#### Draft

**USER MANUAL**

Linux® driver for the ST25R3911B high performance NFC frontends

**Introduction:**

The software package enables the STM32MP157C-DK2 to operate with the X-NUCLEO-NFC05A1.

This package ports the RF abstraction layer (RFAL) onto a STM32MP157C-DK2 based Linux platform to operate with X-NUCLEO-NFC05A1 firmware, which contains the ST25R3911B high-performance NFC frontend. The package contains a sample application that detects different types of NFC tags. 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 (STSW-ST25R002) and X-NUCLEO-NFC05A1 (X-CUBE-NFC5) firmware. This software package supports all the ST25R3911B lower-layer protocols and also some higher layer protocols to abstract RF communication. As the RFAL is written in a portable manner, it can run on a wide range of devices, from 8-bit MCUs up to 64-bit processors running Linux.

This document describes how the RFAL library can be used on a standard Linux system (in this case the STM32MP157C-DK2) for NFC/RF communication. All the code here is highly portable and works with minor changes on any Linux platform

Figure 1: RFAL library on Linux platform

****

Table of Contents

[1. Overview 5](#_Toc42022686)

[**1.1 Features** 5](#_Toc42022687)

[**1.2 Software Architecture** 5](#_Toc42022688)

[2. Hardware setup 7](#_Toc42022689)

[**2.1** **Platform used** 7](#_Toc42022690)

[**2.2** **Hardware requirements** 7](#_Toc42022691)

[**2.3** **Hardware connections** 7](#_Toc42022692)

[3. Software setup, build and run 8](#_Toc42022693)

[**3.1** **Booting STM32MP157C-DK2 with Starter Package** 8](#_Toc42022694)

[**3.2** **Software platform configuration for running the application** 8](#_Toc42022695)

[**3.2.1** **Steps for quick evaluation of software** 8](#_Toc42022696)

[**3.2.2** **Updating platform configuration in developer package** 9](#_Toc42022697)

[**3.2.3** **Build the RFAL Linux application code** 11](#_Toc42022698)

[**3.2.4** **RUN the RFAL Linux application on STM32MP157C-DK2** 11](#_Toc42022699)

[4. How to transfer files using ‘Tera Term’: Windows PC to Discovery Kit 12](#_Toc42022700)

Figures

[Figure 1: RFAL library on Linux platform 1](file:///C:\My%20Projects\STM32MP1_WORK\X-LINUX-NFC5\UserManual_Linux%20application%20for%20NFC%20frontend.docx#_Toc41925826)

[Figure 2: RFAL software architecture on Linux 5](file:///C:\My%20Projects\STM32MP1_WORK\X-LINUX-NFC5\UserManual_Linux%20application%20for%20NFC%20frontend.docx#_Toc41925827)

[Figure 3: Hardware Setup 7](file:///C:\My%20Projects\STM32MP1_WORK\X-LINUX-NFC5\UserManual_Linux%20application%20for%20NFC%20frontend.docx#_Toc41925828)

[Figure 4: Picture of Discovery Kit running the nfcPoller application 9](#_Toc41925829)

[Figure 5: Snapshot of device manager showing virtual com port 13](#_Toc41925830)

[Figure 6: Snapshot of remote terminal via Tera Term 13](#_Toc41925831)

[Figure 7: Select from menu to transfer file from PC to Discovery Kit 14](#_Toc41925832)

[Figure 8: Pop-up window for selecting file(s) to be transferred 14](#_Toc41925833)

[Figure 9: File transfer progress bar 15](#_Toc41925834)

Tables

**No table of figures entries found.**

# 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 high performance NFC frontends 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 an STM32MP157C-DK2
* Sample application to detect several NFC tag types and mobile phones supporting P2P
* Free user-friendly license terms

