AN11646

PN7120 NFC controller SBC kit quick start guide

Rev. 1.5 — 1 February 2021 | Application note
319715 | COMPANY PUBLIC

Document information

Information	Content
Keywords	OM5577, PN7120, Raspberry Pi, BeagleBone, NFC, P2P, Card Emulation, Linux, Android
Abstract	This document gives a description on how to get started with the PN7120 NFC controller SBC kit.

1 Revision history

Revision history

Rev	Date	Description
1.5	20210201	• Removed Windows IoT support • The format of this application note has been redesigned to comply with the new identity guidelines of NXP Semiconductors
1.4	20190708	Updated Linux demo part with link to instructions
1.3	20160520	Adding details about Beaglebone startup
1.2	20151009	- Information about Win10 IoT demo added - Section Section 7 updated
1.1	20150701	- Correction of a syntax error in a referenced link - Explicitly point to the demo images
1.0	20150601	First release

2 Introduction

This document gives a description on how to get started with the PN7120 NFC-Controller SBC Kit. This document provides a step by step guide to the installation procedure of the hardware and the software. Finally it shows PN7120 NFC Controller functionalities through demonstration application.

2.1 OM5577/PN7120S demo kit

OM5577/PN7120S kit is a high performance fully NFC-compliant expansion board for both Raspberry Pi (refer to [1] for more details) and BeagleBone (refer to [2] for more details). It meets compliance with Reader mode, P2P mode and Card emulation mode standards. The board features an integrated high performance RF antenna to insure high interoperability level with NFC devices.

The kit is composed of 3 printed circuit boards and a MIFARE Ultralight EV1 product-based card.

Figure 1. PN7120 NFC Controller SBC Kit content

2.2 Linux driver support

PN7120 NFC Controller is supported under GNU/Linux system using the NXP Linux libnfc-nci software stack (for more details, refer to AN11697 available on PN7120 Product Web Page [5]). The Raspberry Pi and BeagleBone Linux demo images include the complete stack (Kernel mode driver, User mode library and demo application) allowing to demonstrate the NFC functionalities offered by the PN7120.

2.3 Android driver support

PN7120 NFC Controller is supported from the official Android Open Source Project (refer to [7] for more details) with the addition of dedicated patches available through PN7120 Product Web Page [5] (refer to AN11690).

The BeagleBone Black demo image is based on this concept.

3 Quick Startup on Raspberry Pi

3.1 Required items

• Raspberry Pi [1]

• Compatible SD or MicroSD card (depending on the Raspberry Pi model) of at least 4 Gb memory size [3]

• Micro USB power supply (5 V / 1A) [4]

• USB Keyboard

• USB Mouse

• HDMI cable to connect to a Monitor / TV

• Computer (running Windows, Linux or Mac OS X) for SD/MicroSD card installation

3.2 Hardware setup

First of all assemble the PN7120 NFC Controller Board with the Raspberry Pi Interface Board.

Figure 2. OM5577/PN7120S Raspberry Pi configuration

Then stacked together the boards with the Raspberry Pi according to below guidelines.

3.2.1 Raspberry Pi A/B (old models)

On the old models, the Raspberry Pi Interface Board connector fit perfectly the Raspberry Pi one. Assemble the boards as shown in figure below, removing first the 4 white plastic spacers:

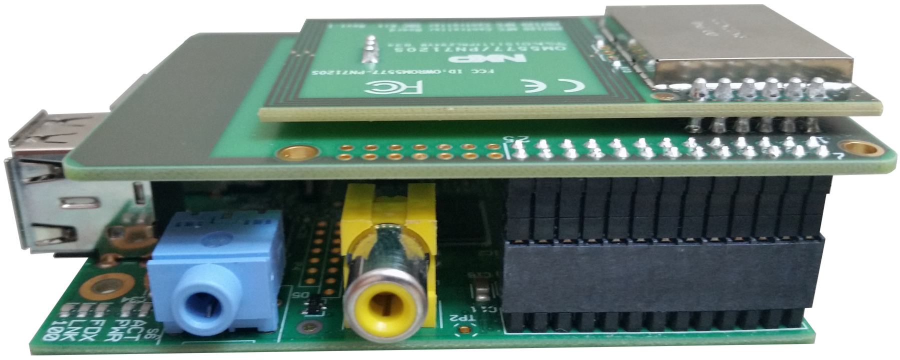

Figure 3. OM5577/PN7120S and Raspberry Pi B model stacked together

3.2.2 Raspberry Pi A+/B+ and Raspberry Pi 2 (new models)

The Raspberry Pi new models have a 40-pin connector allowing to connect an expansion board. The Raspberry Pi Interface Board only makes use of the first 26 ones for compatibility reason with the previous Raspberry Pi models. Assemble the boards as shown in figure below:

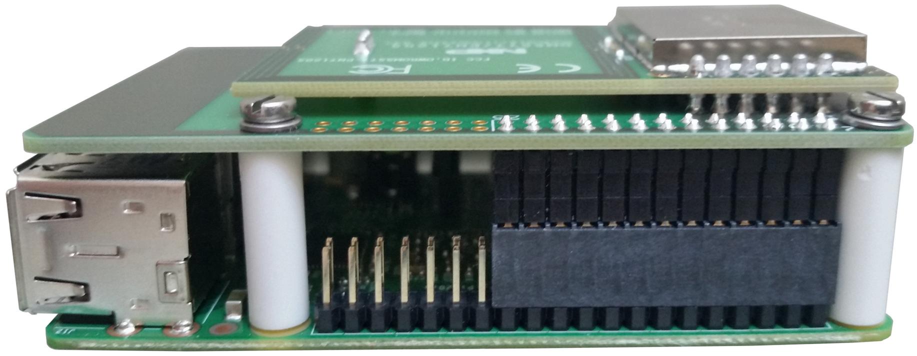

Figure 4. OM5577/PN7120S and Raspberry Pi B+ model stacked together

3.3 Linux NFC demo application

3.3.1 Setup

Guidelines to set up this demonstration are provided here https://community.nxp.com/t5/NXP-Designs-Knowledge-Base/Easy-set-up-of-NFC-on-Raspberry-Pi/ta-p/1099034. Just follow the step-by-step procedure to install the demo from Raspbian distribution.

3.3.2 Application details

The demo application is part of the Linux libnfc-nci stack available on public GitHub repository https://github.com/NXPNFCLinux/linux_libnfc-nci. The related source code can then be found there (more details in document AN11697 [7]).

Refer to Section 5 for the following procedure.

4 Quick Startup on BeagleBone

4.1 Required items

- BeagleBone Black [2]
- MicroSD card of at least 4 Gb (8 Gb for Android)
- 5 V adapter or micro USB cable to power the BeagleBone
- USB Keyboard
- USB Mouse
- USB Hub to connect both Mouse and Keyboard to the BeagleBone
- HDMI cable to connect to a Monitor / TV
- Computer (running Windows, Linux or Mac OS X) for MicroSD card installation
- BeagleBone image file, downloaded from the OM5577/PN7120S demo kit webpage [6]

4.2 Hardware setup

First of all assemble the PN7120 NFC Controller Board with the BeagleBone Interface Board.

Figure 5. OM5577/PN7120S BeagleBone configuration

Then stacked together the boards with the BeagleBone.

Figure 6. OM5577/PN7120S and BeagleBone Black stacked together

4.3 Software setup

Prepare a MicroSD card, with the downloaded BeagleBone demo image (https://www.nxp.com/downloads/en/software/OM5577_BBB_Linux.html or https://www.nxp.com/downloads/en/software/OM5577_BBB_Kitkat.html), following the installation guidelines. First extract the ".img" file from the archive, then flash it on the microSD card according to below guidelines.

