. PCRM_SYS_BOOT1_STS_REG bit fields

Description *PCRM LP System Boot 1 Status Register*

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

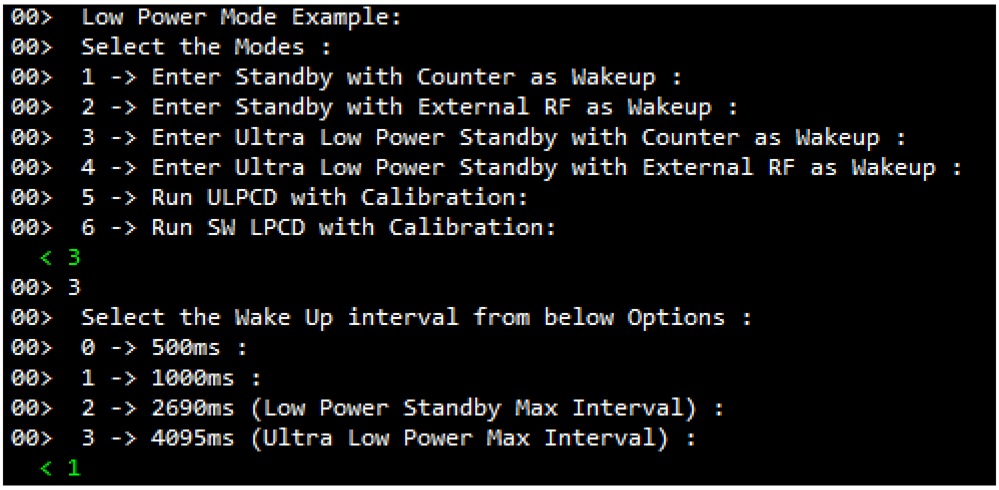
Table 2. PCRM_SYS_BOOT1_STS_REG bit fields*...continued*

Description *PCRM LP System Boot 1 Status Register*

Bits	Field	Access	Reset Value	Description
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

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

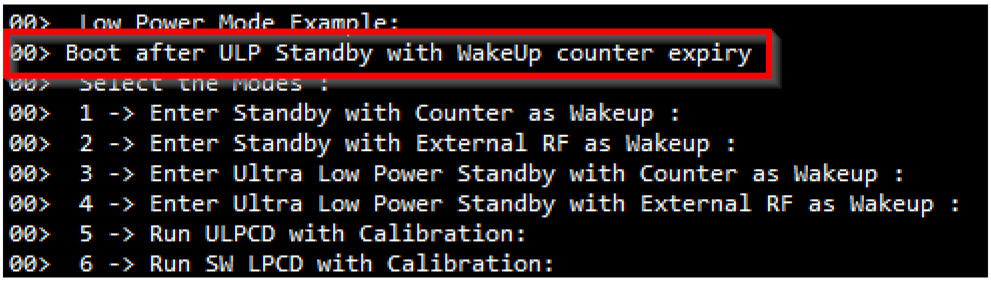
Option 3 of the low-power modes example sets the chip into ultra low power standby with the wake-up counter set to the chosen option (in this showcase "1" which represents 1 second):



```
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 5. Option 3 execution

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



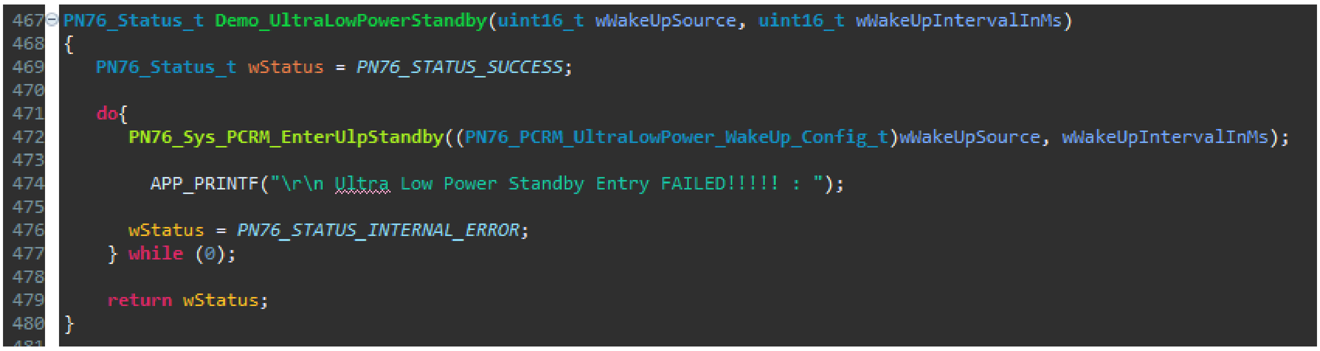
```
00>  Low Power Mode Example:
00> Boot after ULP Standby with WakeUp counter expiry
00>  Select the Modes :
00>  1 -> Enter Standby with Counter as Wakeup :
00>  2 -> Enter Standby with External RF as Wakeup :
00>  3 -> Enter Ultra Low Power Standby with Counter as Wakeup :
00>  4 -> Enter Ultra Low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

