UM2062
User manual

Firmware for the ST25DV-DISCOVERY boards

Introduction

This document describes the functionalities of the STSW-ST25DV001 firmware, developed to fully exploit the capabilities of the ST25DV-DISCOVERY boards, based on ST25DV.

The ST25DV is an IC for contactless applications that can communicate with a microcontroller through an I²C interface; this makes the ST25DV a dynamic tag. On the RF side, the reader (for instance a smart phone) can retrieve and update the content of the tag when close to it.

The reader communicates with the ST25DV-DISCOVERY board using the ISO 15693 protocol, while the STM32F405 microcontroller communicates with the ST25DV through an I²C bus.

With this discovery kit, the ST25DV can be programmed with different contents, following the NFC Forum standard. This means that a smart phone can read it natively, without any specific application being previously installed.

A specific fast transfer mode can be used to transfer data directly to the microcontroller, without using the embedded EEPROM.

Contents

List of tables

List of figures

1 List of acronyms and notational conventions

1.1 Acronyms

APB: Advanced peripheral bus

AAR: Android™ application record

BLE: Bluetooth® Low Energy

CAN: Controller area network

GPS: Global positioning system

GPO: General purpose output

H2R: Host to reader

IEC: International electrotechnical commission

ISO: International organization for standardization

MCU: Micro controller unit (microcontroller)

NFC: Near field communication

OOB: Out of band

R2H: Reader to host

RF: Radio frequency

RFID: Radio frequency identification

RISC: Reduced instruction set computer

SPI: Serial peripheral interface

URI: Uniform resource identifier

URL: Uniform resource locator

USB: Universal serial bus

1.2 Representation of numbers

The following conventions and notations apply in this document unless otherwise stated:

- **Binary numbers** are represented by strings of 0 and 1 digits shown with the most significant bit (MSB) on the left, the least significant bit (LSB) on the right, and “0b” added at the beginning. Example: 0b11110101.
- **Hexadecimal numbers** are represented by using numbers 0 to 9 and characters A to F and adding “0x” at the beginning. The most significant bit (MSB) is shown on the left and the least significant bit (LSB) on the right. Example: 0xF5.
- **Decimal numbers** are represented without any trailing character. Example: 245.

2 Overview

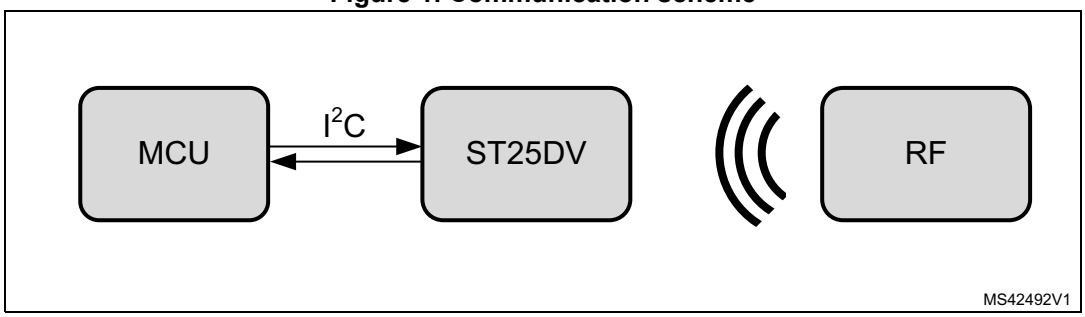
2.1 ST25DV

The ST25DV is a dynamic tag IC for contactless applications (ISO 15693). It manages the RF communication with a reader. It also includes frame coding, RF modulation and manages the anti-collision process.

The ST25DV works as an NFC Forum Type 5 tag, supporting detection, read and write operations.

The ST25DV (see *Figure 1*) can communicate with a reader without any external control.

Figure 1. Communication scheme



2.2 STM32F405

The STM32F405xx microcontrollers are based on the high-performance ARM®(a) Cortex®-M4 32-bit RISC core operating at a frequency of up to 168 MHz. The Cortex®-M4 core features a floating -point unit (FPU) single precision, which supports all ARM® single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory protection unit (MPU) that enhances application security.

The STM32F405xx incorporate high-speed embedded memories (Flash memory up to 1 Mbyte, SRAM up to 192 Kbytes), up to 4 Kbytes of backup SRAM, an extensive range of enhanced I/Os and peripherals connected to two APB buses, three AHB buses and a 32-bit multi-AHB bus matrix.

All devices offer three 12-bit ADCs, two DACs, a low-power RTC, twelve general-purpose 16-bit timers including two PWM timers for motor control, two general-purpose 32-bit timers,

a. Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

a true random number generator (RNG) and a cryptographic acceleration cell. They also feature standard and advanced communication interfaces:

- Up to three I2Cs
- Three SPIs, two I2Ss full duplex: to achieve audio class accuracy, the I2S peripherals can be clocked via a dedicated internal audio PLL or via an external clock enabling synchronization
- Four USARTs and two UARTs
- An USB OTG full-speed and an USB OTG high-speed with full-speed capability (with the ULPI)
- Two CANs
- An SDIO/MMC interface

These features make the STM32F405xx microcontrollers suitable for a wide range of applications, such as

- Motor drive and application control
- Medical equipment
- Industrial applications: PLC, inverters, circuit breakers
- Printers and scanners
- Alarm systems, video intercom and HVAC
- Home and audio appliances

2.3 ST25DV-DISCOVERY boards

The ST25DV-DISCOVERY is an evaluation kit, which allows the user to evaluate the performance of the ST25DV dynamic tag. The kit is composed of two boards:

- motherboard
- ST25DV daughter board

2.3.1 ST25DV-DISCOVERY motherboard

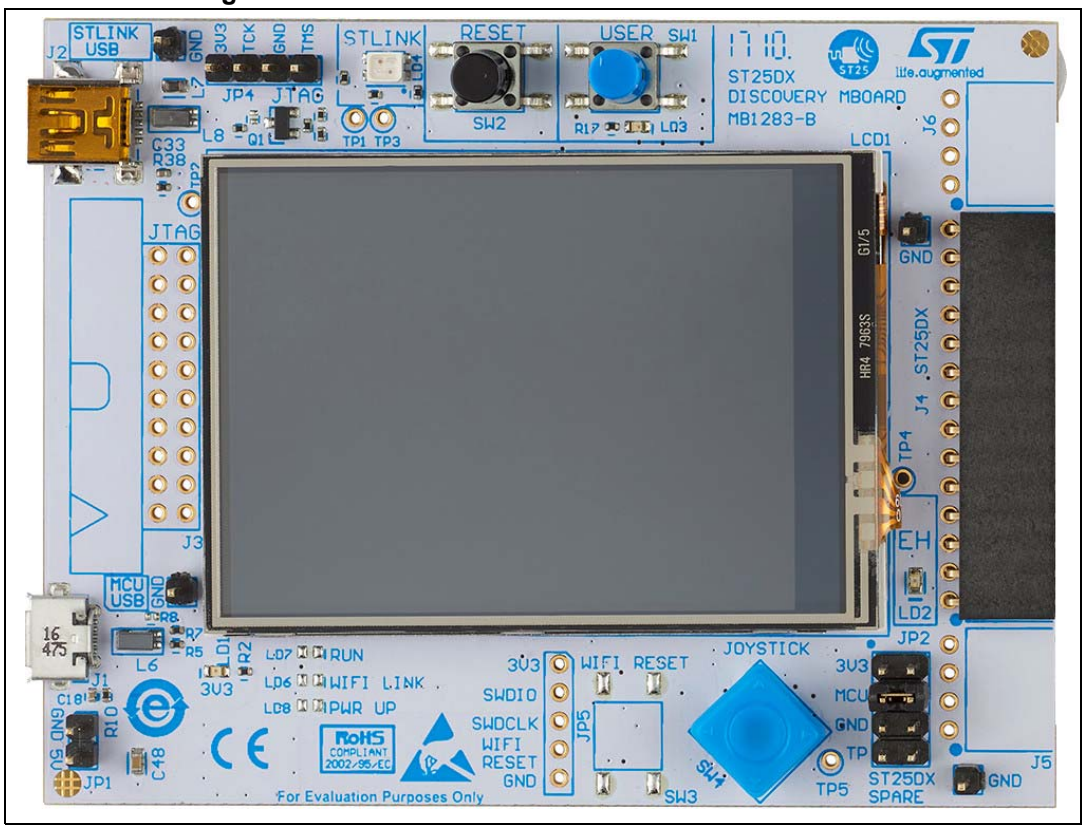
The ST25DV-DISCOVERY is powered through any of the USB bus (micro and mini connectors) and no external power supply is required.

This motherboard embeds the STM32F405VG microcontroller and different peripherals:

- LCD display and touchscreen to interface with the user
- USB connectors to connect to a PC (mini-USB for the STLink debugger and micro-USB available for the user application)
- Optional modules: Wi-Fi® and Bluetooth® Low Energy (BLE) to connect with a smart phone

The connector on the right side of the board is dedicated to the daughter boards that embed the NFC tags.

Figure 2. ST25DX-DISCOVERY motherboard MB1283-B



2.3.2 ST25DV daughter board

This board embeds the ST25DV NFC device and a 31 mm x 30 mm, 13.56 MHz double layer inductive etched antenna (no need for tuning components).

The ST25DV communicates with the STM32F405VG 32-bit MCU via the I²C bus.

Figure 3. ST25DV_Discovery_ANT_C5 board

3 Firmware description

3.1 Prerequisite

The ST25DV products are dynamic tags, their content and configuration can be driven by both a microcontroller (through I²C) and a reader through the RF.

To benefit from the ST25DV demonstration kit, the user must have one of the following:

- A smart phone with the NFC capability enabled, combined with the ST25 NFC application installed on it (see UM2131, available on *www.st.com*)
- An NFC reader with a dedicated PC software. ST25PC-NFC software embeds all features and demonstrations to be used with ST25DV-DISCOVERY board. Refer to UM2444 (Software toolbox for NFC tags) for more information about it.

The ST25DV is fully compliant with the NFC Forum Type 5 standard. This standard has been introduced recently, hence some features could not be fully supported by the oldest smart phones.

3.2 Main menu

The main menu (see *Figure 4*) is composed of three icons, used to access sub-menus.

Figure 4. ST25DV-DISCOVERY main menu display

Each item allows the user to enter a sub-menu containing a set of use cases, as indicated below:

Discover ST25DV	
Fast transfer mode demonstration	
NFC NDEF demonstrations	

Note: If the Wi-Fi® and/or the BLE modules have been added to the board the corresponding icons are available to access the Wi-Fi® and Bluetooth® pairing demonstrations.

To select a category, touch the screen on an icon. Then to start a demonstration, touch the screen on the icon of the desired demonstration (see description below). A long touch on an icon displays, in the top line, the description of the corresponding demonstration(s).

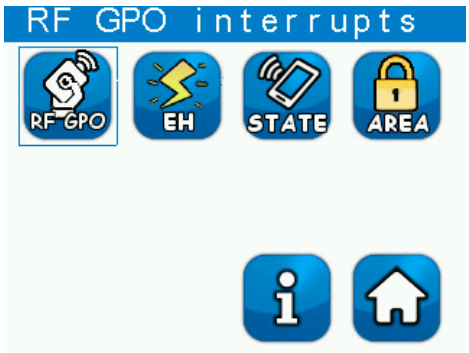
As the purpose of this document is to illustrate the firmware behavior, the descriptions is detailed from the microcontroller point of view.

