UM2766

User manual

Getting started with X-LINUX-NFC5 package for developing NFC/RFID reader applications on Linux OS

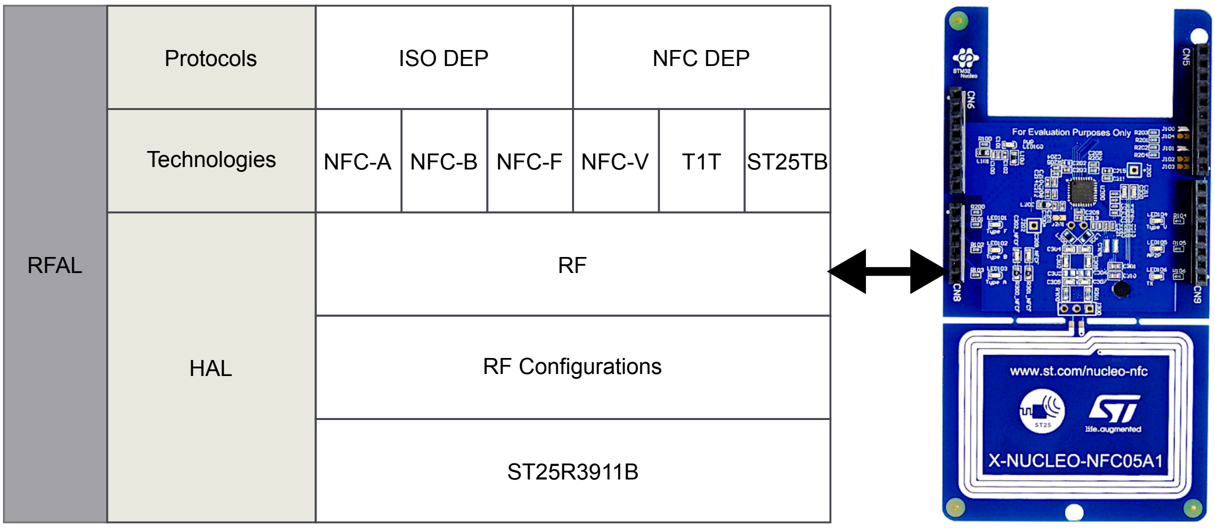
Introduction

This STM32 MPU OpenSTLinux software expansion package demonstrates how you can develop NFC/RF communication for a standard Linux system using our Radio Frequency Abstraction Library (RFAL). The RFAL common interface driver ensures that user function and application software is compatible with any ST25R NFC/RFID reader IC.

The X-LINUX-NFC5 package ports the RFAL onto a Discovery Kit with STM32MP1 Series microprocessor running Linux to drive an ST25R3911B NFC front end on an STM32 Nucleo expansion board. The package includes a sample application to help you understand detection of different types of NFC tags and mobile phones supporting P2P.

The source code is designed for portability across a wide range of processing units running Linux and supports all lower layers and some higher layer protocols of ST25R ICs to abstract RF communication.