Figure 6. Boot reason counter expiry

3.2.1 Code flow

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

The method "*Demo_UltraLowPowerStandby(…)*" is very simple as well and 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;
}
```

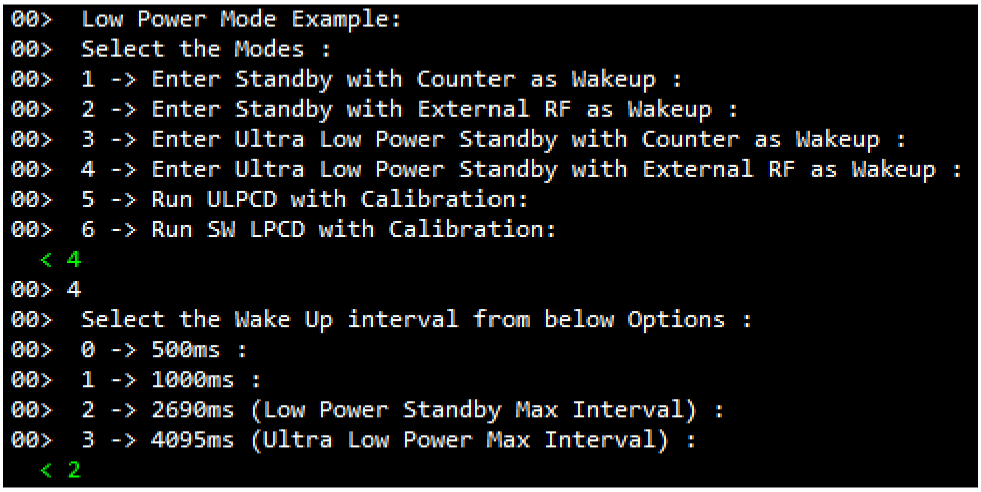
Figure 7. API PCRM EnterUlpStandby

3.3 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 and its waking up either by timer expiry or external RF-Field.

See Section 2.2.1 for more details of the used API.

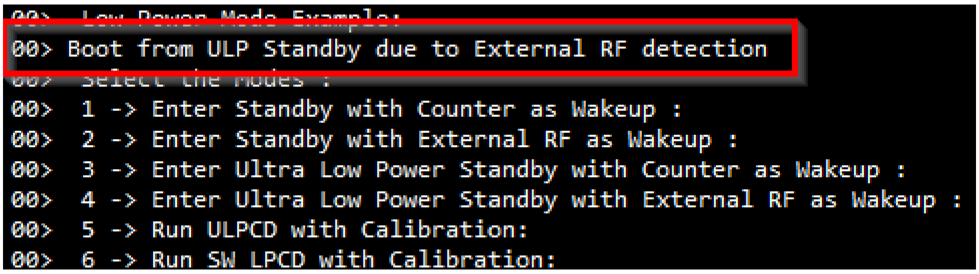
Selecting option 4 and interval of "2" (having more time to put manually something RF-Field emitting on the antenna) will put it to ULP standby:



