

STMicroelectronics UM2375 Linux Driver User Manual

[Home](#) » [STMicroelectronics](#) » [STMicroelectronics UM2375 Linux Driver User Manual](#) 

Contents

[1 STMicroelectronics UM2375 Linux Driver User Manual](#)

- [1.1 Overview](#)
 - [1.1.1 Features](#)
 - [1.1.2 Software architecture](#)
- [1.2 Hardware setup](#)
 - [1.2.1 Platform used](#)
 - [1.2.2 Hardware requirements](#)
- [1.3 Linux environment setup](#)
 - [1.3.1 Booting of Raspberry Pi](#)
 - [1.3.2 Enable SPI on Raspberry Pi](#)
- [1.4 Build of RFAL library and application](#)
- [1.5 How to run the application](#)
- [1.6 Revision history](#)
- [1.7 List of tables](#)
- [1.8 List of figures](#)
- [2 Documents / Resources](#)
 - [2.1 References](#)
- [3 Related Posts](#)

STMicroelectronics UM2375 Linux Driver User Manual



Linux® driver for the ST25R3911B and ST25R3912/14/15 high performance NFC frontends

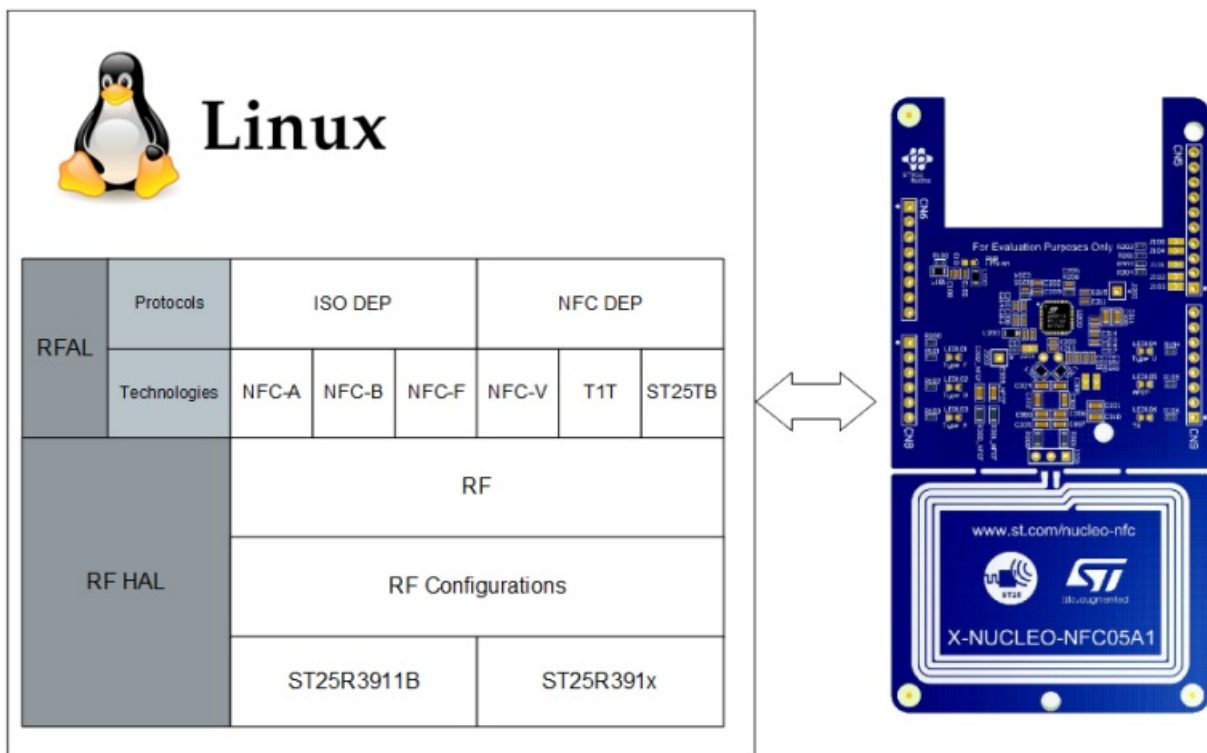
Introduction

The STSW-ST25R009 Linux® driver enables the Raspberry Pi 4 to operate with the X-NUCLEO-NFC05A1, which contains the ST25R3911B high performance NFC universal device.