4.3.1 On Windows

Insert the MicroSD card into your computer (note the device drive letter), and using Win32 Disk Imager, write the image into it:

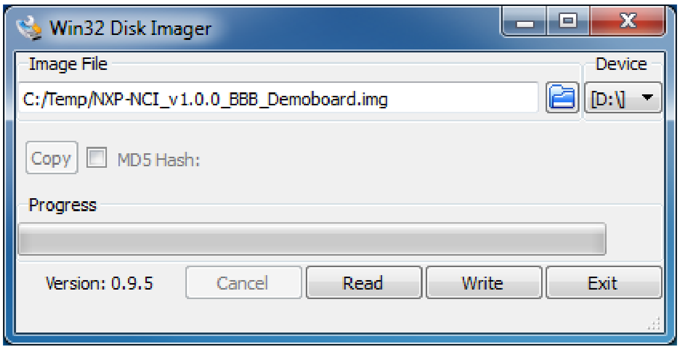

Figure 7. Win32 Disk Imager

4.3.2 On Linux

Insert the MicroSD card into your computer and determine the device node assigned to it (ignore the device number; e.g. /dev/sdb, not sdb1):

```
$ sudo dmesg | tail -20
```

```
$ sudo dmesg | tail -20
[95300.848154] usb 2-1: new high-speed USB device number 33 using ehci-
pci
[95300.983859] usb 2-1: New USB device found, idVendor=14cd, id
Product=6d00
[95300.983872] usb 2-1: New USB device strings: Mfr=1, Product=3,
SerialNumber=2
[95300.983880] usb 2-1: Product: USB 2.0 SD/MMC READER
[95300.983888] usb 2-1: Manufacturer: SDMMC M121
[95300.983895] usb 2-1: SerialNumber: 800340070270
[95300.984593] usb-storage 2-1:1.0: USB Mass Storage device detected
[95300.984882] scsi18 : usb-storage 2-1:1.0
[95301.985555] scsi 18:0:0:0: Direct-Access USB 2.0 SD/MMC Reader PQ:
0 ANSI: 0 CCS
[95301.986856] sd 18:0:0:0: Attached scsi generic sg2 type 0
[95301.988277] sd 18:0:0:0: [sdb] Attached SCSI removable disk
```

Figure 8. Identifying device number under Linux

Then, unmount the device node using following command:

```
sudo umount /dev/devicenode
```

Finally flash the image to the device node using following command:

```
sudo dd if=path_to_image_file.img of=/dev/devicenode bs=1M
```

4.3.3 On MAC OS X

Using PiFiller (see http://www.nxp.com/redirect/learn.adafruit.com/beaglebone-black-installing-operating-systems/mac-os-x.md), select the image file then insert the MicroSD card into your computer to flash it.

4.4 Starting NFC demo

Then power up the Raspberry Pi by plugging the USB power cable.

Insert the MicroSD card in the BeagleBone. Connect HDMI Display, mouse and keyboard. Finally supply the BeagleBone using 5 V adapter or micro USB cable.

This triggers power-up of the BeagleBone, then depending of the demo image used:

4.4.1 Linux image

The Raspberry Pi boots and displays the bone-debian GUI:

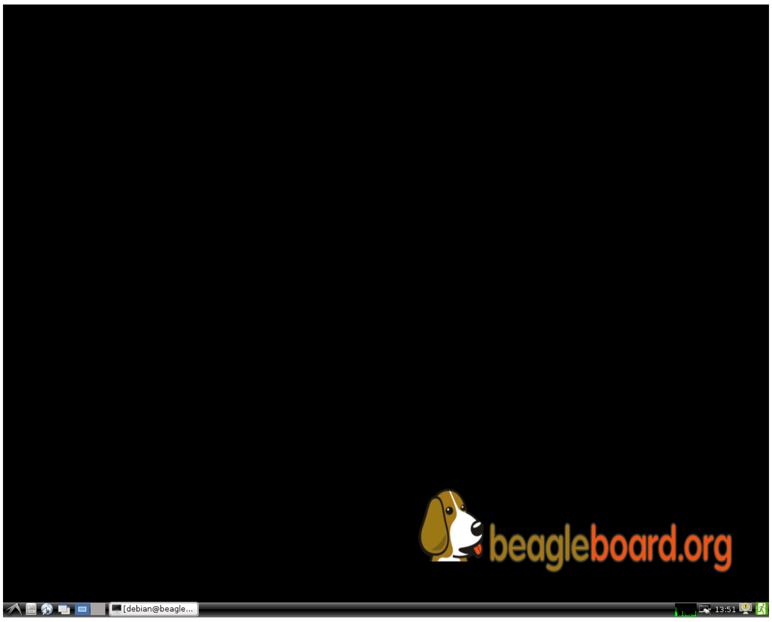

Figure 9. bone-debian GUI

Open a terminal and browse to the Linux libnfc-nci stack directory (refer to Section 2.2 for more details about the Linux NFC software stack).

```
$ cd ~/linux_libnfc-nci
```

Refer to Section 5 for the following procedure.

4.4.2 Android image

After a few seconds Android boots up, NFC is then running, ready to read tags or interact with remote NFC device (e.g. NFC phone).

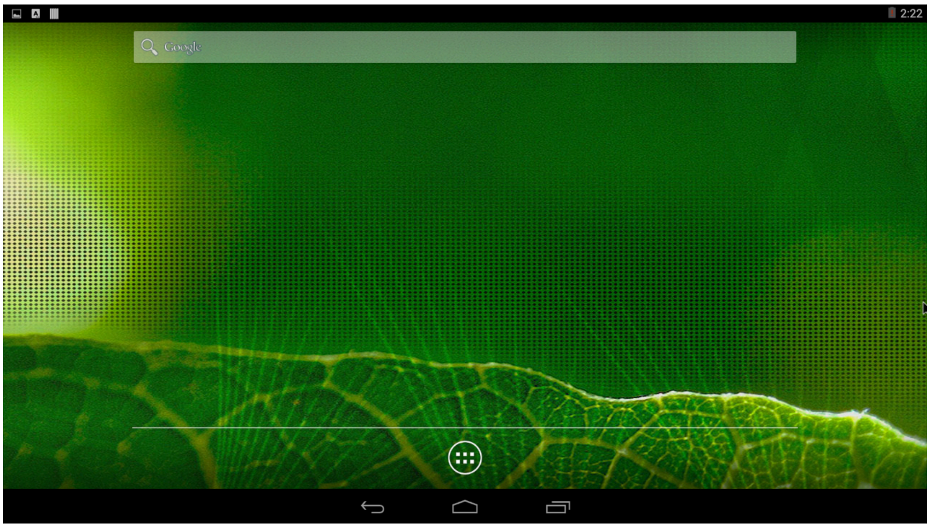

Figure 10. Android home screen

You can enable/disable the NFC function via “Settings/Wireless & Network/More…”

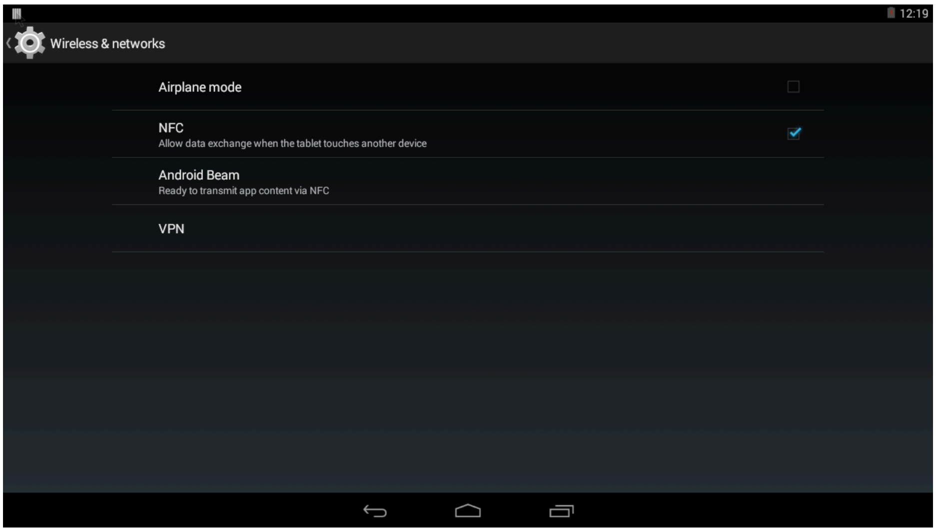

Figure 11. Android “Setting/Wireless&Network” menu

Using provided NXP TagInfo and NXP TagWriter applications you can get information from discovered tag and write content.