```
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 8. Option 4 execution

Bringing a device, which emits an RF-Field (e.g. cell phone), into the antennas area will wake-up the chip with the boot reason of external RF detected (do not forget to reconnect the RTT viewer as connection is lost when entering ULP):

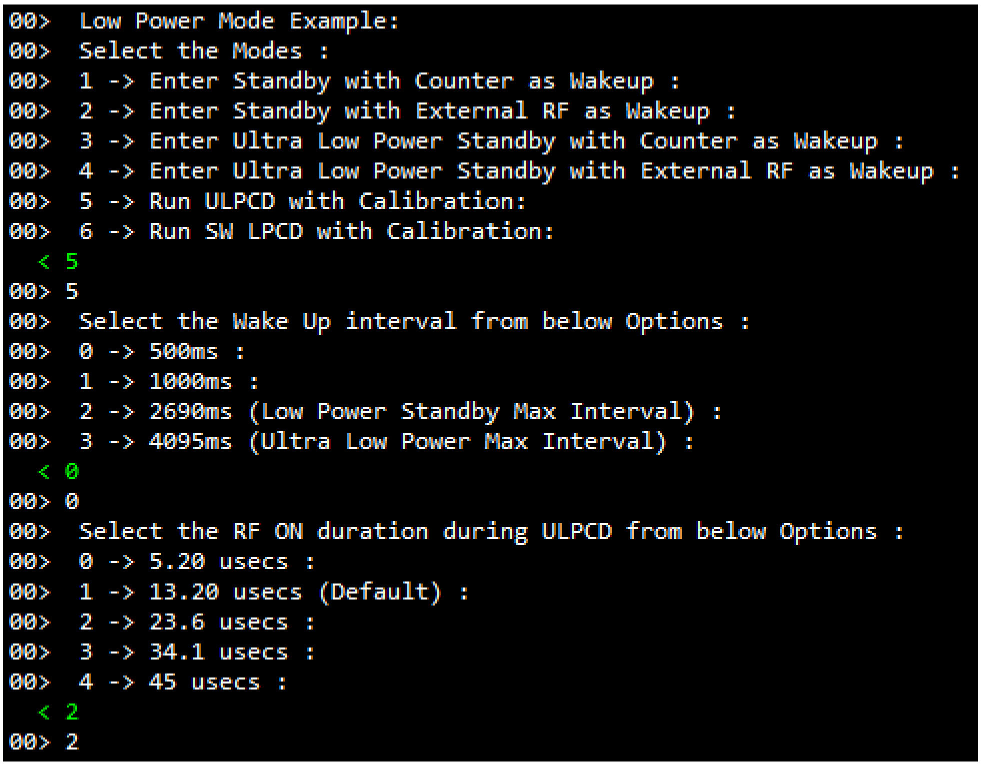


```
00>  Low Power Mode Example:
00> Boot from ULP Standby due to External RF detection
00>  Select the Modes :
00>  1 -> Enter Standby with Counter as Wakeup :
00>  2 -> Enter Standby with External RF as Wakeup :
00>  3 -> Enter Ultra Low Power Standby with Counter as Wakeup :
00>  4 -> Enter Ultra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

Figure 9. Boot due external RF

3.4 Option 5: Run ULPCD with calibration

By entering option 5, and some more configurations, the ULPCD is first calibrated and restarted for the actual detection.