Figure 1. Radio Frequency Abstraction Library for Linux



1 X-LINUX-NFC5 overview

1.1 Main features

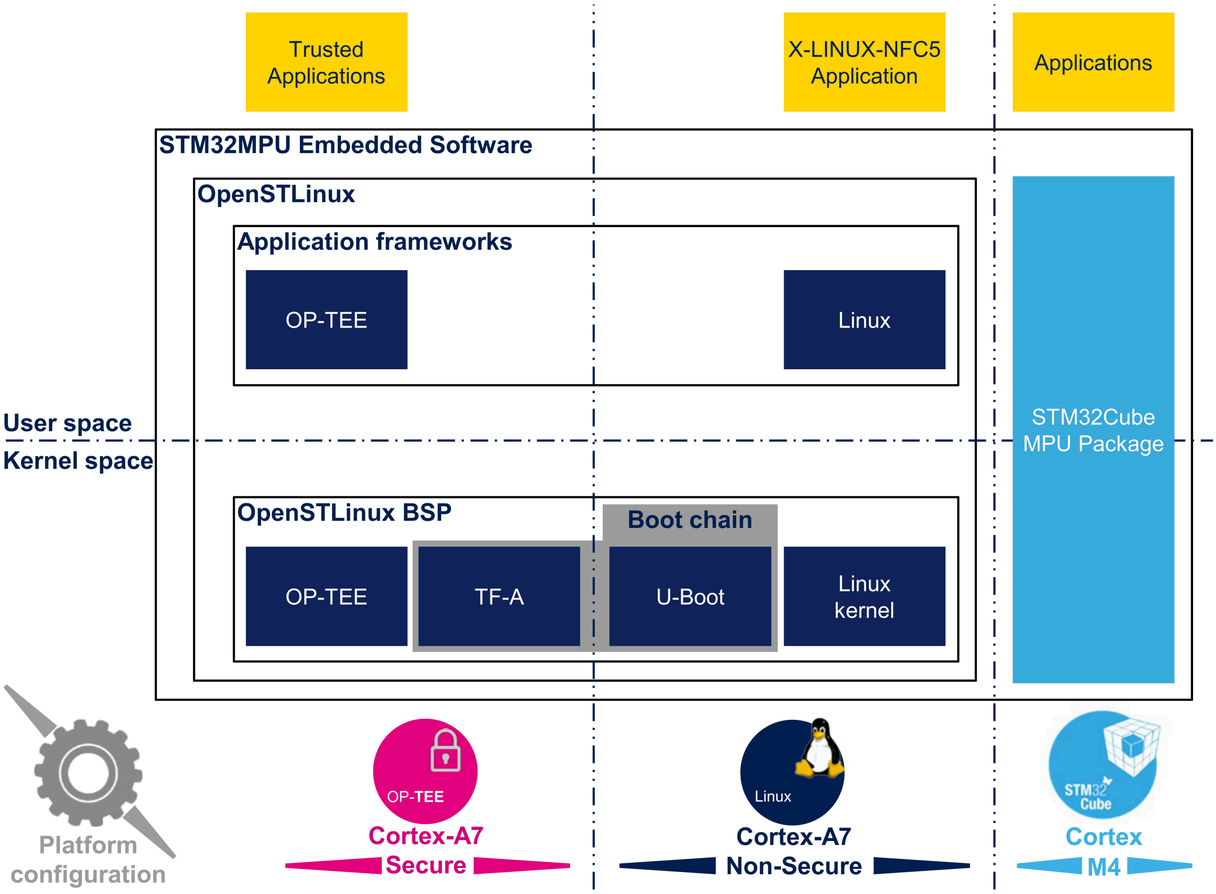
The X-LINUX-NFC5 software expansion package includes the following features:

- Complete Linux user space driver (RF abstraction layer) to build NFC enabled applications using the ST25R3911B/ST25R391x NFC front ends with up to 1.4 W output power.
- Linux host communication with the ST25R3911B/ST25R391x via high speed 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, etc.)
- Sample implementation available with X-NUCLEO-NFC05A1 expansion board plugged on an STM32MP157F-DK2
- Sample application to detect several NFC tags types

1.2 Package architecture

The software package runs on the A7 core of the STM32MP1 series . The X-LINUX-NFC5 interacts with the lower layers libraries and SPI lines exposed by the Linux software framework.

Figure 2. X-LINUX-NFC5 application architecture in Linux environment



2 Hardware setup

Hardware requirements:

- Ubuntu-based PC/Virtual-machine version 16.04 or higher
- STM32MP157F-DK2 board (Discovery Kit)
- X-NUCLEO-NFC05A1
- 8 GB micro SD card to boot the STM32MP157F-DK2
- SD card reader / LAN connectivity
- USB Type-A to Type-micro B USB cable
- USB Type A to Type-C USB cable
- USB PD compliant 5V 3A power supply