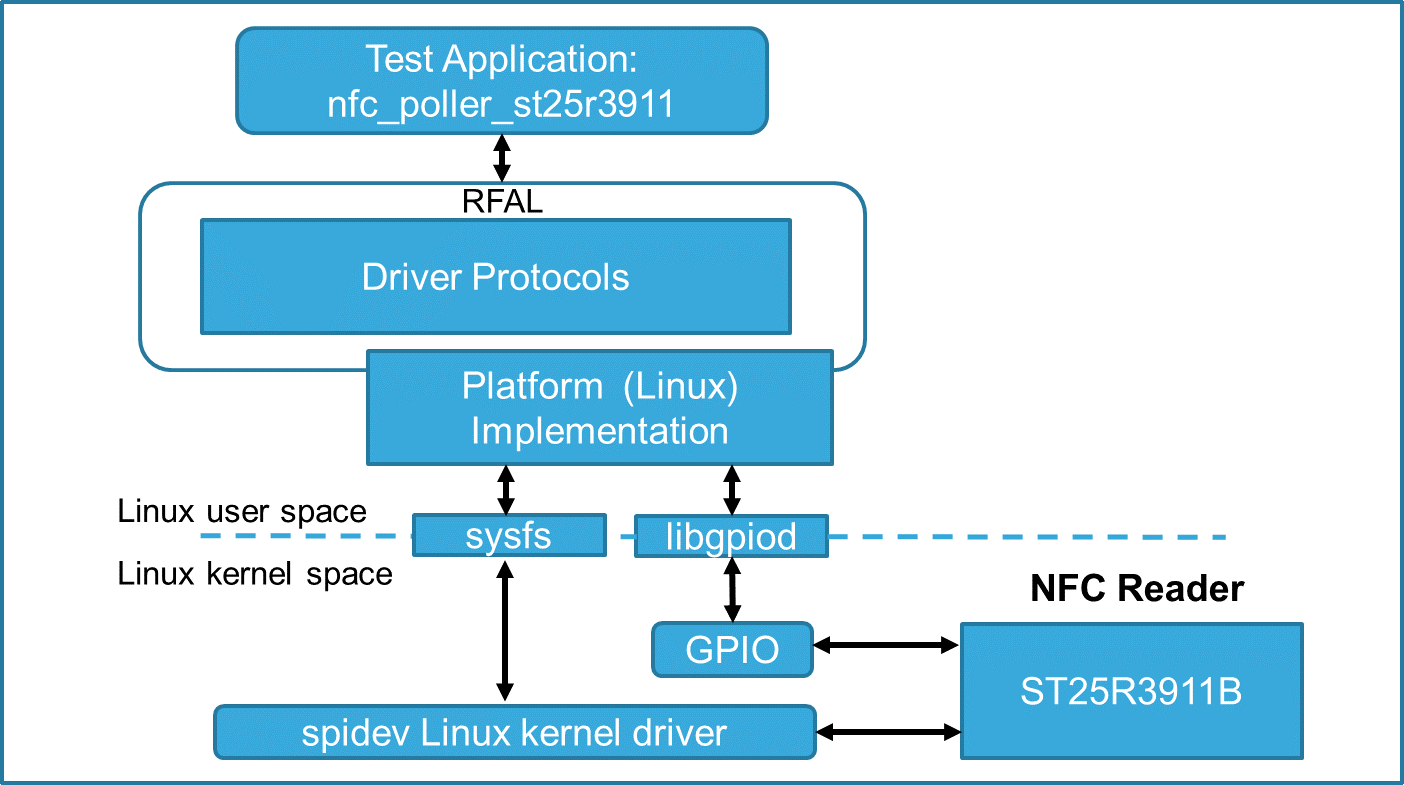
## **Software Architecture**

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 (platform.h) contains macro definitions, which are expected to be provided and implemented by the platform owner. It provides platform specific settings such as GPIO assignment, system resources, locks and IRQs, which are required for correct operation of the RFAL.