```
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 10. Option 5 execution

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

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

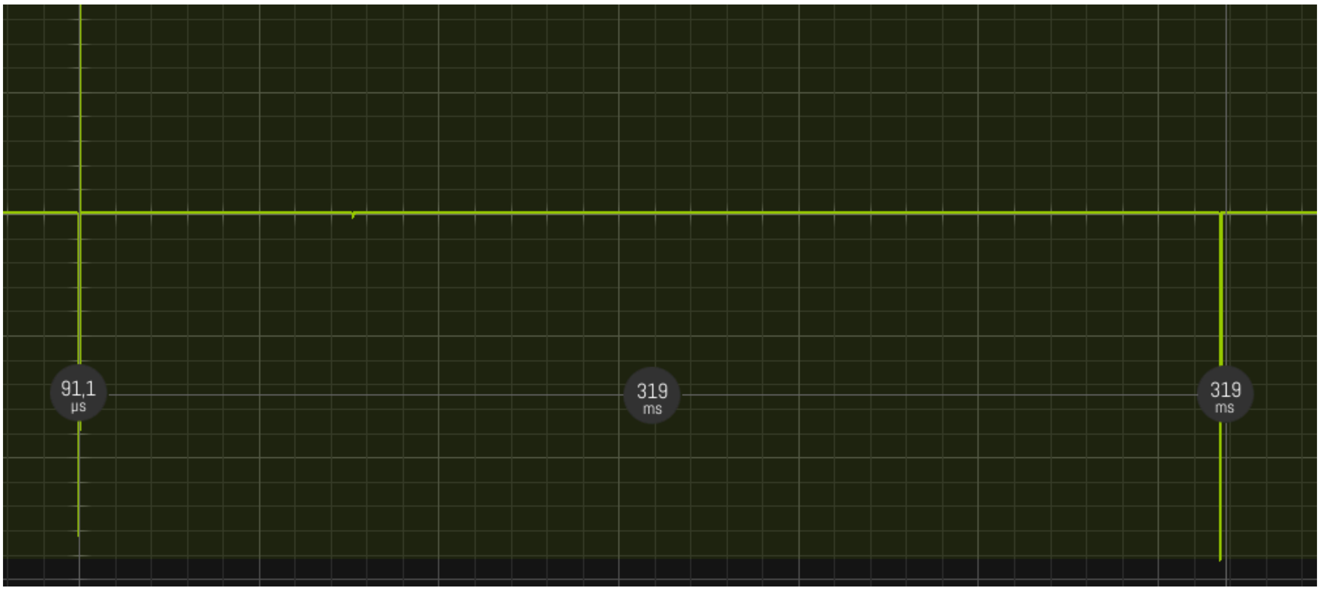
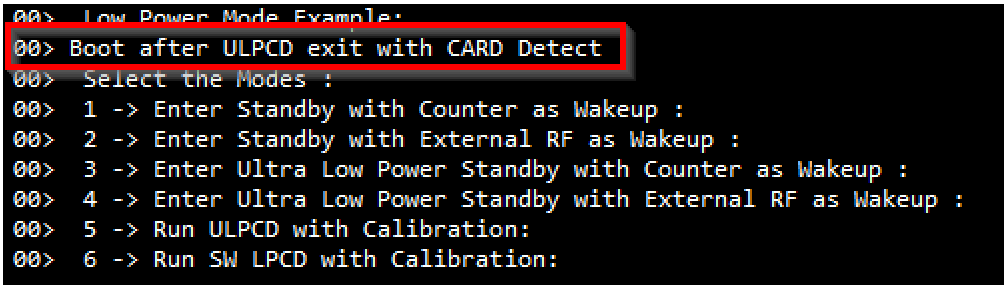


Figure 11. ULPCD oscilloscope inspection

If you bring a card into the RF-field, the chip is waking up and the boot reason will be card detection (reconnect the RTT viewer to see the output, as it looses connection when entering ULPCD):

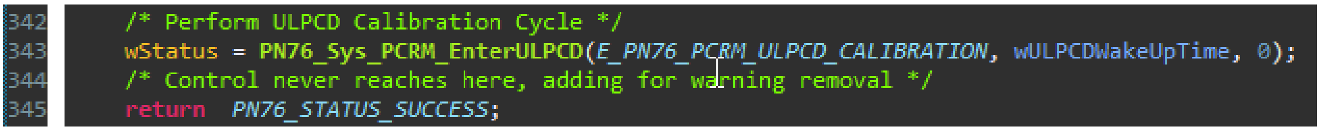


```
00>  Low Power Mode Example:
00> Boot after ULPCD exit with CARD Detect
00>  Select the Modes :
00>  1 -> Enter Standby with Counter as Wakeup :
00>  2 -> Enter Standby with External RF as Wakeup :
00>  3 -> Enter Ultra Low Power Standby with Counter as Wakeup :
00>  4 -> Enter Ultra Low Power Standby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

Figure 12. ULPCD boot reason card detection

3.4.1 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 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;
```