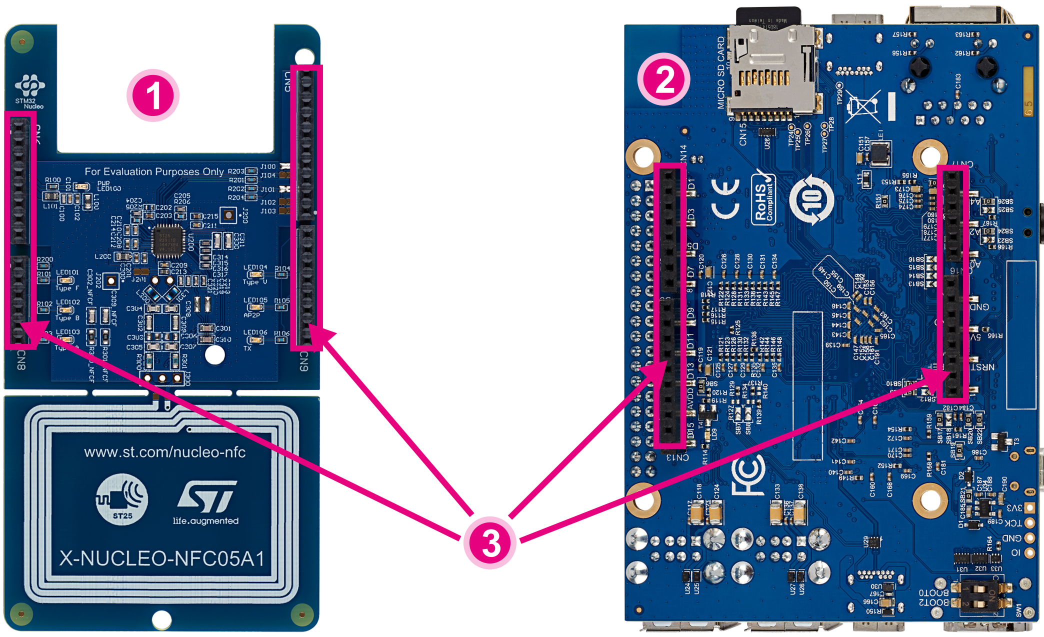
The PC/Virtual-machine forms the cross-development platform to build the RFAL library and application code to detect and communicate with NFC devices through the ST25R3911B IC.

2.1 How to connect the hardware

Step 1. Plug the X-NUCLEO-NFC05A1 expansion board onto the Arduino connectors on the bottom side of the STM32MP157F-DK2 discovery board.

Figure 3. Nucleo board and Discovery board Arduino connectors

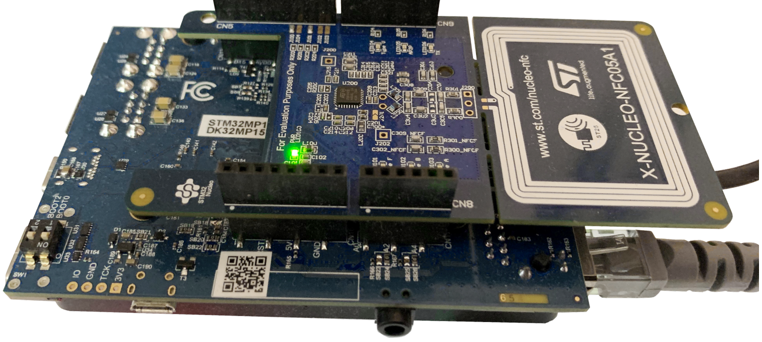
1. X-NUCLEO-NFC05A1 expansion board
2. STM32MP157F-DK2 discovery board
3. Arduino connectors



Step 2. Connect the ST-LINK programmer/debugger embedded on the discovery board to your host PC via the USB micro B type port (CN11).

Step 3. Power the discovery board through the USB Type C port (CN6).

Figure 4. Full hardware connection setup



RELATED LINKS

Refer to this wiki for more details related to power supply and communication ports

3 Software setup

Before you begin, power the STM32MP157F-DK2 Discovery kit via a USB PD compliant 5 V, 3 A power supply and install the Starter Package according to the instructions in the Getting Started wiki. You will need a minimum 2 GB microSD Card to flash the bootable images.

To run the application, the platform configuration needs to be updated by updating the device tree to enable the relevant peripherals. You can do this quickly by using the pre-built images available, or you can develop the device tree and build your own kernel images.