This demo implements the platform functions and provides a port of the RFAL into user space of Linux. A shared library file is generated, which is used by a test/demo application to showcase the functionalities provided by the RFAL layer. Linux host uses libgpiod interface for accessing the interrupt GPIO and SPI interface via spidev (device driver) for communication with the ST25R3911B device.

The demo application software (nfcPoller) has been developed and tested for the [STM32MP1 OpenSTLinux Developer Package](https://www.st.com/content/st_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32-mpu-openstlinux-distribution/stm32mp1dev.html)

Figure 2: RFAL software architecture on Linux



# Hardware setup

## **Platform used**

An Ubuntu 16.04 based PC/Virtual-machine or higher, shall be used as a cross-development platform to build RFAL library and the application code.

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

## **Hardware requirements**

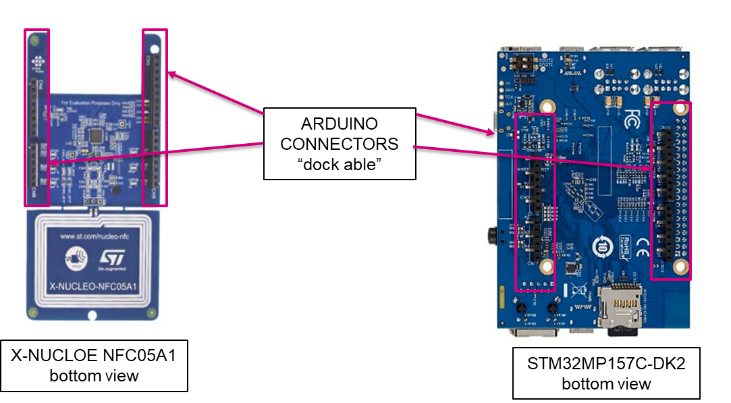
* STM32MP157C-DK2 board (Discovery Kit)
* X-NUCLEO-NFC05A1
* 8 GB micro SD card to boot the STM32MP157C-DK2
* USB Type-A to Type-micro B cable
* USB Type A to Type-C cable
* USB PD compliant 5V 3A power supply

## **Hardware connections**

1. Dock the [X-NUCLEO-NFC05A1](https://www.st.com/content/st_com/en/products/ecosystems/stm32-open-development-environment/stm32-nucleo-expansion-boards/stm32-ode-connect-hw/x-nucleo-nfc05a1.html) board on to the Arduino connector present on the bottom side of the [STM32MP157C-DK2](https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-mpu-eval-tools/stm32-mcu-mpu-eval-tools/stm32-discovery-kits/stm32mp157c-dk2.html) Discovery Kit as shown in Figure 3 below
2. Connect the on-board ST-Link (CN11) to your host PC via USB micro B type cable.
3. Connect power adapter via USB Type C cable to CN6

Refer to [wiki page](https://wiki.st.com/stm32mpu/wiki/Getting_started/STM32MP1_boards/STM32MP157C-DK2/Let%27s_start/Unpack_the_STM32MP157C-DK2_board) for more details related to power & communication ports on the discovery kit.

Figure 3: Hardware Setup



# Software setup, build and run

## **Booting STM32MP157C-DK2 with Starter Package**

Unpack the discovery kit and power it up using a USB PD compliant 5V, 3A power supply. Use minimum 2GB microSD Card to flash the bootable images. Please follow instructions mentioned in the following link for installing Starter Package onto discovery kit–

<https://wiki.st.com/stm32mpu/wiki/Getting_started/STM32MP1_boards/STM32MP157C-DK2>

## **Software platform configuration for running the application**

Before running the application, platform configuration needs to be updated. User need to download Developer Package and update device tree for enabling the relevant peripherals.

For quick evaluation of the software user can refer to [section 3.2.1](#_Summary_of_steps_1) and use the pre-built images and easily have the application running on discovery kit.