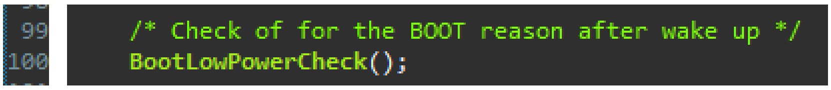
Figure 13. Example: Enter ULPCD calibration mode

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. Also the core itself is turned off 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. As the current values are taken as unloaded condition.

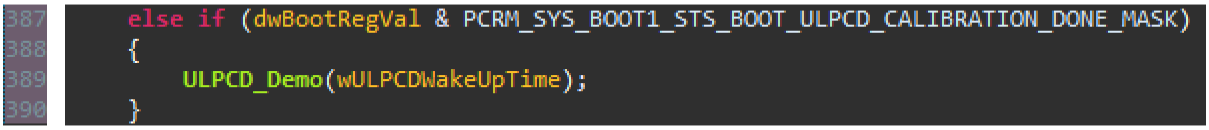
After the calibration is done, the chip wakes up automatically. Following the code flow the function "BootLowPowerCheck()" shall be investigated closer:



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

Figure 14. Example: Method BootLowPowerCheck()

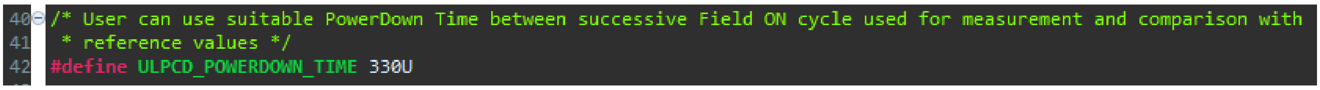
This method distinguishes between the different boot reasons. As we are coming from ULPCD calibration, file lpm.c line 387, should evaluate as true and call "ULPCD_Demo(…)":



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

Figure 15. Example: Checking wake-up reason

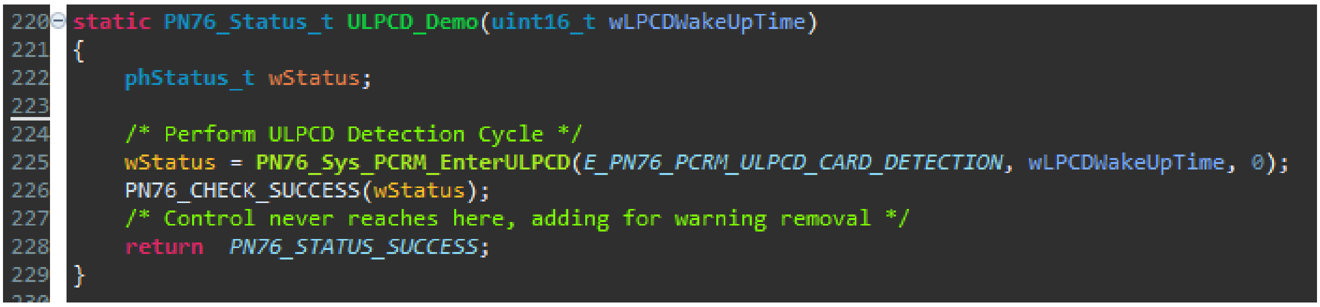
The initialization of parameter "*wULPCDWakeUpTime*" is done by using define. This define can be adjusted to the users liking:



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

Figure 16. Example: Powerdown 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 then "*E_PN76_PCRM_ULPCD_CALIBRATION*" as before:



```
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, 0);
    PN76_CHECK_SUCCESS(wStatus);
    /* Control never reaches here, adding for warning removal */
    return  PN76_STATUS_SUCCESS;
}
```

Figure 17. 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 reasons:

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

3.4.2 Abort with GPIO3

Another option always available to go out of ULPCD mode is abort via GPIO3. As the development board can have multiple configurations the below shown option is not the most elegant, but the one working without caring about the boards 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:

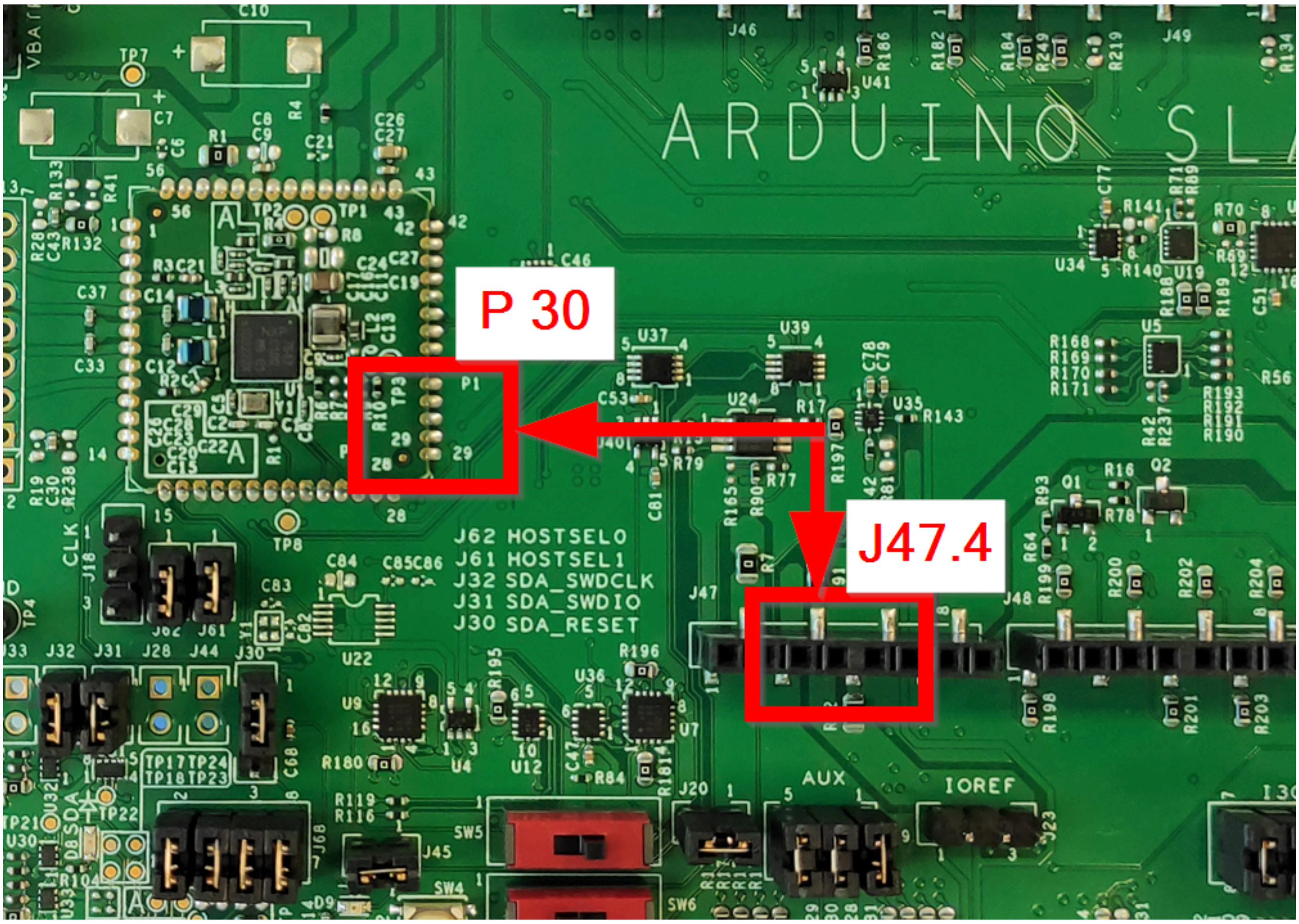


Figure 18. GPIO3 abort wiring

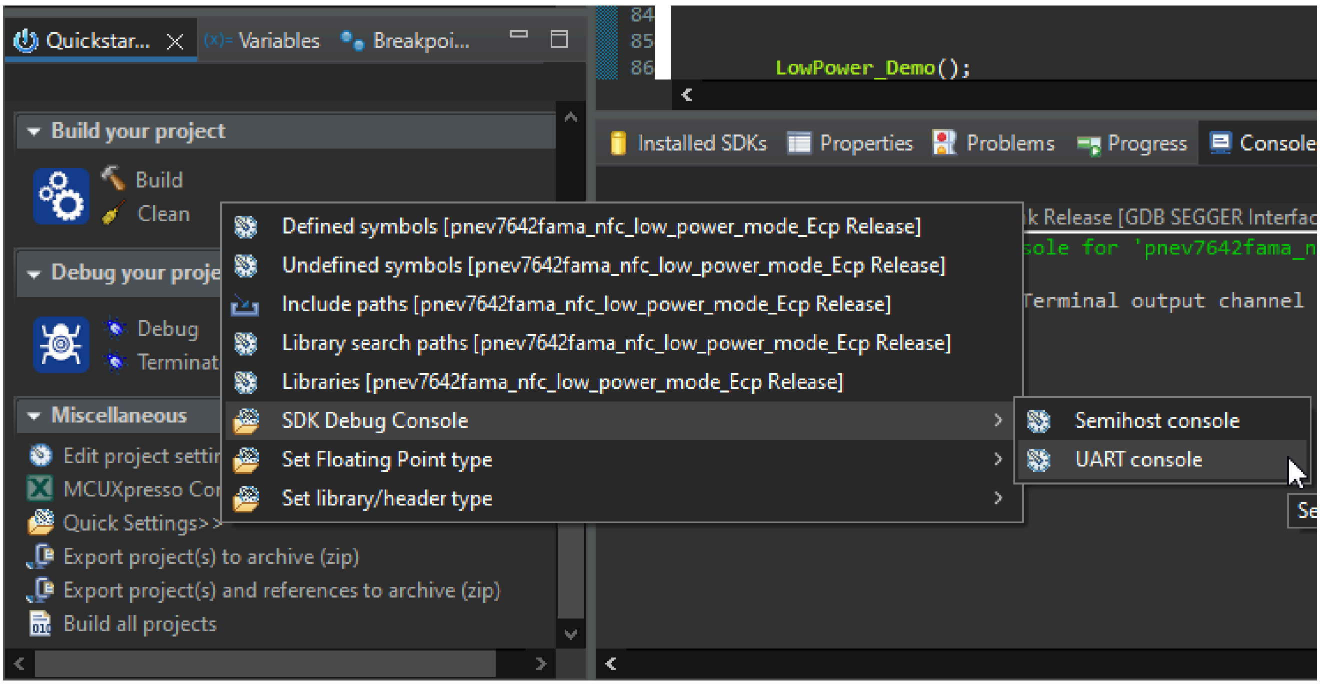
P.30 is the second pin of the right row (above 29).

If the chip is woken up, you should see the green LED on.

4 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 in/output the J-Link RTT Viewer is used. This is more lightweight, easier to reattach.

Either at import of the example you can choose "UART" or "Semihost". While UART represents the RTT Viewer option and Semihost the output in the IDE console. Later you can still switch the debug console at the quick settings: Quick Settings → SDK Debug Console → Semihost/UART