Figure 12. Android applications

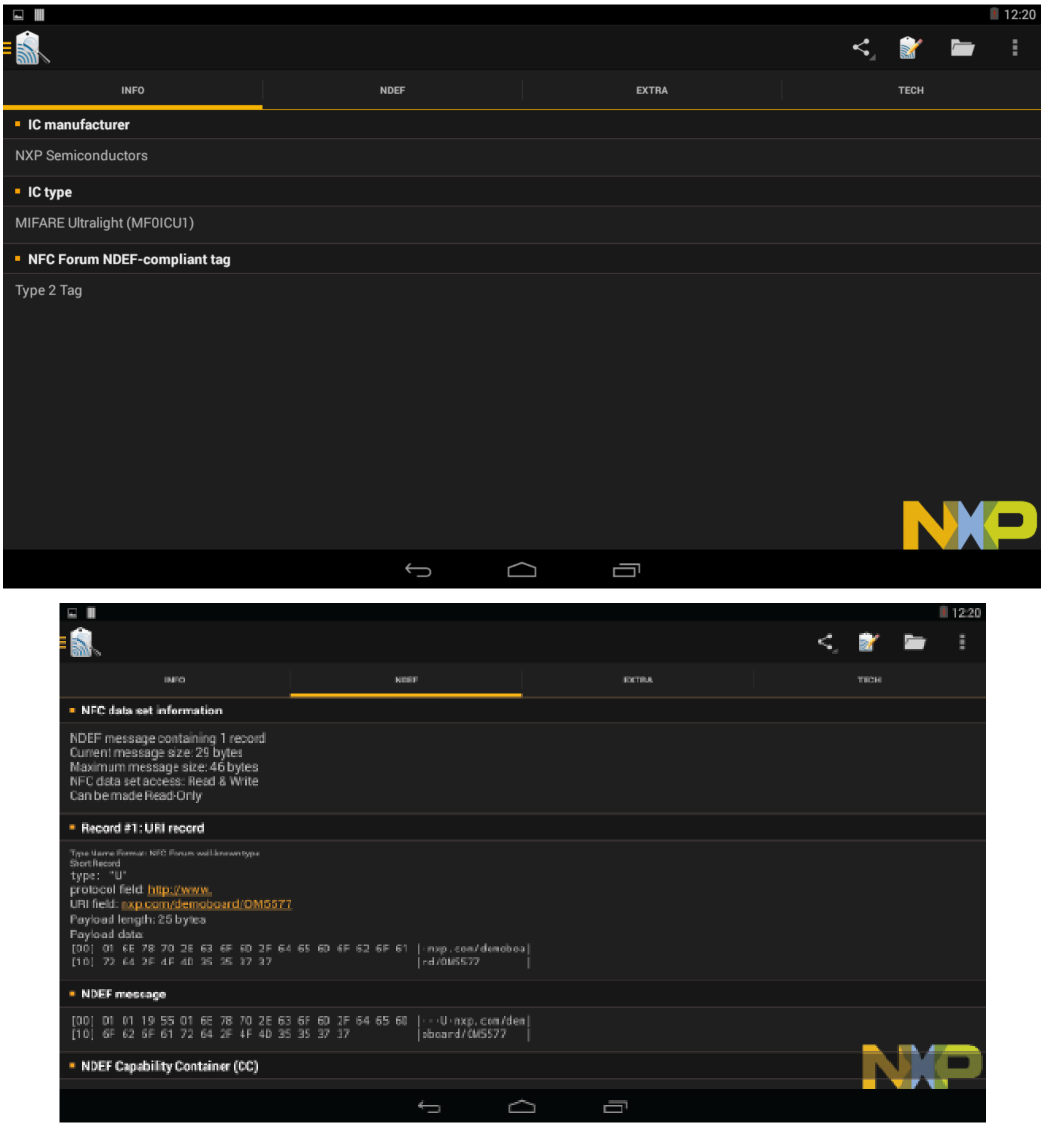

Figure 13. Android TagInfo application

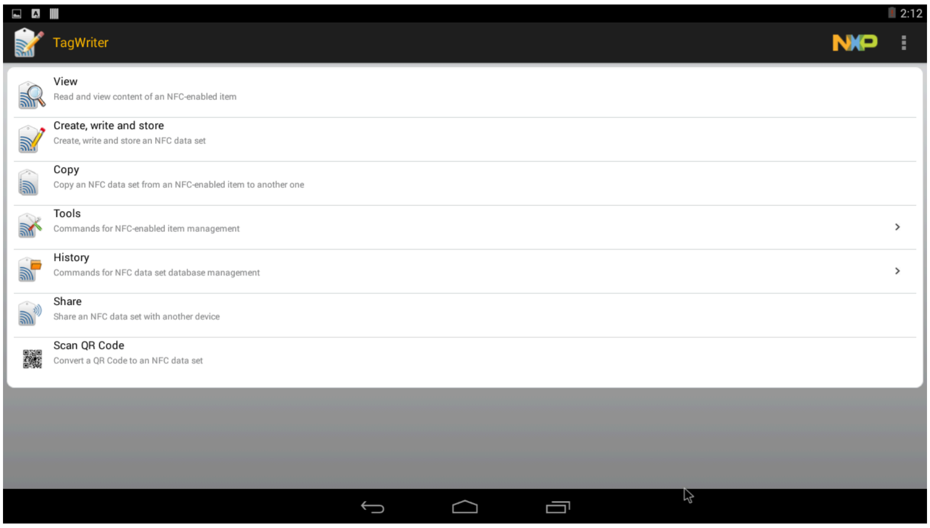

Figure 14. Android TagWriter application

5 Linux NFC demo application

5.1 Application details

The demo application is part of the Linux libnfc-nci stack delivery. More details can be found in document AN11697 [7] available on PN7120 product web page [5].

5.2 Using the application

The application must be started with parameters:

```
$ ./nfcDemoApp <OPTIONS>
```

You can get the parameters details by launching the application help menu:

```
$ ./nfcDemoApp write --help
```

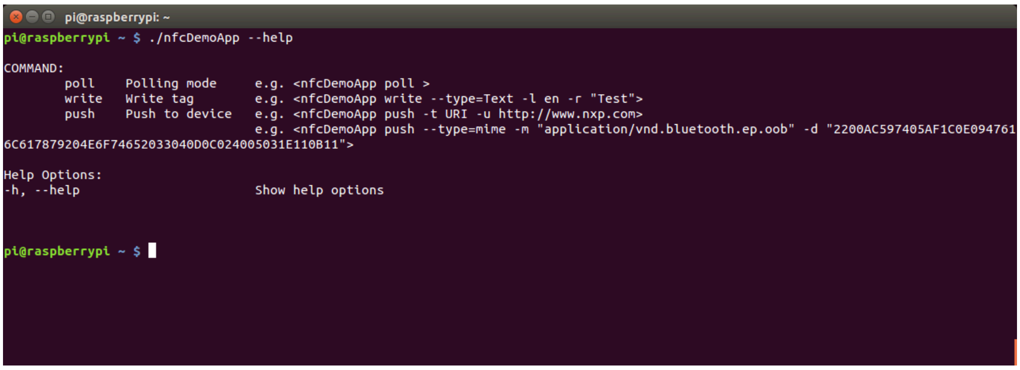

Figure 15. Linux demo application commands

The demo application offers 3 modes of operation:

- **Polling**: continuously waiting for a remote NFC device (tag or peer device) and displays related information
- **Tag writing**: allows writing NDEF content to an NFC tag
- **Device push**: allows pushing NDEF content to a remote NFC peer device

5.2.1 Polling mode

When in this mode, the application displays information of any discovered NFC tags or remote NFC device.