You can also (optionally) build this software package by including the Yocto layer (meta-nfc5) in the ST distribution package. This operation creates the source code and includes the device-tree modifications along with compiled binaries in the final flashable images. For detailed steps describing the process, see Section 3.5 .

You can connect to the Discovery Kit from the host PC via TCP/IP network using `ssh` and `scp` commands, or through serial UART or USB links using tools like minicom for Linux or Tera Term for Windows.

RELATED LINKS

Refer to these instructions for installing the Starter Package onto the Discovery kit

3.1 Steps for quick evaluation of software on page 6

3.2 How to update the platform configuration in the developer package on page 7

4 How to transfer files using Tera Term on page 10

Refer to this wiki page for alternative ways of estashing PC communication with the board

3.1 Steps for quick evaluation of software

Step 1. Flash the Starter Package on the SD Card.

Step 2. Boot the board with Starter Package.

Step 3. Enable internet connectivity on the board via Ethernet or Wi-Fi.
Refer to relevant wiki pages for help.

Step 4. Download pre-built images from the X-LINUX-NFC5 web page on the ST website

Step 5. Use the following commands to copy the device tree blob and update the new platform configuration:
If network connectivity is not available, you can transfer the files locally from your Windows PC to the Discovery Kit using Tera Term.
For further details on transferring data files using Tera Term, see Section 4 .

```
PC $> cd X-LINUX-NFC5_v1.1.0/STM32MP157F-DK2_DeviceTree/Binaries
PC $> scp stm32mp157f-dk2.dtb root@<ip address of board>:/boot/
PC $> ssh root@<ip address of board>
Board $> /sbin/depmod -a
Board $> sync
Board $> reboot
```

Step 6. After the board boots up, copy the application binary and the shared lib to discovery board.

```
PC $> cd X-LINUX-NFC5_v1.1.0/NFCPollerApplication/Binaries
PC $> scp ./nfcpoller_st25r3911 root@<ip address of board>:/usr/bin
PC $> scp ./librfal_st25r3911.so root:<ip address of board>:/usr/lib
PC $> ssh root@<ip address of board>
Board $> cd /usr/bin
Board $> chmod +x nfc_poller_st25r3911
Board $> ./nfc_poller_st25r3911
```

The application will start running once these commands are executed.

RELATED LINKS
4 How to transfer files using Tera Term on page 10

3.2 How to update the platform configuration in the developer package

The following steps will allow you to set up the development environment.

Step 1. Download Developer Package and install the SDK in the default folder structure on your Ubuntu machine.

You can find the instructions here: Install SDK

Step 2. Open the device tree file 'stm32mp157f-dk2.dts' in the Developer Package source code and add the code snippet below to the file:

This updates the device tree to enable and configure the SPI4 driver interface.

```
&spi4 {
    pinctrl-names = "default", "sleep";
    pinctrl-0 = <&spi4_pins_b>;
    pinctrl-1 = <&spi4_sleep_pins_b>;
    /*status = "disabled";*/
    cs-gpios = <&gpioe 11 0>;
    status = "okay";

    spidev@0x00 {
        compatible = "semtech,sx1301";
        spi-max-frequency = <5000000>;
        reg = <0>;
    };
};
```

Step 3. Compile the Developer package to get the stm32mp157f-dk2.dtb file.

Refer to the following link for help: Modify, rebuild and reload the Linux® kernel.

RELATED LINKS
3.1 Steps for quick evaluation of software on page 6

3.3 How to build the RFAL Linux application code

Before you begin, the SDK must be downloaded, installed and enabled. Download the application from the link: X-LINUX-NFC5

Step 1. Run the commands below to cross-compile the code:

These commands will build following files:

– The example application: nfc_poller_st25r3911
– shared lib for running the example application: librfal_st25r3911.so

```
PC $> sudo apt-get install cmake
PC $> cd RFAL_STMPU_release_v1.1/NFCPollerApplication/Source/
Linux_RFAL_st25r3911_v2.1.0/linux_demo/build
PC $> cmake..
PC $> make
```