In case the user wants a full-fledged development environment, detailed description of how to modify the device tree and build updated kernel images is mentioned in [section 3.2.2](#_Summary_of_steps)

**Note:** In this document, we will access the Discovery Kit from host PC via TCP/IP network (using ssh and scp commands). Another method to access the board could be through serial link (UART / USB) using tools such as ‘minicom’ (on Linux PC) or ‘Tera-term’ (on Windows PC). [Section 4](#_How_to_transfer) explains how to transfer file using Tera-Term. For more details, please refer to wiki page: [How to get Terminal](https://wiki.st.com/stm32mpu/wiki/How_to_get_Terminal).

## **Steps for quick evaluation of software**

In order to speed-up the development process and quickly evaluate the software, please follow below mentioned steps–

* Please follow instructions in the link : [STM32MP157C-DK2 Let's start](https://wiki.st.com/stm32mpu/wiki/Getting_started/STM32MP1_boards/STM32MP157C-DK2/Let%27s_start). to flash the Starter Package on the SD Card.
* Boot the board with Starter Package.
* Enable internet connectivity on the board via Ethernet OR Wi-Fi. Refer to wiki pages for help.
* Download pre-built images - Link for download: TBD after release on www.st.com
* Use below commands to copy the device tree blob and update the new platform configuration –

PC $> cd RFAL\_STMPU\_release\_v1.0/STM32MP157C-DK2\_DeviceTree/Binaries

PC $> scp stm32mp157c-dk2.dtb root@<ip address of board>:/boot/

PC $> ssh root@<ip address of board>

Board $> /sbin/depmod –a

Board $> sync

Board $> reboot

Note: In case network connectivity is not available, please refer to section: 'How to transfer files using ‘Tera Term’: Windows PC to Discovery Kit' for doing file transfer locally.

* After the board boots up, copy the application binary and the shared lib to discovery board –

PC $> cd RFAL\_STMPU\_release\_v1.0/NFCPollerApplication/Binaries

PC $> scp ./\* root@<ip address of board>:/usr/local

PC $> ssh root@<ip address of board>

Board $> cd /usr/local

Board $> export LD\_LIBRARY\_PATH=/usr/local/:$LD\_LIBRARY\_PATH

Board $> chmod +x nfc\_poller\_st25r3911

Board $> ./nfc\_poller\_st25r3911

The application will start running once the above-mentioned commands are executed. Figure 5 shows the application running on the discovery board. 