3.2.1 Discover ST25DV menu

As shown in *Figure 5*, the “Discover ST25DV” menu proposes several demonstrations of the ST25DV specific features, among them:

- GPO interrupts
- Energy harvesting
- ST25DV states
- Multi-areas and passwords

Figure 5. ST25DV-DISCOVERY features menu display



3.2.2 RF GPO interrupt demonstration

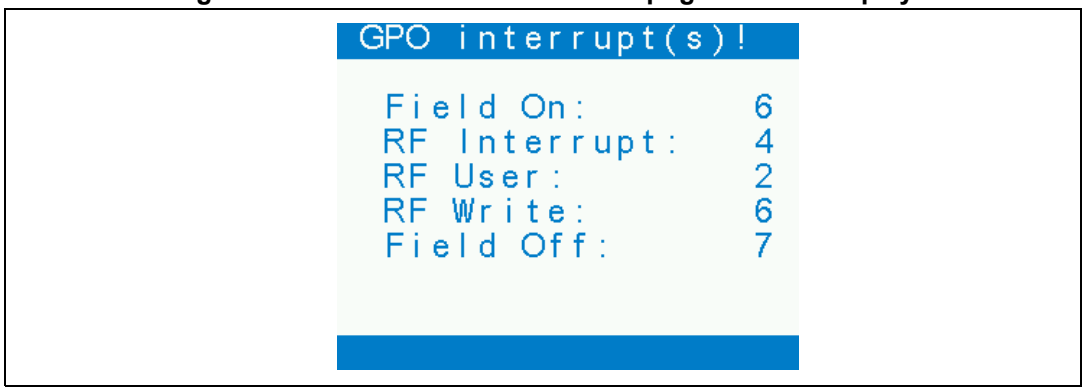
The ST25DV has a GPO that can be used to send interrupts to the microcontroller.

Several kinds of event can be configured on the RF side to trigger the GPO interrupt. In this demonstration all possible interrupt sources are enabled, except the “RF activity”, which occurs too often for correct readability when using a smart phone as a reader.

To start the demonstration, select the “GPO control Demo” menu, this displays the instructions until it detects a GPO interrupt.

As soon as the first GPO interrupt is received, the LCD displays the list of sources of received interrupt, and for each of them the number of times they occurred (see *Figure 6*).

Figure 6. ST25DV-DISCOVERY interrupt generation display



The event types and of the way to generate them are listed below:

- Field On: place a reader near the ST25DV
- Field Off: move a reader away from the ST25DV
- EEPROM written: modify the content of the ST25DV
- RF Interrupt: requires the use of a specific reader command, that can be sent either by a smart phone running the ST25 NFC application, or by a reader (see the instructions in *Smart phone GPO management* and *Manage GPO using ST25PC-NFC software*)
- RF User: requires the use of a specific reader command, that can be sent either by a smart phone running the ST25 NFC application or by a reader (see the instructions in *Smart phone GPO management* and *Manage GPO using ST25PC-NFC software*)

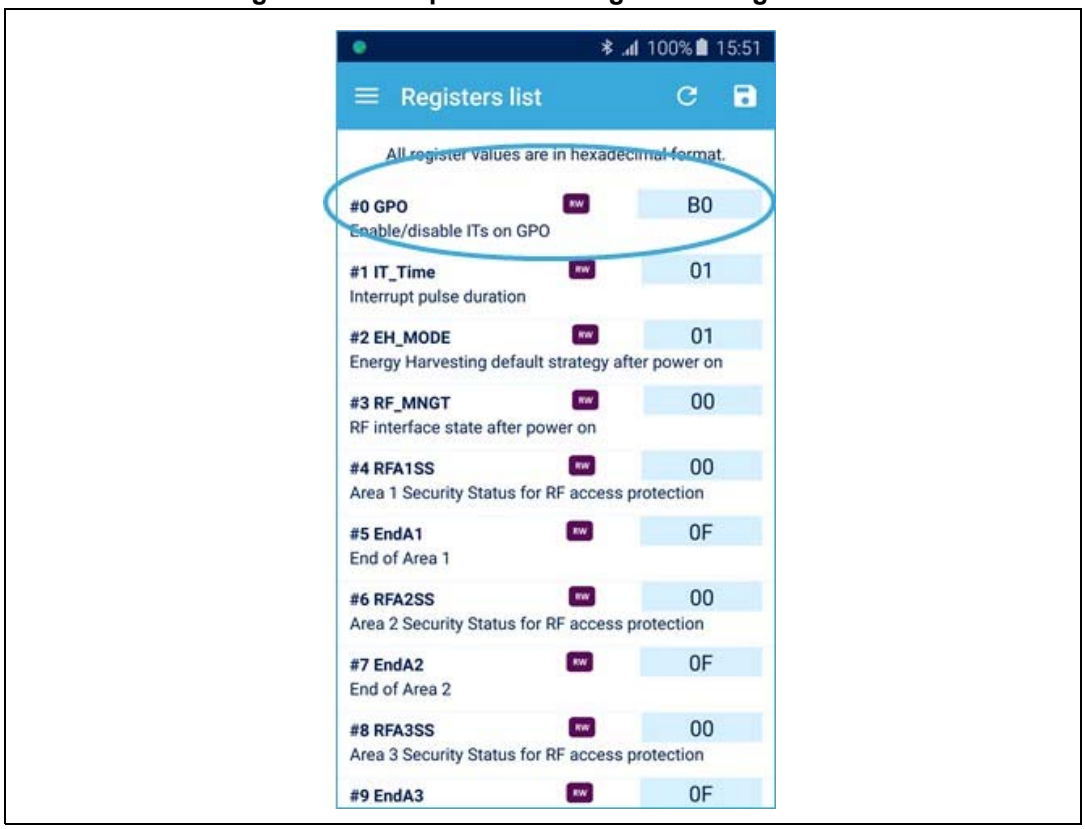
To get the ST25 NFC application, read the ST25DV with a smart phone (after entering the GPO demo), this automatically opens the required application (or propose the installation of the application). The automatic application selection is done using an NDEF with an Android™ application record (AAR).

Note: The mailbox events can also be used to trigger the GPO interrupts; they are not mentioned here, as they would require entering the ST25DV in the Mailbox mode.

Smart phone GPO management

The ST25 NFC application can handle the RF Interrupt and the RF user interrupt using register fields information. Do not forget to present configuration password before any register change.

Figure 7. Smart phone GPO register configuration



The ST25 NFC application can generate the RF Interrupt and the RF user interrupt using features located in the Tools fragment in association with the RF GPO interrupt demonstration.