RELATED LINKS
3.1 Steps for quick evaluation of software on page 6

3.4 How to run the RFAL Linux application on STM32MP157F-DK2

Step 1. Copy generated binaries onto the Discovery Kit using below commands

```
PC $> scp
X-LINUX-NFC5_v1.1.0/NFCPollerApplication/Source/Linux_RFAL_st25r3911v2.2.0/
linux_demo/build/nfc_poller/nfc_poller_st25r3911
root@<board ip address>:/usr/bin
PC $> scp
X-LINUX-NFC5_v1.1.0/NFCPollerApplication/Source/
Linux_RFAL_st25r3911_v2.2.0/linux_demo/build /rfal/st25r3911/librfal_st25r3911.so
root@<board ip address>:/usr/lib
```

Step 2. Open terminal on the Discovery Kit board or use `ssh` login and run the application using the following commands.

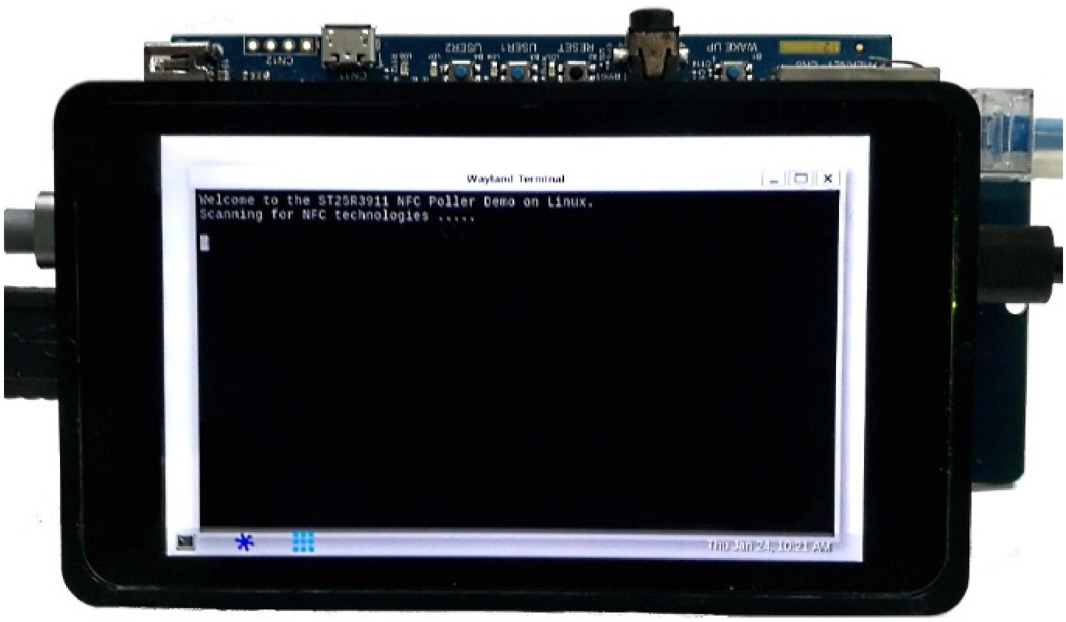
```
PC $> ssh root@<board ip address>
Board $> cd /usr/bin/  #enter directory where executable file was copied
Board $> ./nfc_poller_st25r3911 # Run the application
```

The user will see the below message on the screen:

```
Welcome to the ST25R3911B NFC Poller Demo on Linux. Scanning for NFC
Technologies ......
```

Step 3. When an NFC tag is brought near the NFC receiver, the UID and NFC tag type is displayed on the screen.

Figure 5. **Discovery Kit running the nfcPoller application**



3.5 How to include meta-nfc5 layer in the Distribution Package

Step 1. Download and compile the Distribution Package on your Linux machine.

Step 2. Follow the default directory structure suggested by ST wiki page to follow this document synchronously.