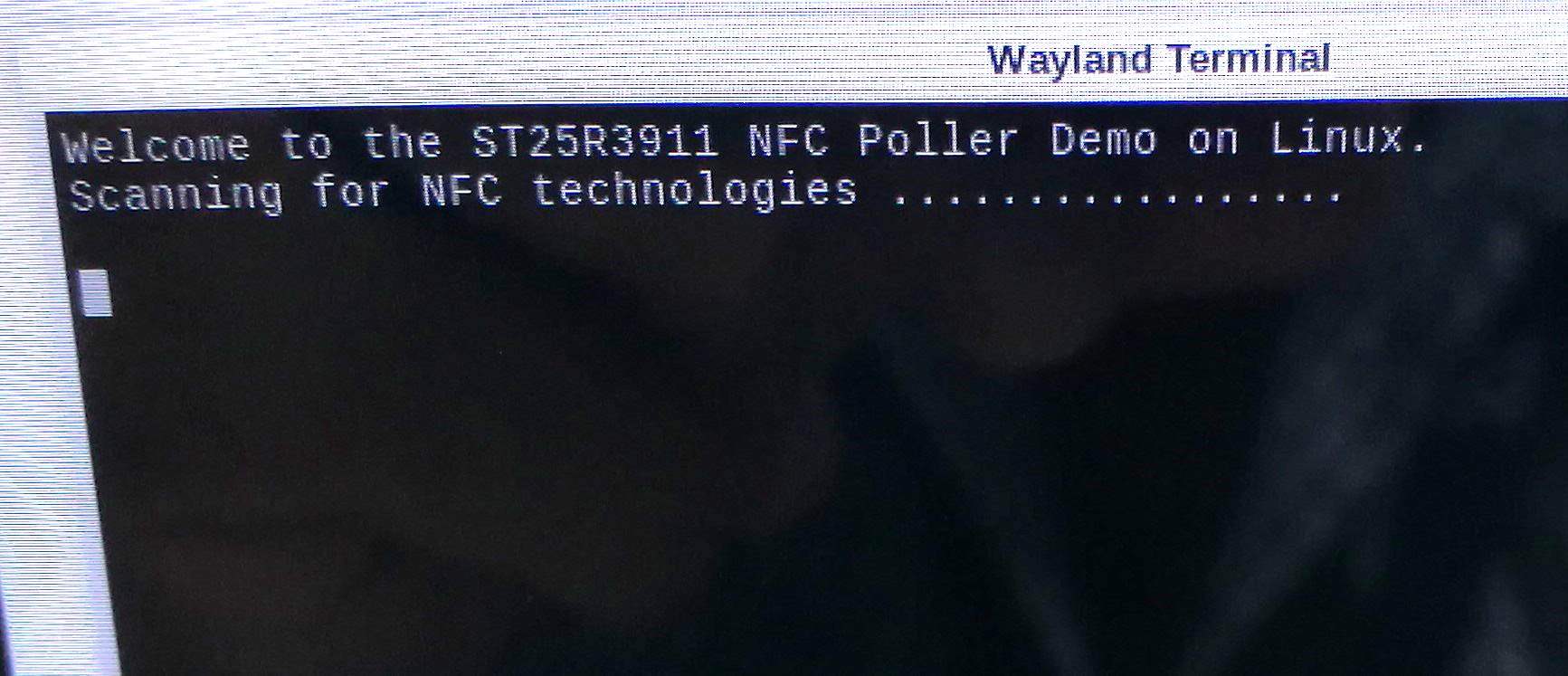


Figure 4: Picture of Discovery Kit running the nfcPoller application

**Note:** In case network connectivity is not available, please refer to [chapter 4](#_How_to_transfer) for doing file transfer locally.

## **Updating platform configuration in developer package**

In order to set up the development platform, please follow below mentioned steps –

* Download Developer Package and install the SDK in the default folder structure on your Ubuntu machine. Please follow : [Getting\_started/STM32MP1\_boards/STM32MP157C-DK2/Develop\_on\_Arm®\_Cortex®-A7/Install\_the\_SDK](https://wiki.st.com/stm32mpu/wiki/Getting_started/STM32MP1_boards/STM32MP157C-DK2/Develop_on_Arm%C2%AE_Cortex%C2%AE-A7/Install_the_SDK)
* Update the device tree as mentioned below to enable and configure the SPI4 driver interface. Open the device tree file ‘stm32mp157c-dk2.dts’ in the Developer Package source code and add below code snippet to the file-

&pinctrl{

spi4\_pins\_a: spi4-0 {

pins1 {

pinmux = <STM32\_PINMUX('E', 12, AF5)>, /\* SPI4\_SCK \*/

<STM32\_PINMUX('E', 14, AF5)>; /\* SPI4\_MOSI \*/

bias-disable;

drive-push-pull;

slew-rate = <1>;

};

pins2 {

pinmux = <STM32\_PINMUX('E', 13, AF5)>; /\* SPI4\_MISO \*/

bias-disable;

};

};

spi4\_sleep\_pins\_a: spi4-sleep-0 {

pins {

pinmux = <STM32\_PINMUX('E', 12, ANALOG)>, /\* SPI4\_SCK \*/

<STM32\_PINMUX('E', 13, ANALOG)>, /\* SPI4\_MISO \*/

<STM32\_PINMUX('E', 14, ANALOG)>; /\* SPI4\_MOSI \*/

};

};

};