This package ports the RF abstraction layer (RFAL) onto a Raspberry Pi 4 Linux platform to operate with X-NUCLEO-NFC05A1 firmware. The package provides a sample application detecting different types of NFC tags and mobile phones supporting P2P. The RFAL is the ST standard driver for ST25R NFC/RFID Reader ICs ST25R3911B, ST25R3912, ST25R3913, ST25R3914 and ST25R3915. It is used, for instance, by the ST25R3911B-DISCO firmware (STSW-ST25R002) and by the X-NUCLEONFC05A1 firmware (X-CUBE-NFC5).

STSW-ST25R009 supports all the ST25R3911B lower-layer protocols and also some higher layer protocols for communication. The RFAL is written in a portable manner, so it can run on a wide range of devices based on Linux®. This document describes how the RFAL library can be used on a standard Linux system (in this case the Raspberry Pi 4) for NFC/RF communication. The code is highly portable and works with minor changes on any Linux platform.

Figure 1. RFAL library on Linux platform



Overview

Features

- Complete Linux user space driver (RF abstraction layer) to build NFC enabled applications using the ST25R3911B/ST25R391x high performance NFC frontends with up to 1.4 W output power
- Linux host communication with the ST25R3911B/ST25R391x using SPI interface
- Complete RF/NFC abstraction (RFAL) for all major technologies and higher layer protocols:
 - NFC-A (ISO14443-A)
 - NFC-B (ISO14443-B)
 - NFC-F (FeliCa™)
 - NFC-V (ISO15693)
 - P2P (ISO18092)

- ISO-DEP (ISO data exchange protocol, ISO14443-4)
- NFC-DEP (NFC data exchange protocol, ISO18092)
- Proprietary technologies (Kovio, B', iClass, Calypso®, ...)
- Sample implementation available with the X-NUCLEO-NFC05A1 expansion board, plugged into a Raspberry Pi 4
- Sample application to detect several NFC tag types and mobile phones supporting P2P
- Free user-friendly license terms

Software architecture

Figure 2 shows the software architecture details of RFAL library on a Linux® platform.

The RFAL is easily portable to other platforms by adapting the so-called platform files.

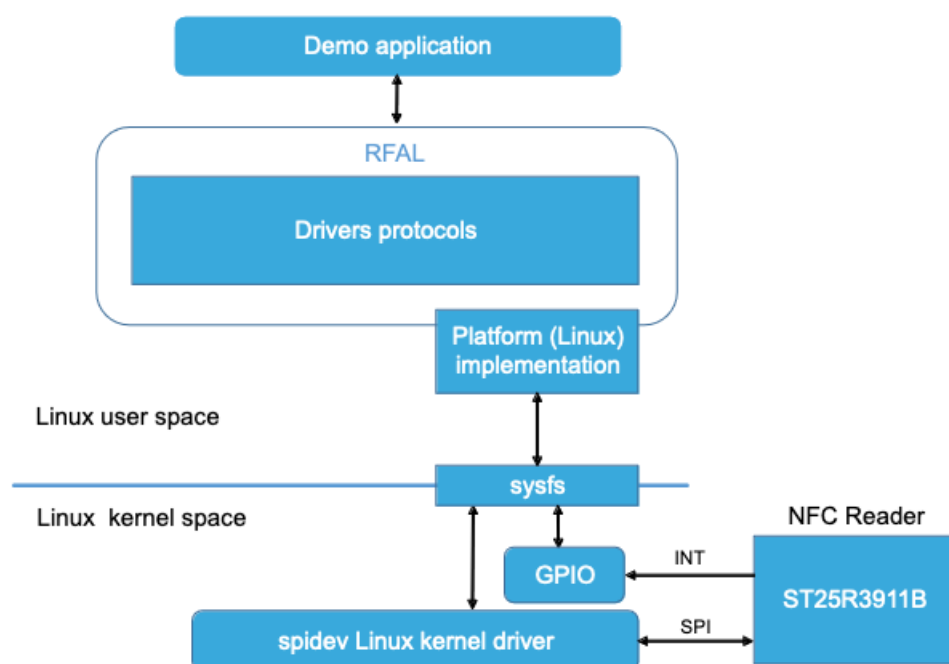
The header file *rfal_platform.h* contains macro definitions, which need to be provided and implemented by the platform owner. Furthermore, it provides platform specific settings like GPIO assignment, system resources, locks and IRQs, which are required for correct operation of the RFAL.