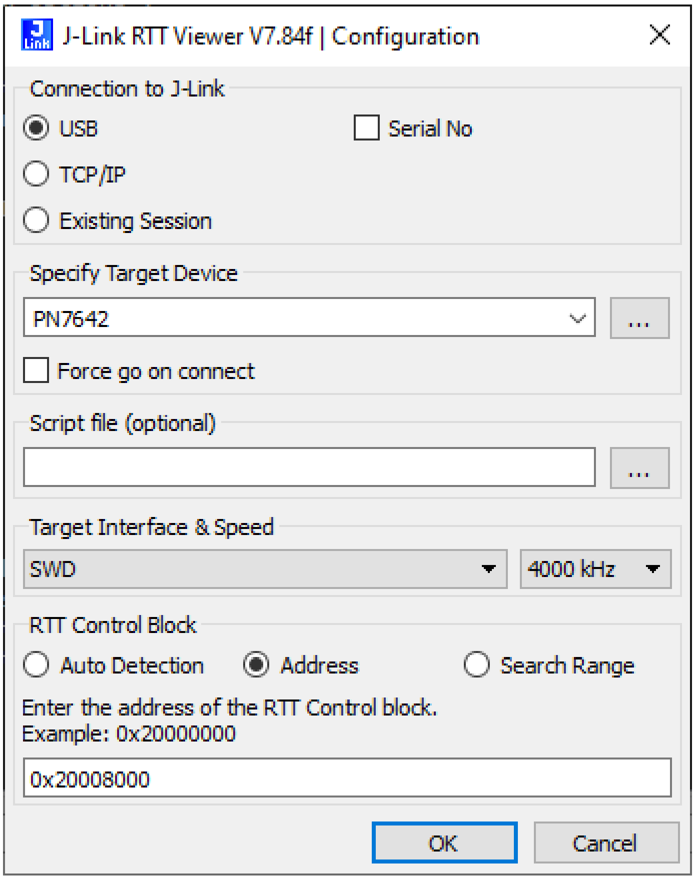


```
        LowPower_Demo();
```

Quickstar...
Variables
Breakpoi...
Build your project
Build
Clean
Debug your proje
Debug
Terminat
Miscellaneous
Edit project settir
MCUXpresso Cor
Quick Settings>>
Export project(s) to archive (zip)
Export project(s) and references to archive (zip)
Build all projects
Installed SDKs
Properties
Problems
Progress
Console
k Release [GDB SEGGER Interfac
sole for 'pnev7642fama_n
Terminal output channel
Defined symbols [pnev7642fama_nfc_low_power_mode_Ecp Release]
Undefined symbols [pnev7642fama_nfc_low_power_mode_Ecp Release]
Include paths [pnev7642fama_nfc_low_power_mode_Ecp Release]
Library search paths [pnev7642fama_nfc_low_power_mode_Ecp Release]
Libraries [pnev7642fama_nfc_low_power_mode_Ecp Release]
SDK Debug Console
Set Floating Point type
Set library/header type
Semihost console
UART console
Se

Figure 19. MCUXpresso Debug Console Option

The configuration to attach to the PN7642 is as below:



J-Link RTT Viewer V7.84f | Configuration

Connection to J-Link

● USB □ Serial No

○ TCP/IP

○ Existing Session

Specify Target Device

PN7642

□ Force go on connect

Script file (optional)

Target Interface & Speed

SWD 4000 kHz

RTT Control Block

○ Auto Detection ● Address ○ Search Range

Enter the address of the RTT Control block.
Example: 0x20000000

0x20008000

Figure 20. J-Link RTT Viewer configuration

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

5 Tips and tricks

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

A: This usually 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 (e.g. LED Blinky).

Else you can try to catch the chip in its active state by trying multiple times. Or hold reset, 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 shall 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 would be to use semihost and make the output of the boot reason with LEDs. So that after the boot up at least you see visually what the boot reason has been.

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.

6 References

[1] PN7642 product data sheet: https://www.nxp.com/doc/PN7642

[2] UM11566 PN76 family NFC open controller user manual, available on www.nxp.com/secure files

[3] PN7642 NFC controller user API documentation

[4] MCUXpresso SDK: https://www.nxp.com/webapp/Download?colCode=PN7642_MCUXPRESSO_SDK&appType=license

[5] PN76 family evaluation board quick start guide: https://www.nxp.com/doc/AN13134

7 Note about the source code in the document

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

8 Legal information

8.1 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.

8.2 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 http://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.

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.

8.3 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.

8.4 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

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

For more information, please visit: http://www.nxp.com