&spi4 {

pinctrl-names = "default", "sleep";

pinctrl-0 = <&spi4\_pins\_a>;

pinctrl-1 = <&spi4\_sleep\_pins\_a>;

/\*status = "disabled";\*/

cs-gpios = <&gpioe 11 0>;

status = "okay";

spidev@0x00 {

compatible = "semtech,sx1301";

spi-max-frequency = <5000000>;

reg = <0>;

};

};

* Compile the Developer package to get the stm32mp157c-dk2.dtb file. Please refer to following link for help. [Getting\_started/STM32MP1\_boards/STM32MP157C-DK2/Develop\_on\_Arm®\_Cortex®-A7/Modify,\_rebuild\_and\_reload\_the\_Linux®\_kernel](https://wiki.st.com/stm32mpu/wiki/Getting_started/STM32MP1_boards/STM32MP157C-DK2/Develop_on_Arm%C2%AE_Cortex%C2%AE-A7/Modify,_rebuild_and_reload_the_Linux%C2%AE_kernel)
* Please refer to section 3.2.1 of this article for pushing the compiled device tree blob onto the discovery kit.

## **Build the RFAL Linux application code**

1. SDK must be downloaded, installed and enabled. This should already be in place if step #1 of [section 3.2.2](#_Updating_platform_configuration) is executed.
2. Download the application from the link: Link for download: TBD after release on www.st.com
3. Follow below commands to cross-compile the code -

**PC $>** sudo apt-get install cmake

**PC $>** cd RFAL\_STMPU\_release\_v1.0/NFCPollerApplication/Source/Linux\_RFAL\_st25r3911\_v2.1.0/linux\_demo/build

**PC $>** cmake ..

**PC $>** make

1. The above-mentioned commands would build following files –
   * **The example application: nfc\_poller\_st25r3911**
   * **Shared library for running the example application: librfal\_st25r3911.so**

## **RUN the RFAL Linux application on STM32MP157C-DK2**

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

**PC $>** scp RFAL\_STMPU\_release\_v1.0/NFCPollerApplication/Source/Linux\_RFAL\_st25r3911\_v2.1.0/linux\_demo/build/nfc\_poller/nfc\_poller\_st25r3911 root@<board ip address>:/usr/local

**PC $>** scp RFAL\_STMPU\_release\_v1.0/NFCPollerApplication/Source/Linux\_RFAL\_st25r3911\_v2.1.0/linux\_demo/build /rfal/st25r3911/librfal\_st25r3911.so root@<board ip address>:/usr/local

1. Open terminal on the Discovery Kit board or use **ssh** login and run the application using below commands.

**PC $>** ssh root@<board ip address>

**Board $>** export LD\_LIBRARY\_PATH=/usr/local/:$LD\_LIBRARY\_PATH #export the shared lib to the environment

**Board $>** cd /usr/local/ #enter directory where binaries were copied

**Board $>** ./nfc\_poller\_st25r3911 # Run the application

1. The user will see the below message on the screen:
   1. “Welcome to the ST25R3911B NFC Poller Demo on Linux. Scanning for NFC Technologies ……...”
2. When an NFC tag is brought near the NFC receiver, the user can view the UID and NFC tag type on the on the display screen.
3. Figure 4 Picture of Discovery Kit running the nfcPoller application.

# How to transfer files using ‘Tera Term’: Windows PC to Discovery Kit

The developer can use any Windows terminal emulator applications. "Tera Term" is one of them: (<http://tera-term.en.lo4d.com/>). The configuration is mentioned below:

1. Plug the power cable to power-up the board.
2. Connect the Discovery Kit to your PC via USB micro B type cable through CN11.
3. Check the Virtual COM port number visible in the device manager. For example, in the snapshot below, the COM port number is 14.