It is reached starting the application with "poll" parameter:

```
$ ./nfcDemoApp poll
```

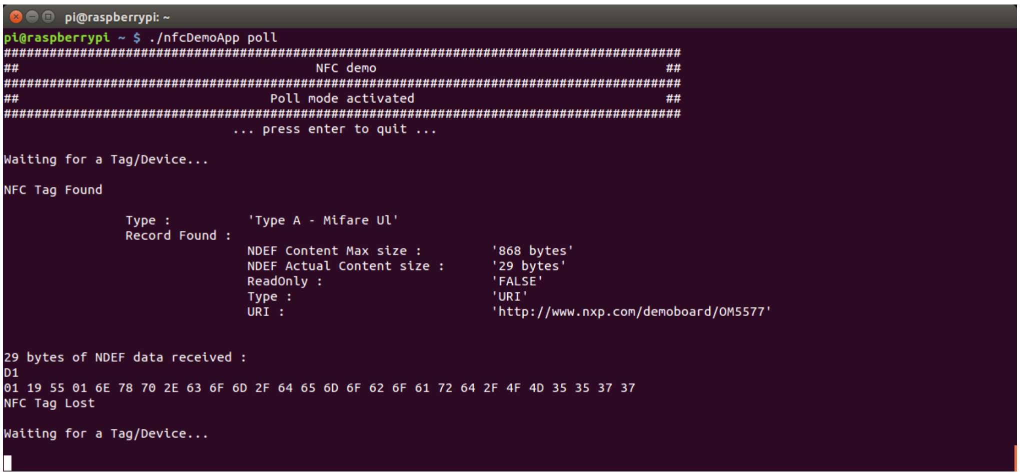

Figure 16. Linux demo application polling mode

5.2.2 Tag writing mode

This mode allows writing data to an NFC tag. It is reached using “write” parameter:

```
$ ./nfcDemoApp write <OPTIONS>
```

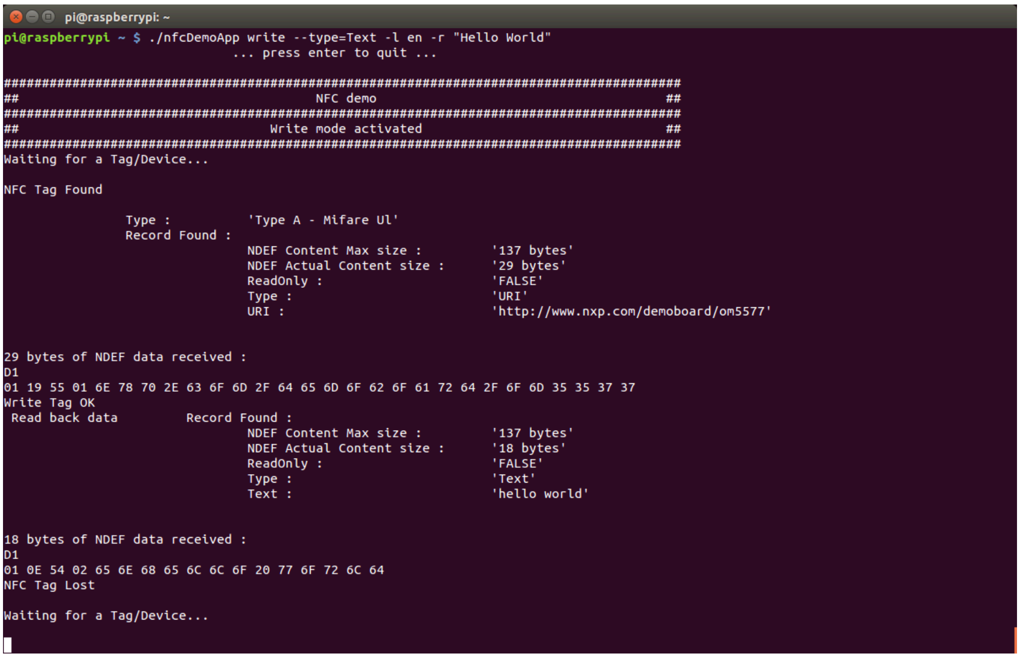

Figure 17. Linux demo application tag writing mode

You can get more information about the message format using “-h” or “--help” parameter:

```
$ ./nfcDemoApp write --help
```