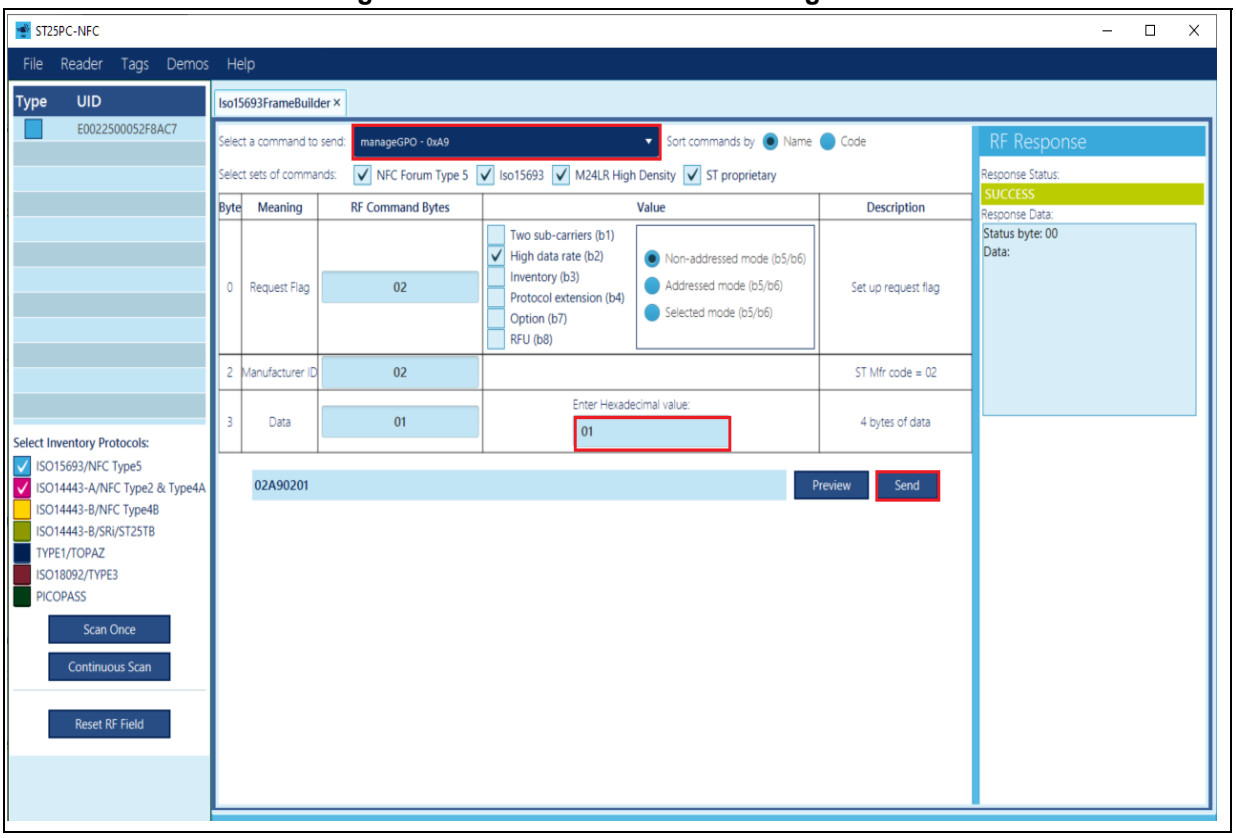
Manage GPO using ST25PC-NFC software

STMicroelectronics develops a PC software able to drive some dedicated NFC reader and communicate with NFC tags.

Using this software user is able to manage GPO and to send RF interrupts to the ST25DV-DISCOVERY. *Figure 8* show the use of ManageGPO command.

For additional information about ST25PC-NFC software, Refer to UM2444 (Software toolbox for NFC tags) for more information about it, available on *www.st.com*.

Figure 8. Use of PC software to manage GPO



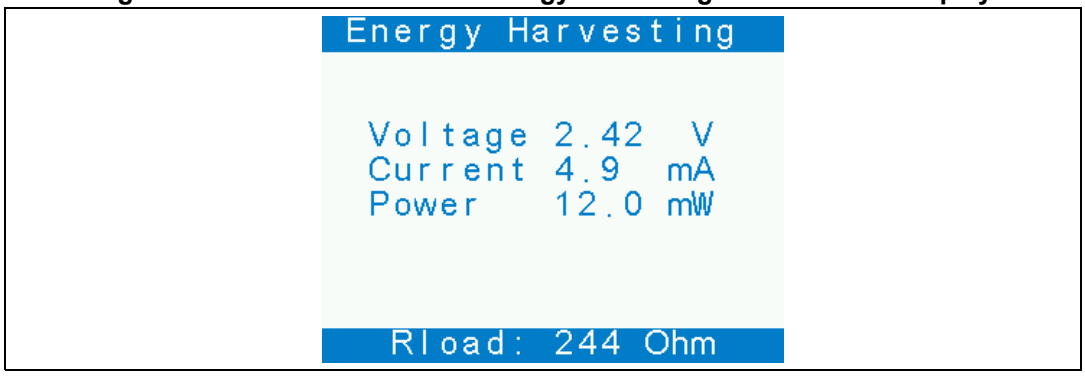
ST25DV energy harvesting demonstration

The ST25DV is able to harvest the energy provided by the RF to power other devices.

This demonstration displays the voltage, current and power provided by the ST25DV. A led is also powered-up to simulate the effect of the energy harvesting when the RF field is approaching or when it leaves.

The current consumption is simulated by a digital potentiometer set to 240 Ohm (plus the measurement resistor).

Figure 9. ST25DV-DISCOVERY energy harvesting measurement display



ST25DV states demonstration

This menu illustrates the possibility to change the state of the ST25DV (for example, to save power). This demonstration proposes three different modes:

1. RF disabled mode: RF commands are interpreted but not executed. ST25DV responds with the error code 0x0F
2. Sleep mode: all RF communication are disabled, the RF interface does not interpret the commands, but minimizes consumption of RF interface
3. Low power down: the ST25DV is fully inactive from host-I^2C side, but the memory content can be accessed on the RF side

For all above cases, the demonstration consists in writing an NDEF with the ST25 URL and changing the ST25DV state. A message is displayed that invites the user to try to read the content of the ST25DV, and check that it is not possible.

The content of the ST25DV can be read again as soon as the demonstration screen has been passed.

Note: For the Low power mode, the ST25DXSPARE jumper must be present on the ST25DV-DISCOVERY board on the MCU pin, and a 12-pin package must be used for the ST25DV.

ST25DV multi area and password demonstration

The ST25DV can be configured to define up to four different areas in memory. Each area may have a custom security level, requiring one of the three passwords to be provided in order to read and/or write the memory.