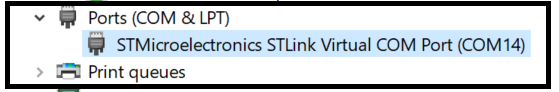


Figure 5: Snapshot of device manager showing virtual com port

1. Open Tera-Term on your PC and select the COM port (as found in step 3 above). The baud rate should be 115200 baud. The virtual terminal (remote access) will appear as shown in below snapshot.

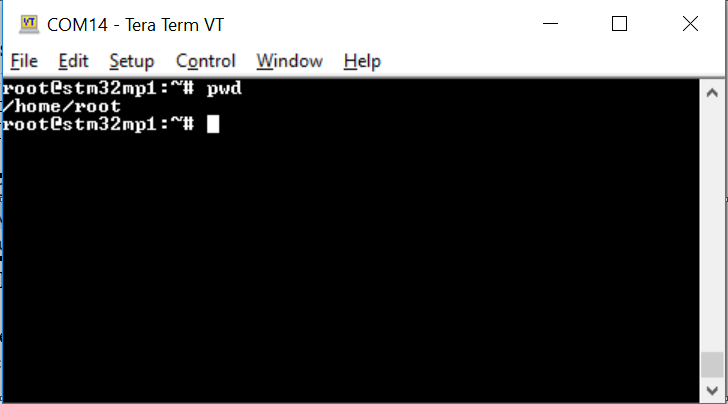


Figure 6: Snapshot of remote terminal via Tera Term

1. To transfer a file from host PC to Discovery Kit, click on File menu on top left corner of the Tera-Term window and go to File>>Transfer>>ZMODEM>>Send. Refer to below snapshot –