This demonstration implements the platform functions and provides a port of the RFAL library into user space of Linux®. A shared library file is generated, which is used by the demo application to showcase the functionalities provided by the RFAL layer.

Linux® host uses sysfs interface available from Linux® user space to enable SPI communication with the ST25R3911B device. Inside the Linux® kernel the SPI sysfs interface uses Linux® kernel driver spidev to send/receive SPI frames to/from ST25R3911B.

For handling the interrupt line of ST25R3911B, the driver uses the libgpiod to get notified of changes on this line.

Figure 2. RFAL software architecture on Linux



Hardware setup

Platform used

A Raspberry Pi 4 board with Raspberry Pi OS is used as Linux platform to build RFAL library and interact with ST25R3911B over SPI.

ST25R3911B enables an application on Linux platform to detect and communicate with NFC devices.

Hardware requirements

- Raspberry Pi 4
- 8 GBytes micro SD card to boot Raspberry Pi OS
- SD card reader
- Bridge board to connect X-NUCLEO-NFC05A1 with Raspberry Pi Arduino Adapter for Raspberry Pi, part number ARPI600.
- X-NUCLEO-NFC05A1. Refer to latest Raspberry Pi OS requirements.

Hardware connections

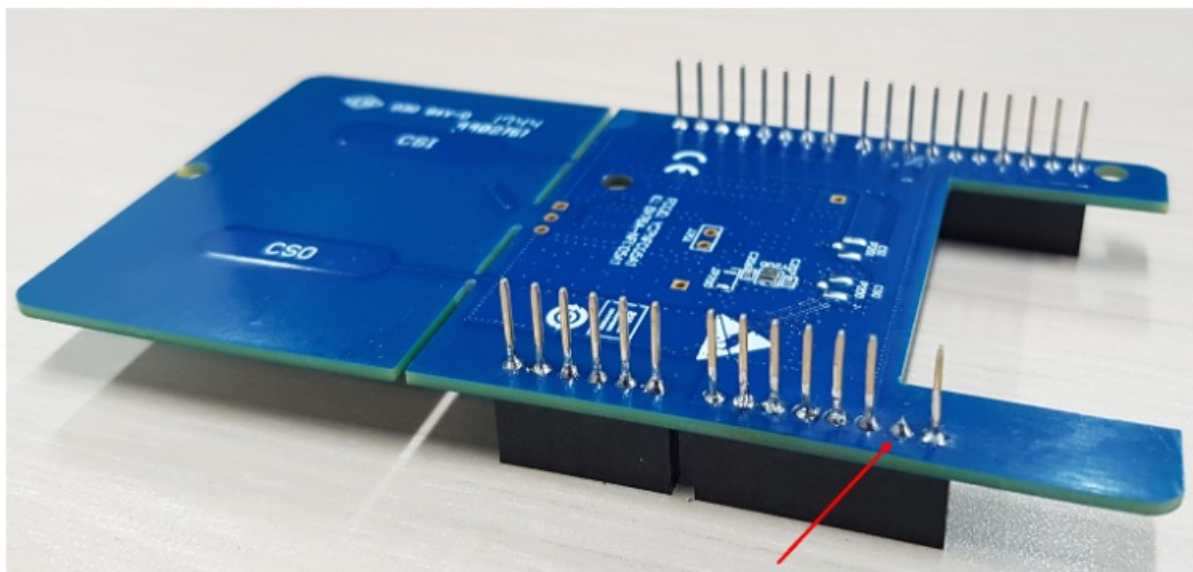
The ARPI600 Raspberry Pi to Arduino adapter board is used to connect the X-NUCLEO-NFC05A1 with the Raspberry Pi. It is required to modify the jumpers of adapter board to connect it with the X-NUCLEO-NFC05A1.