This demonstration sets two different areas:

- Area1 contains an NDEF with a vCard. This area is readable by anyone but can only be written after presenting the password 1.
- Area2 contains an NDEF with a different vCard. This area cannot be read without the password 1 and cannot be written even after the password presentation.

By default, all the passwords are set to 0000 0000 0000 0000. These values can only be updated from the RF side.

To execute the demonstration, the user must first read the ST25DV without any specific application. The phone should display the vCard stored in Area1.

Then the user can open the ST25 NFC application, and tap the tag again, two areas are detected, and the Area1 vCard is displayed.

The user may try to write a different NDEF to the Area1, and check that the write fails unless password 1 is presented.

If the user selects the Area2 NDEF, the application requests the password1 before displaying the Area2 vCard.

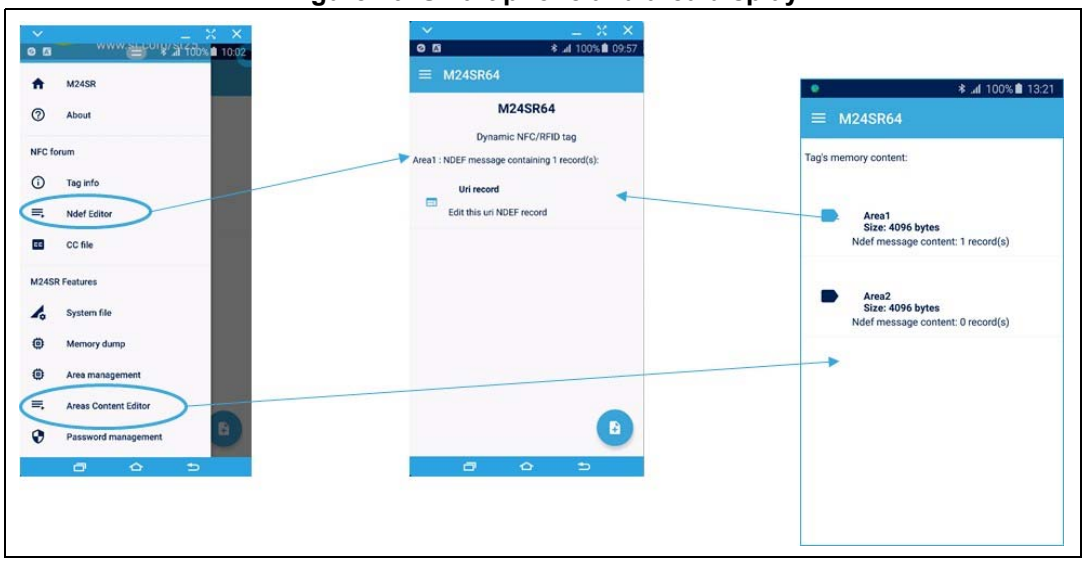
The user may try to write a different NDEF in Area2, this does not work even after presenting the password1. This is expected as the security level of Area2 prevents any write access to the memory.

Smart phone and multi areas

When the tag has been configured in multi areas, the ST25 NFC application can handle areas and present to the user areas and content. User can choose the desired area selecting the corresponding menu in the application drawer.

In case of special protections in an area, the application displays security status information. On click, a popup is used to present the area password needed to access area content.

Figure 10. Smart phone and area display



3.2.3 “Fast Transfer Mode Demo” menu

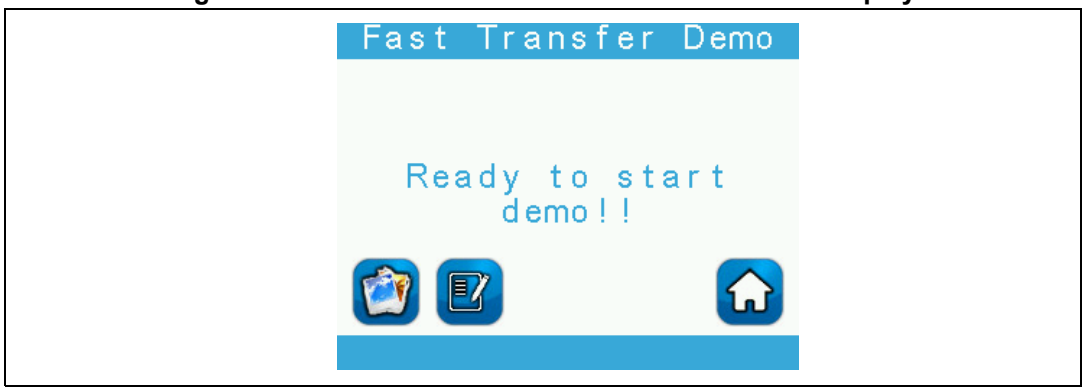
This demonstration is intended to demonstrate the capability of the ST25DV to exchange data between a reader and the MCU in a faster way, than exchanging data through EEPROM, using proprietary protocol through ISO 15693 standard. To perform the data exchange the ST25DV shares a RAM buffer of 256 bytes available for the reader and the MCU (also called Mailbox).

Data exchange can only be carried out in one direction at a time. The mailbox contains indicators that allow the reader to know its status and generate pulses on a GPO, which can be connected to an external interrupt (GPIO input) of the microcontroller. For more details on the Mailbox feature, refer to the ST25DV datasheet.

When the user selects the Fast Transfer Mode icon (see *Figure 4*), a message displays “Starting Demo…”. During this time, the firmware is initialized, enabling the Mailbox functionality and erasing the Flash memory on the MCU (in order to store data downloaded through NFC). Then, when the demonstration has finished its initialization the message “Ready to start demo!!” is displayed, as shown on *Figure 11*.

Note: When Mailbox feature is enabled, EEPROM memory is in read only. To modify EEPROM again you'll need to disable the Mailbox feature.

Figure 11. ST25DV-DISCOVERY Fast Transfer Mode display



The picture and note icons, at the bottom left, point to a sub-menu that allows the user choose a picture or a random data buffer, that is transferred to a reader (see *Host to reader image upload* and *Host to reader data transfer*).

The home icon (bottom right) allows the user to go back to the main menu (*Figure 4*).

At this step, the firmware is waiting for an action from the reader or from the user to start. These actions are described in the next sections.