Step 3. Download the X-LINUX-NFC5 application package:

```
PC$> cp -rf X-LINUX-NFC5_v1.1.0/NFCPollerApplication/Source/meta-nfc5/
STM32MP15-Ecosystem-v3.0.0/Distribution-Package/openstlinux-5.10-dunfell-
mp1-21-03-31/layers
PC$> cd STM32MP15-Ecosystem-v3.0.0/Distribution-Package/openstlinux-5.10-dunfell-
mp1-21-03-31/
```

Step 4. Set up the build configuration.

```
PC$> DISTRO=openstlinux-weston MACHINE=stm32mp1 source layers/meta-st/scripts/
envsetup.sh
```

Step 5. Add the meta-nfc5 layer to the build configuration of the Distribution Package configuration.

```
PC$> bitbake-layers add-layer ../layers/meta-nfc5
```

Step 6. Update the configuration to add new components in your image.

```
PC$> echo 'IMAGE_INSTALL_append += "nfc5"' >>
../layers/meta-st/meta-st-openstlinux/conf/layer.conf
```

Step 7. Build your layer separately and then build the complete Distribution Layer.

```
PC$> bitbake st-image-weston
```

Note: *Building the distribution page for the first time may take several hours. However, it takes only few minutes to build meta-nfc5 layer and install the executables in the final images. Once the build is complete, the images are present in the following directory: build-<distro>-<machine>/tmp-glibc/deploy/images/stm32mp1.*

Step 8. Follow instructions on ST wiki page: Flashing the built image to flash the new built images onto the discovery kit.

Step 9. Run the application as mentioned in Step 2 of Section 3.4 .

4 How to transfer files using Tera Term

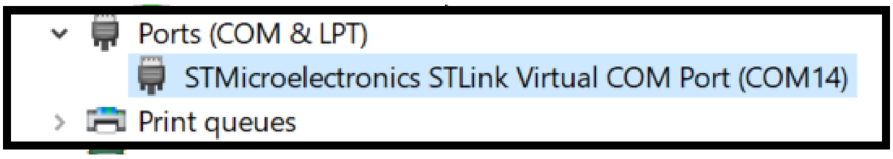
You can use a Windows terminal emulator application like Tera Term to transfer files from your PC to the Discovery Kit.

Step 1. Supply USB power to the Discovery Kit.

Step 2. Connect the Discovery Kit to your PC via the USB micro B type connector (CN11).

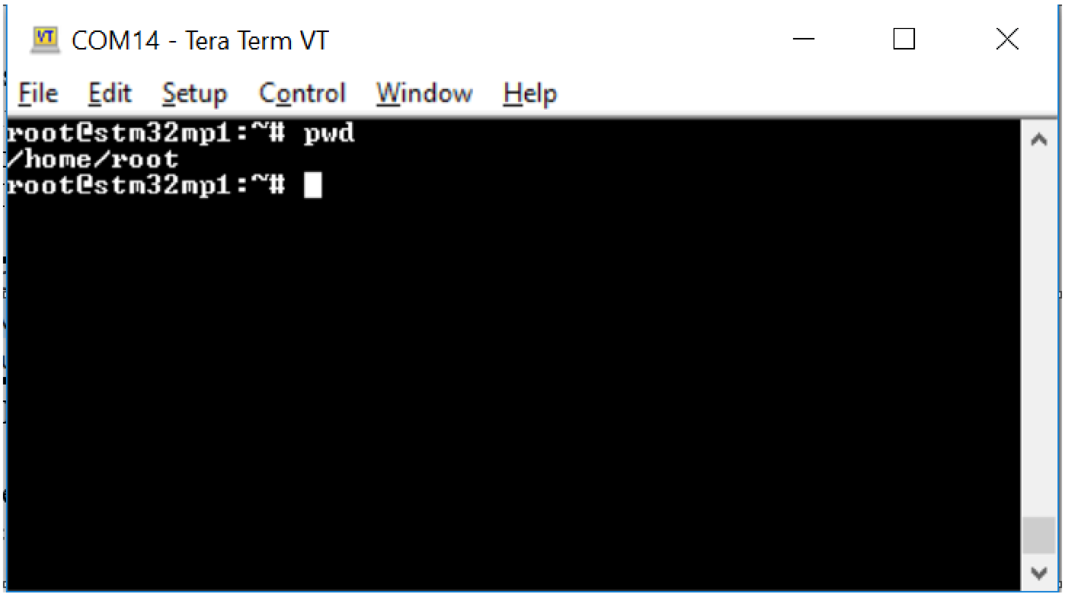
Step 3. Check the Virtual COM port number in the device manager.
In the screenshot below, the COM port number is 14.