Caution: The ARPI600 incorrectly supplies 5 V to the Arduino IOREF pin. Directly attaching the X-NUCLEO-NFC05A1 feeds back 5 V on some pins, this may damage the Raspberry Pi board. There are reports especially of Raspberry Pi 4B+ actually being destroyed. To avoid this situation adapt either the ARPI600 (a rather difficult operation) or the X-NUCLEO-NFC05A1 (easier operation).

The easiest fix is to cut the CN6.2 (IOREF) pin on the X-NUCLEO-NFC05A1 as shown in Figure 3.

Cutting this pin does not affect the operation in conjunction with Nucleo boards (NUCLEO-L474RG, NUCLEO-F401RE, NUCLEO-8S208RB, etc.).

Figure 3. Hardware connection fix

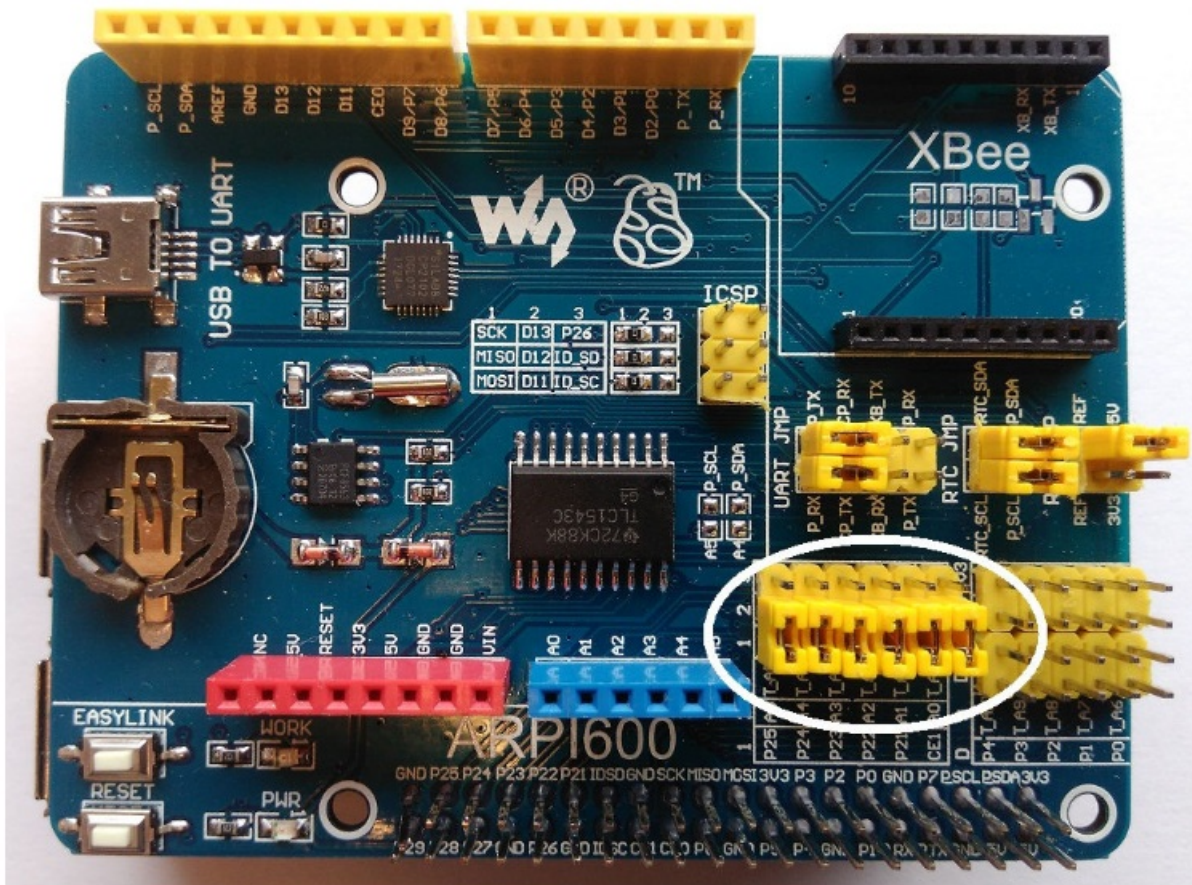


DT61615V1

Jumper setting

The jumpers for A5, A4, A3, A2, A1 and A0 shown in Figure 4 have to be changed to P23, P22, P21 and CE1. With these jumper setting, Raspberry's GPIO pin number 7 is used as interrupt line for X-NUCLEO-NFC05A1.