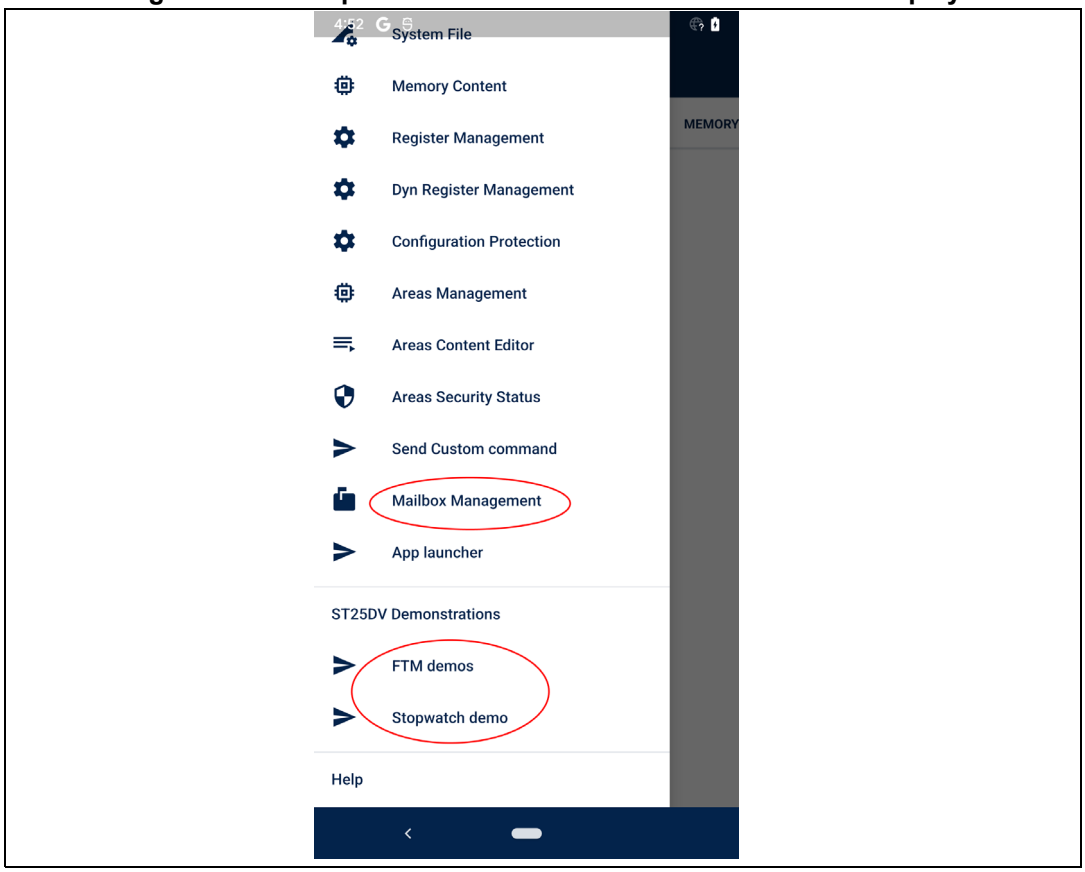
Smart phone and Fast Transfer Mode use cases

Available actions can be accessed after tag has been taped, and are available in the ST25DV features or demonstrations menus within the application “Drawer”.

Available use cases are:

- Mailbox management, enable/disable mailbox and display mailbox register fields status
- Data Transfer, to demonstrate basics transfers
- Firmware Upgrade, to demonstrate the firmware upgrade use case
- Picture transfers, to demonstrate how to upload or download pictures
- Stopwatch synchronization, to demonstrate fast transfer latency with a smart phone