Figure 6. Screenshot of device manager showing virtual com port



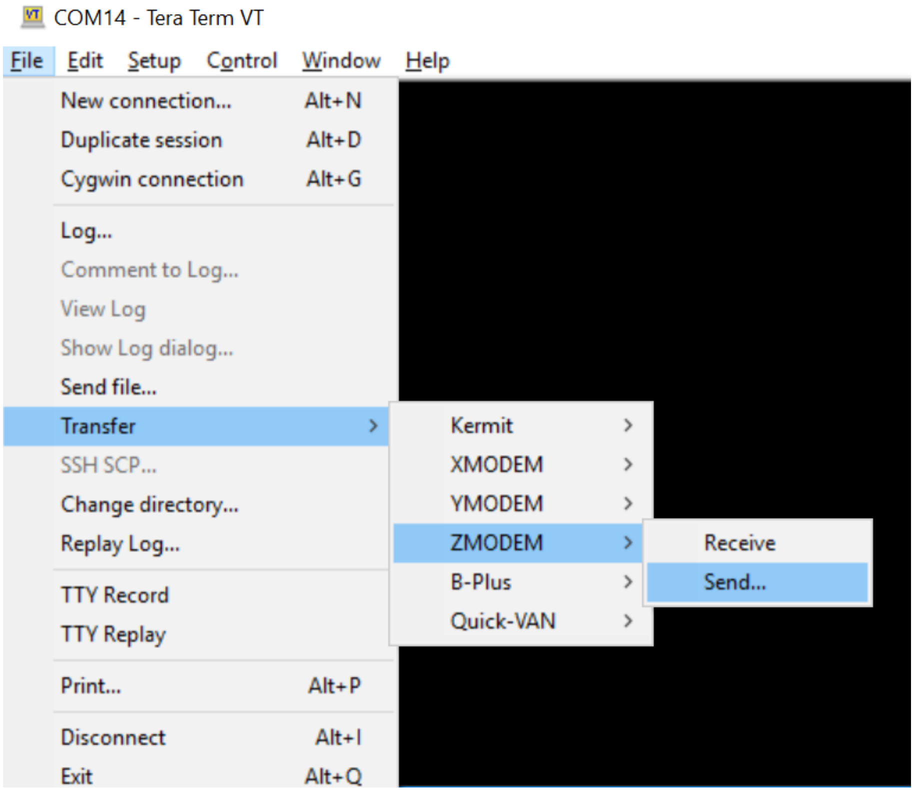
Step 4. Open Tera Term on your PC and select the COM port identified in the previous step.
The baud rate should be 115200 baud.