Figure 4. Position of jumpers A5, A4, A3, A2, A1 and A0 on the adapter board

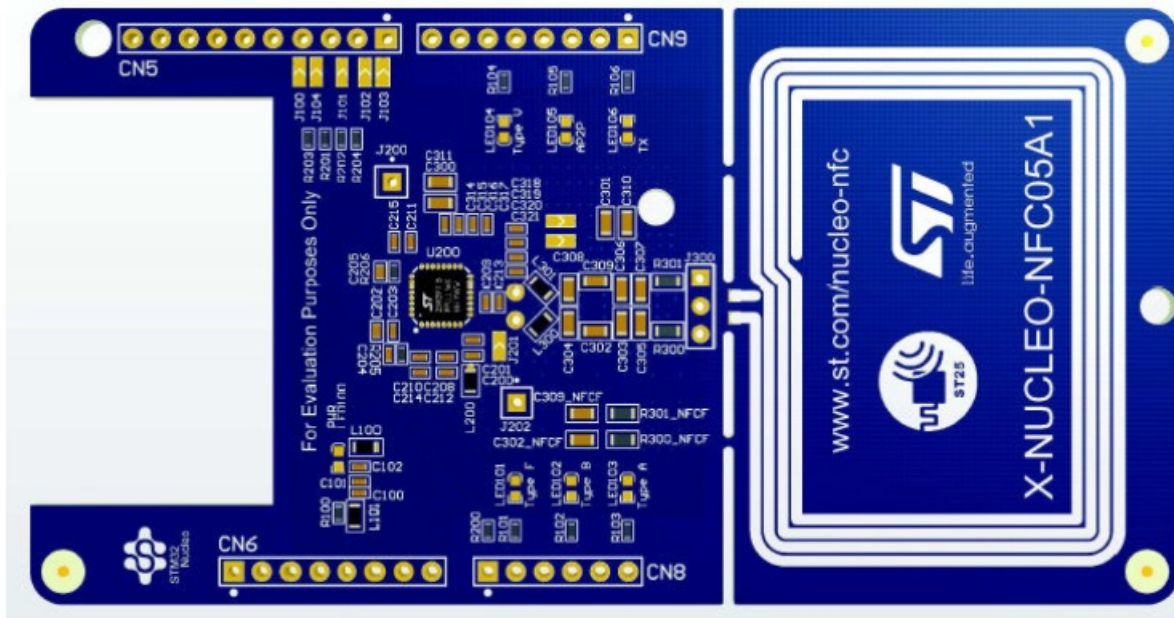


DT48978V2

Currently, this RFAL library port uses the pin GPIO7 as the interrupt line, according to the jumper settings. If there is a requirement to change the interrupt line from GPIO7 to a different GPIO, the platform specific code (in file *pltf_gpio.h*) needs to be modified to change the definition of macro "ST25R_INT_PIN" from 7 to the new GPIO pin, to be used as interrupt line.