5.2.3 Device push mode

This mode allows pushing data to a remote NFC device (e.g. an NFC phone). It is reached using “push” parameter:

```
$ ./nfcDemoApp push <OPTIONS>
```

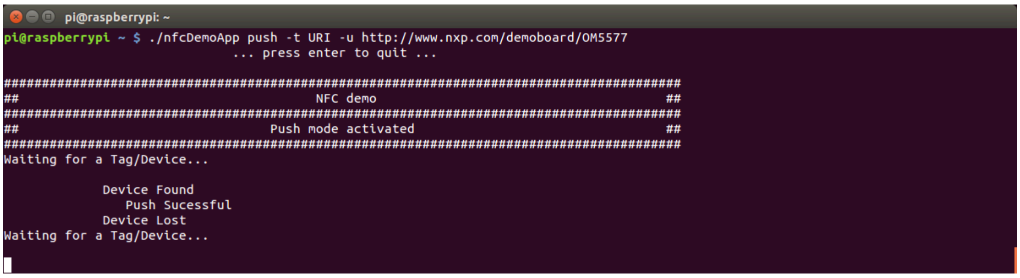

Figure 18. Linux demo application device push mode

You can get more information about the message format using “-h” or “--help” parameter:

```
$ ./nfcDemoApp push --help
```


6 References

[1] The Raspberry Pi is a credit card sized computer. The initial idea behind it was to develop a small and cheap computer to be used by kids all over the world to learn programming. In the end, it became very popular among developers all over the world. The heart of the Raspberry Pi is a SoC (System on Chip). This contains an ARM11 running at 700 MHz and a graphics processor that is capable of Blu-ray quality playback, using H.264 at 40MBits/s. It has a fast 3D core accessed using the supplied OpenGL ES2.0 and Open VG libraries. In addition, the Model B has 512 MB RAM included in its SoC. To get started quickly, the Raspberry Pi Foundation provides several preconfigured Linux distributions.
For more information about it please visit https://www.raspberrypi.org/.

[2] BeagleBone is a low-power open source hardware single-board credit-card-sized Linux computer that connects to the Internet and runs software such as Android and Ubuntu. With plenty of I/O and processing power for real-time analysis provided by a 720 MHz ARM® processor-based SoC (System on Chip), BeagleBone can be complemented with cape plug-in boards to augment functionality.
For more information about it, please visit http://beagleboard.org/bone.

[3] List of verified SD cards: https://elinux.org/RPi_SD_cards.

[4] List of verified USB power adapters: https://elinux.org/RPi_VerifiedPeripherals#Power_adapters

[5] PN7120 Product Web Page: https://www.nxp.com/products/rfid-nfc/nfc-hf/nfc-readers/nfc-controller-with-integrated-firmware-and-nci-interface-for-home-appliances:PN7120

[6] OM5577/PN7120S demo kit webpage: https://www.nxp.com/products/rfid-nfc/nfc-hf/nfc-readers/development-kits-for-pn7120-plugn-play-nfc-controller:OM5577

[7] AN11697 PN71x0 Linux Software Stack Integration Guidelines: https://www.nxp.com/docs/en/application-note/AN11697.pdf

7 Legal information

7.1 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.

7.2 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.

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.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages. Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Translations — A non-English (translated) version of a 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 or documented vulnerabilities. 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.

7.3 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.

7.4 Trademarks

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

MIFARE — is a trademark of NXP B.V.

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

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

Figures

Contents

Date of release: 1 February 2021
Document identifier: AN11646
Document number: 319715