Figure 12. Smart phone and Fast Transfer Mode use cases display

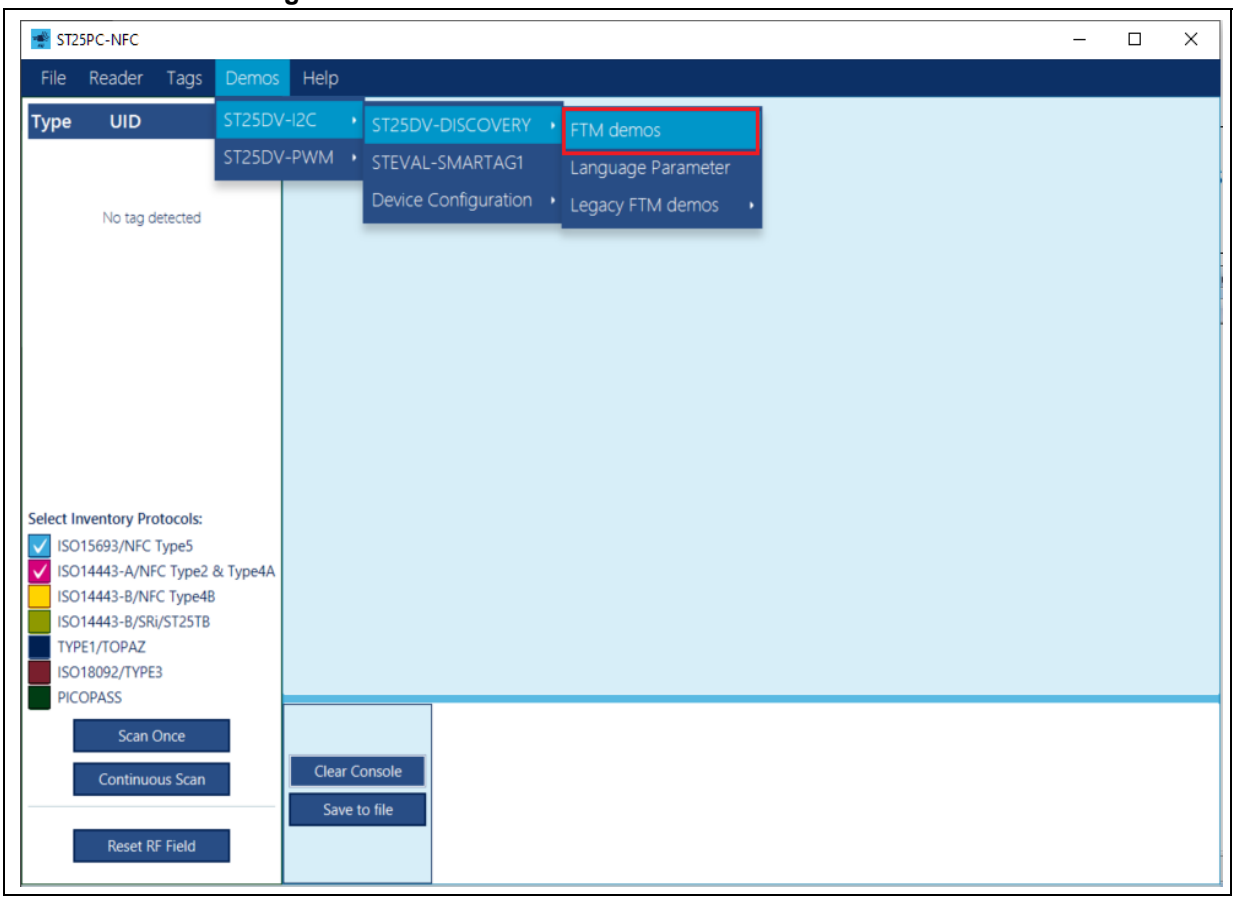


PC software and Fast Transfer Mode use cases

As with Android™ phones, all the Fast Transfer Mode demonstrations can be played using ST25PC-NFC software delivered by STMicroelectronics and able drive some NFC reader to communicate with NFC tags. Refer to UM2444 (Software toolbox for NFC tags) for more information about it, available on *www.st.com*.

Select “FTM Demos”, in ST25DV-DISCOVERY menu to access to “FTM demos”.

Figure 13. FTM demonstration selection on ST25PC-NFC



The *Figure 14* shows the user interface used to play every FTM demonstrations.

Figure 14. FTM demonstration user interface

Available demonstrations are:

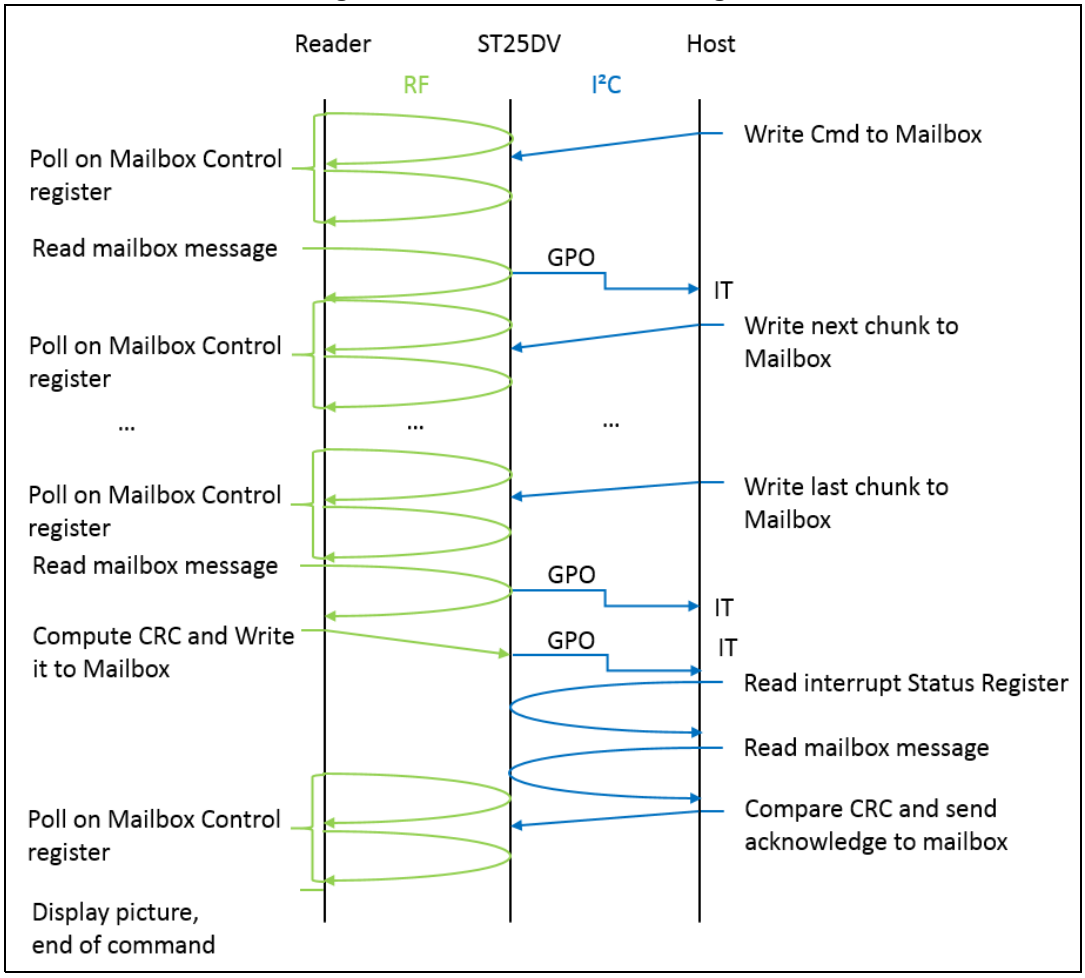
- Check board version (R2H)
- Firmware Upgrade demo (R2H)
- Send a picture: select a picture to be sent to the ST25DV-DISCOVERY board using FTM protocol (R2H)
- Receive a picture: select a picture on the ST25DV-DISCOVERY board and transfer it on PC user interface using FTM protocol (H2R)
- Send a 100-kbyte random buffer: send data to the ST25DV-DISCOVERY board (R2H)
- send a file: send a selected file to the ST25DV-DISCOVERY board (R2H)
- Receive a buffer in a file: select the data size on the ST25DV-DISCOVERY and transfer it to a selected file in user's computer using FTM protocol (H2R)
- Stop watch demo: demonstrate the synchronization between the PC and the DISCOVERY board using FTM protocol (R2H)

Reader to host data transfer

This function allows the user to send data from the reader to the STM32F405. In this demonstration data are not used after receipt, they are only stored in the Flash memory and CRC calculation is sent to the reader for data integrity check.