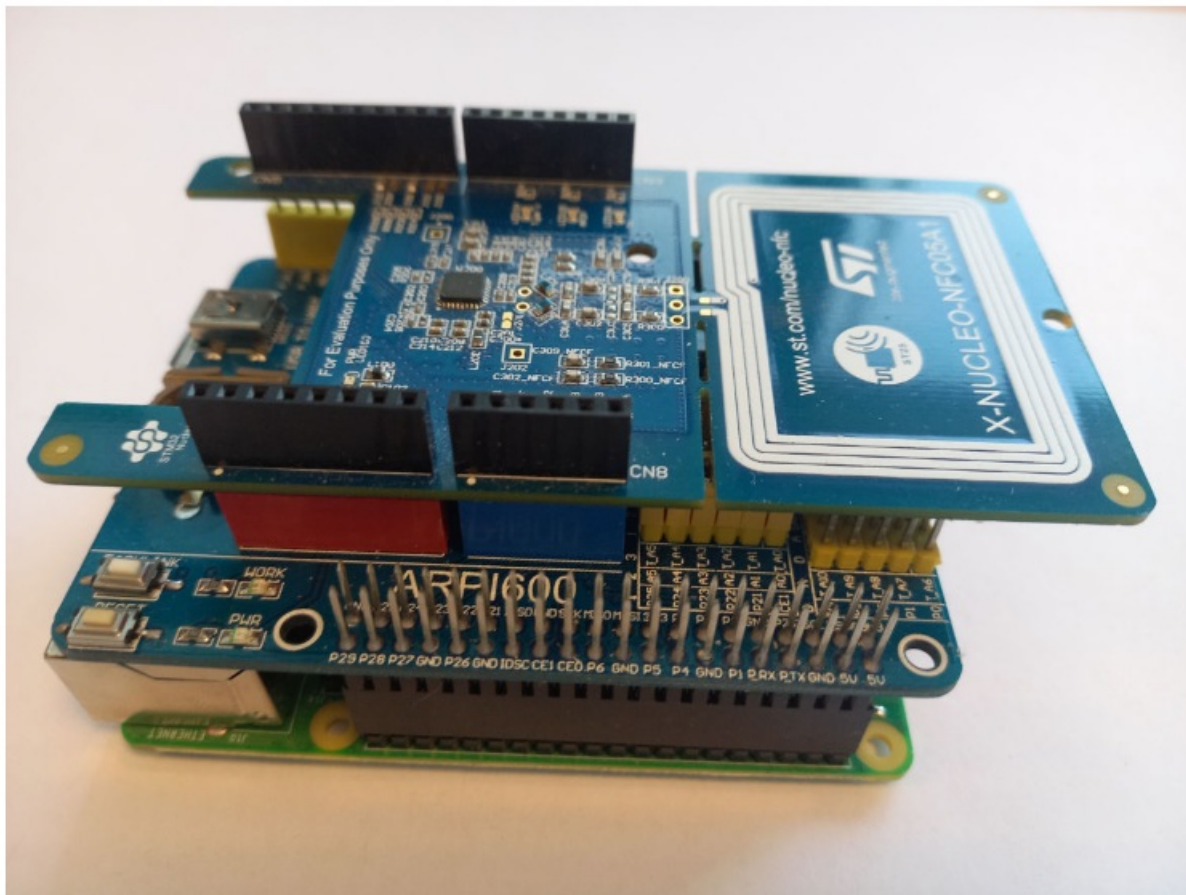
With the above jumper settings, the adapter board can be used to connect X-NUCLEO-NFC05A1 with Raspberry Pi board as shown in Figure 5.

Figure 5. Hardware setup top view



DT48979V2

Figure 6. Hardware setup side view



DT48980V2

Linux environment setup

Bootling of Raspberry Pi

To setup the Linux environment, first step is to install and boot the Raspberry Pi 4 with Raspberry Pi OS as explained below:

Step 1

Download the latest Raspberry Pi OS image from the link:

- <https://www.raspberrypi.com>

Choose Raspberry Pi OS with desktop. For the tests below the following version was used: September 2022 (2022-09-22-raspbian-bullseye-armhf.img.xz).

Step 2

Unzip the Raspberry Pi image and write it into the SD card by following the instructions available in the section named "Writing an image to the SD card".

Step 3

Connect the hardware:

- Connect the Raspberry Pi to a monitor using a standard HDMI cable.
- Connect mouse and keyboard to Raspberry Pi's USB ports.

It is also possible to work with Raspberry Pi using ssh. In that case it is not required to connect the monitor, keyboard and mouse with Raspberry Pi. The only requirement is to have the PC with ssh inside the same network as the Raspberry Pi and configure the IP address accordingly.

Step 4

Boot the Raspberry Pi with SD card.

After booting, a Debian based Linux desktop appears on monitor.

Note: Sometimes, it is observed that after booting Raspberry Pi, some keyboard keys do not work. To make them work, open the file /etc/default/keyboard and set XKBLAYOUT="us" and reboot the Raspberry Pi.

Enable SPI on Raspberry Pi

The SPI driver inside the kernel communicates with X-NUCLEO-NFC05A1 via SPI. It is important to check if SPI is already enabled in the Raspberry Pi OS/kernel configuration.

Check if /dev/spidev0.0 is visible in the Raspberry Pi environment. If it is not visible, enable the SPI interface using the utility "raspi-config" by following the steps described below.