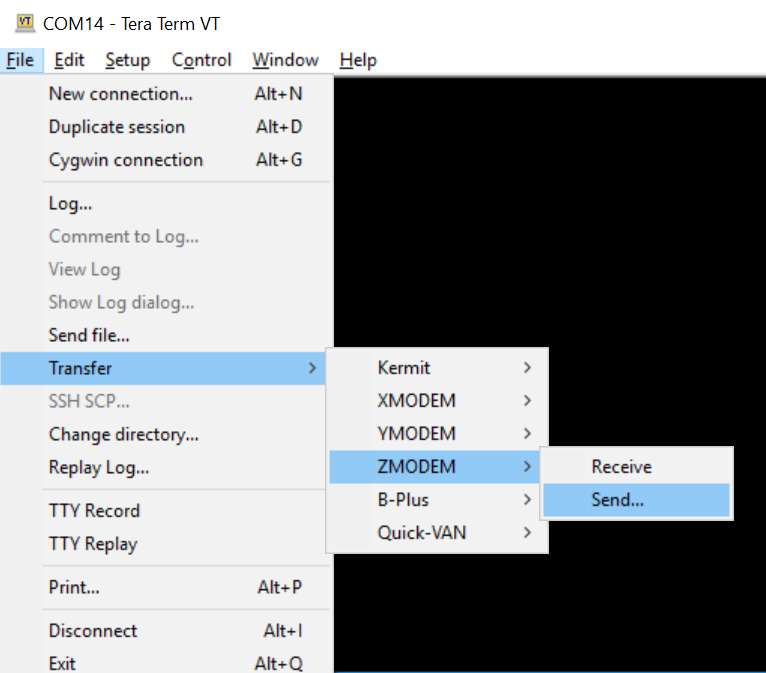


Figure 7: Select from menu to transfer file from PC to Discovery Kit

1. Select the file from pop up window (example in snapshot below) to be transferred and click on the ‘Open’ tab.

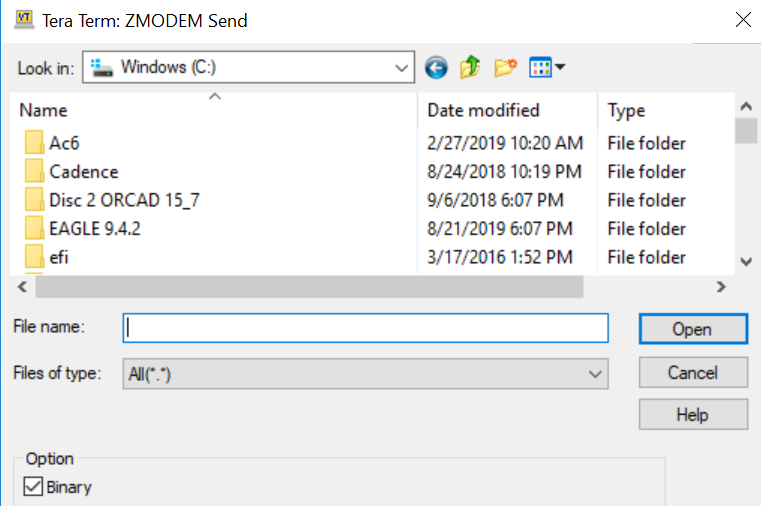


Figure 8: Pop-up window for selecting file(s) to be transferred

1. A progress bar will show the status of file transfer as shown in figure 9 below.

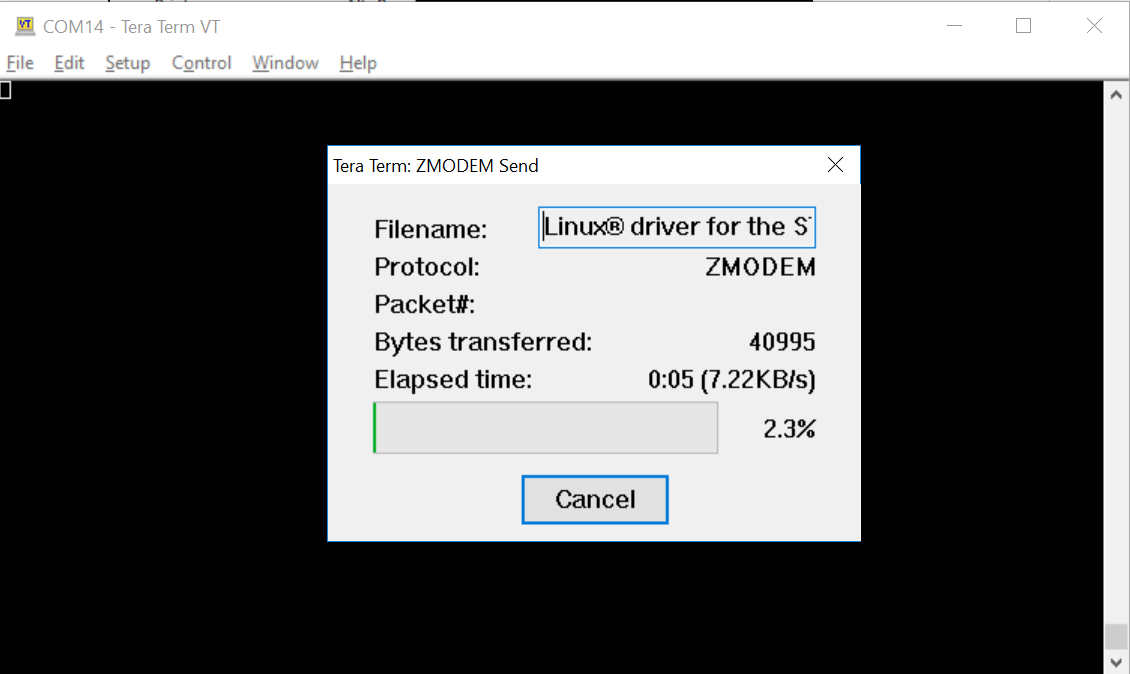


Figure 9: File transfer progress bar

**Please Read Carefully:**

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

**UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZE REPRESENTATIVE OF ST, ST PRODUCTS ARE NOT DESIGNED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS, WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE.**

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2006 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -

Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

**www.st.com**