Figure 7. Snapshot of remote terminal via Tera Term



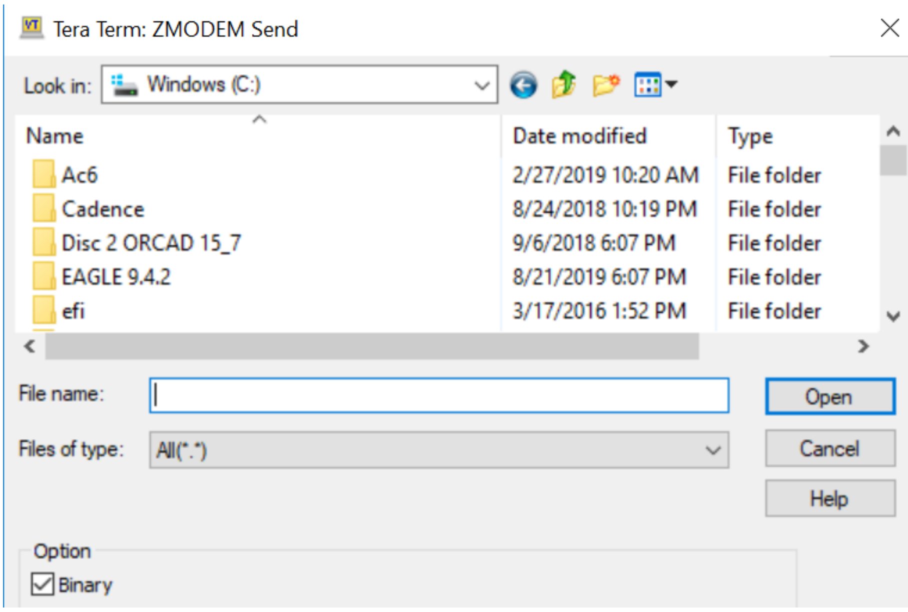
Step 5. To transfer a file from the host PC to Discovery Kit, select [**File**]>[**Transfer**]>[**ZMODEM**]>[**Send**] in top left corner of the Tera Term window.

Figure 8. Tera Term file transfer menu



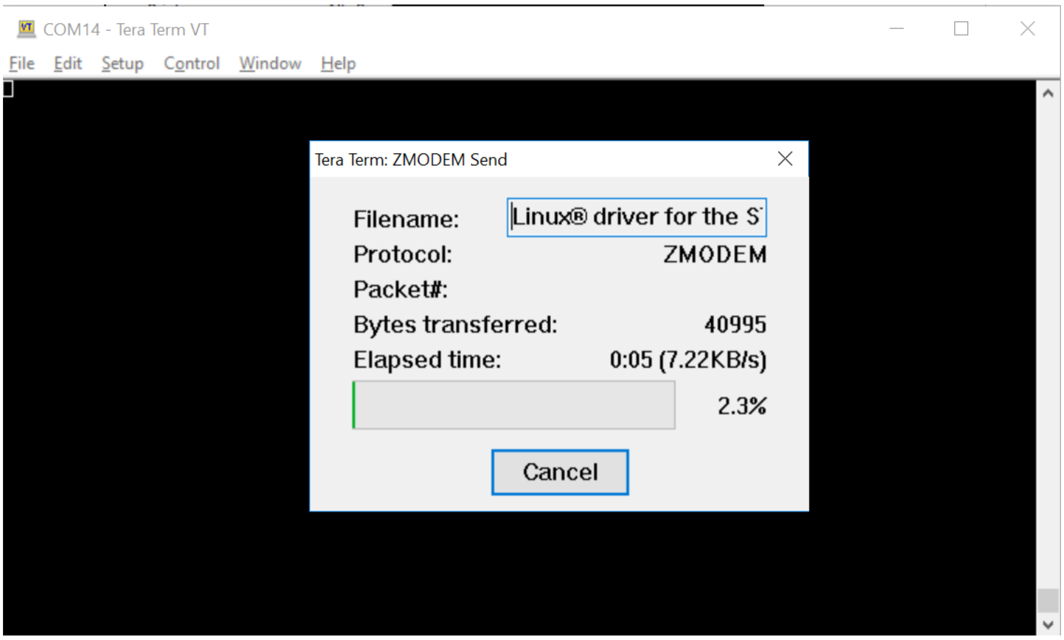
Step 6. Select the file to be transferred in the file browser and select [**Open**].

Figure 9. File browser window for sending files



Step 7. A progress bar will show the status of file transfer.

Figure 10. File transfer progress bar



Revision history

Table 1. Document revision history

Date	Version	Changes
30-Oct-2020	1	Initial release.
15-Jul-2021	2	Updated Section 1.1 Main features, Section 2 Hardware setup, Section 2.1 How to connect the hardware, Section 3 Software setup, Section 3.1 Steps for quick evaluation of software, Section 3.2 How to update the platform configuration in the developer package and Section 3.3 How to build the RFAL Linux application code. Added Section 3.5 How to include meta-nfc5 layer in the Distribution Package. Added STM32MP157F-DK2 discovery kit compatibility information.

Contents

List of figures

List of tables