Step 1

Open a new terminal on the Raspberry Pi and run the command "raspi-config" as root:

```
sudo raspi-config
```

This step opens a graphical interface.

Step 2

Select in the graphical interface the option named “Interfacing Options”.

Step 3

This step lists various options.

Select the option named “SPI”.

A new window appears with following text:

“Would you like the SPI interface to be enabled?”

Step 4

Select <Yes> in this window to enable SPI.

Step 5

Reboot Raspberry Pi.

The above steps will enable the SPI interface in Raspberry Pi environment after a reboot.

Build of RFAL library and application

The RFAL demo of Linux is provided in an archive. Let's assume its name is:

ST25R3911B_v2.8.0_Linux_demo_v1.0.tar.xz.

To build the RFAL library and application on Raspberry Pi, follow the below steps:

Step 1

Unzip the package on Raspberry Pi using the below command from the home directory:

```
tar -xJvf ST25R3911B_v2.8.0_Linux_demo_v1.0.tar.xz
```

Step 2

If not done before, install cmake, using below command:

```
apt-get install cmake
```

RFAL library and application build system is based on cmake, for this reason it is required to install cmake for compilation of the package.

Step 3

To build the RFAL library and application, go to the “build” directory:

```
cd ST25R3911B_v2.8.0_Linux_demo_v1.0/Linux_demo/build
```

and run the below command from there:

```
cmake ..
```

In the above command “..” indicates that top level CMakeLists.txt exists in the parent directory, i.e.

ST25R3911B_v2.8.0_Linux_demo_v1.0.

The cmake command creates the makefile that is used in the next step to build the library and application.

Step 4

Run the “make” command to build the RFAL library and application:

```
make
```

The “make” command first builds the RFAL library and then builds the application on top of it.

How to run the application

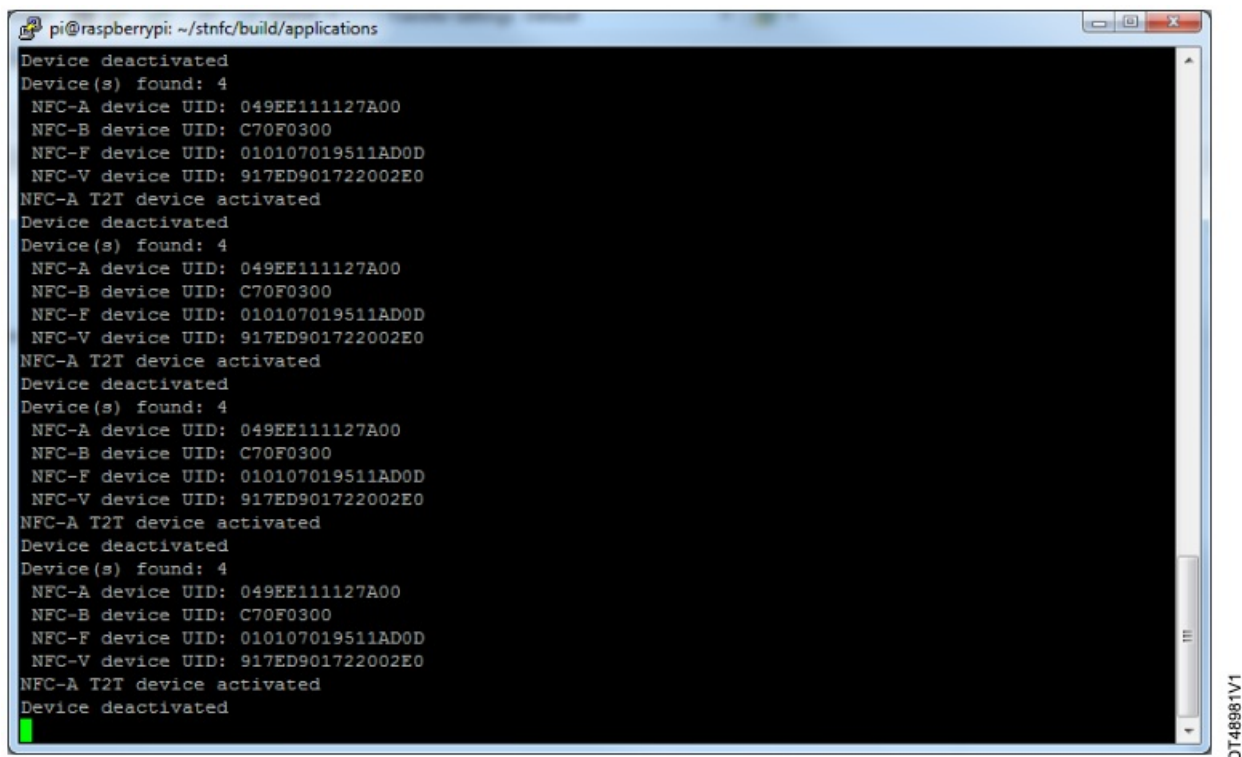
Successful build-up generates an executable named “nfc_demo_st25r3911b” at the following location:
/build/applications.

By default the application needs to be run with root rights from the path:
ST25R3911B_v2.8.0_Linux_demo_v1.0/linux_demo/build:

```
sudo ./demo/nfc_demo_st25r3911b
```

The application starts to poll for NFC tags and mobile phones. It displays the found devices with their UID as shown in Figure 7.

Figure 7. Display of found devices



```
pi@raspberrypi: ~/stnfc/build/applications
Device deactivated
Device(s) found: 4
NFC-A device UID: 049EE111127A00
NFC-B device UID: C70F0300
NFC-F device UID: 010107019511AD0D
NFC-V device UID: 917ED901722002E0
NFC-A T2T device activated
Device deactivated
Device(s) found: 4
NFC-A device UID: 049EE111127A00
NFC-B device UID: C70F0300
NFC-F device UID: 010107019511AD0D
NFC-V device UID: 917ED901722002E0
NFC-A T2T device activated
Device deactivated
Device(s) found: 4
NFC-A device UID: 049EE111127A00
NFC-B device UID: C70F0300
NFC-F device UID: 010107019511AD0D
NFC-V device UID: 917ED901722002E0
NFC-A T2T device activated
Device deactivated
Device(s) found: 4
NFC-A device UID: 049EE111127A00
NFC-B device UID: C70F0300
NFC-F device UID: 010107019511AD0D
NFC-V device UID: 917ED901722002E0
NFC-A T2T device activated
Device deactivated
Device(s) found: 4
NFC-A device UID: 049EE111127A00
NFC-B device UID: C70F0300
NFC-F device UID: 010107019511AD0D
NFC-V device UID: 917ED901722002E0
NFC-A T2T device activated
Device deactivated
```

To terminate the application press Ctrl + C.

Revision history

Table 1. Document revision history

Date	Revision	Changes
09-Mar-2018	1	Initial release.
11-Dec-2018	2	Updated: <ul style="list-style-type: none">Section 2.2.1 Hardware connections Added: <ul style="list-style-type: none">Figure 3. Hardware connection fix
01-Apr-2023	3	Replaced Raspberry Pi 3 with Raspberry Pi 4. Updated: <ul style="list-style-type: none">Introduction, Section 1.1 Features, Section 1.2 Software architecture, Section 2.2 Hardware requirements, Section 2.2.1 Hardware connections, Section 3.1 Booting of Raspberry Pi, Section 4 Build of RFAL library and application, Section 5 How to run the applicationFigure 4. Position of jumpers A5, A4, A3, A2, A1 and A0 on the adapter board, Figure 5. Hardware setup top view, Figure 6. Hardware setup side view

List of tables

Table 1. Document revision history 11

List of figures

Figure 1. RFAL library on Linux platform 1

Figure 2. RFAL software architecture on Linux 3

Figure 3. Hardware connection fix 4

Figure 4. Position of jumpers A5, A4, A3, A2, A1 and A0 on the adapter board 5

Figure 5. Hardware setup top view 6

Figure 6. Hardware setup side view 6

Figure 7. Display of found devices 10

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers’ products.


No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.



ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

Documents / Resources

	<p>STMicroelectronics UM2375 Linux Driver [pdf] User Manual</p> <p>UM2375 Linux Driver, UM2375, Linux Driver, Driver</p>
---	--

References

-  [STMicroelectronics: Our technology starts with you](#)
-  [STMicroelectronics Trademark List - STMicroelectronics](#)