Figure 15 details the communication flow from the reader to the host (R2H) through the ST25DV during data transfer.

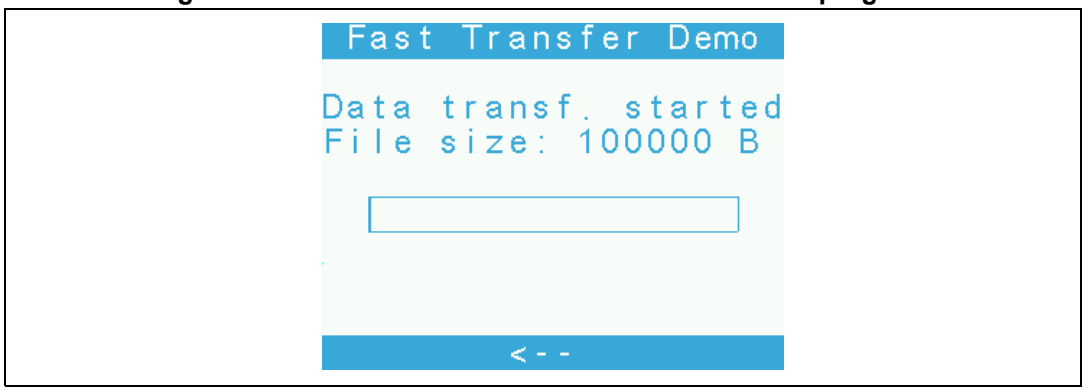
Figure 15. R2H data transfer diagram



To prepare the ST25 Discovery for the R2H data transfer demonstration, power on the board then touch the FTM icon to start the fast transfer mode demonstration (*Figure 11*). The kit is now waiting for a reader action (as can be seen in *Smart phone and Fast Transfer Mode use cases* and *PC software and Fast Transfer Mode use cases*) to continue the demonstration.

During the transfer the full length of data transmitted is displayed and a bar indicates the progress (see *Figure 16*).

Figure 16. ST25DV-DISCOVERY R2H data transfer in progress



When the transfer is successfully completed, the computed CRC and the transfer time are displayed on the screen (*Figure 17*). In case of failure an error message is displayed.

Figure 17. ST25DV-DISCOVERY R2H data transfer done

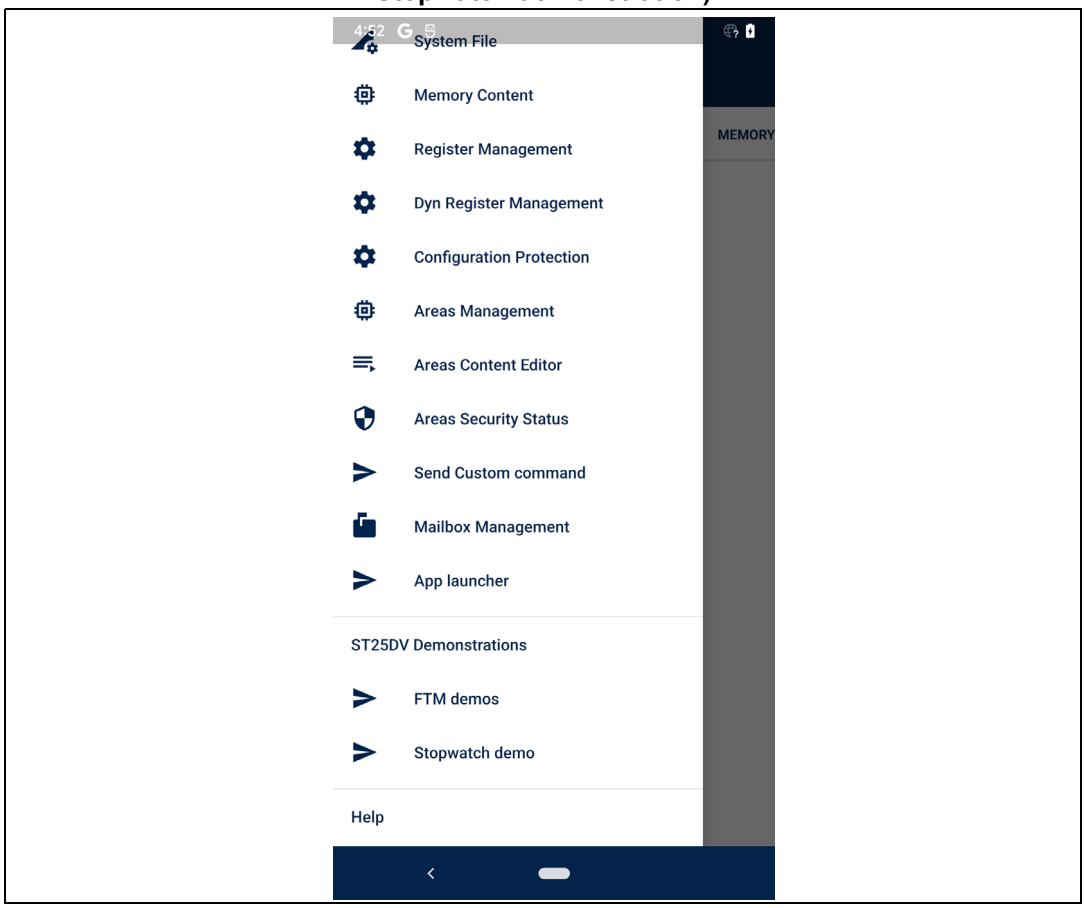
To return to the Fast Transfer Mode demonstration main screen simply touch the screen.

Smart phone and FTM data transfers

Several menus accessible by the drawer menu are dealing with the fast transfer mode data transfers:

- Mailbox management. Radio buttons give the current status of the mailbox, and this menu permits to enable/disable the mailbox feature.
- FTM demonstrations:
  - Send or receive data
  - Firmware update.
  - Send or receive pictures
- Stopwatch transfers: This demonstration shows a chronometer synchronized between the smart phone and the tag using the mailbox. It shows the low latency.

Figure 18. ST25DV drawer menu (Mailbox management, FTM demonstration, Stopwatch demonstration)



As prerequisite to any data transfers the mailbox has to be enabled, without any pending message. The correct status is shown in *Figure 19*.

Figure 19. Mailbox Status

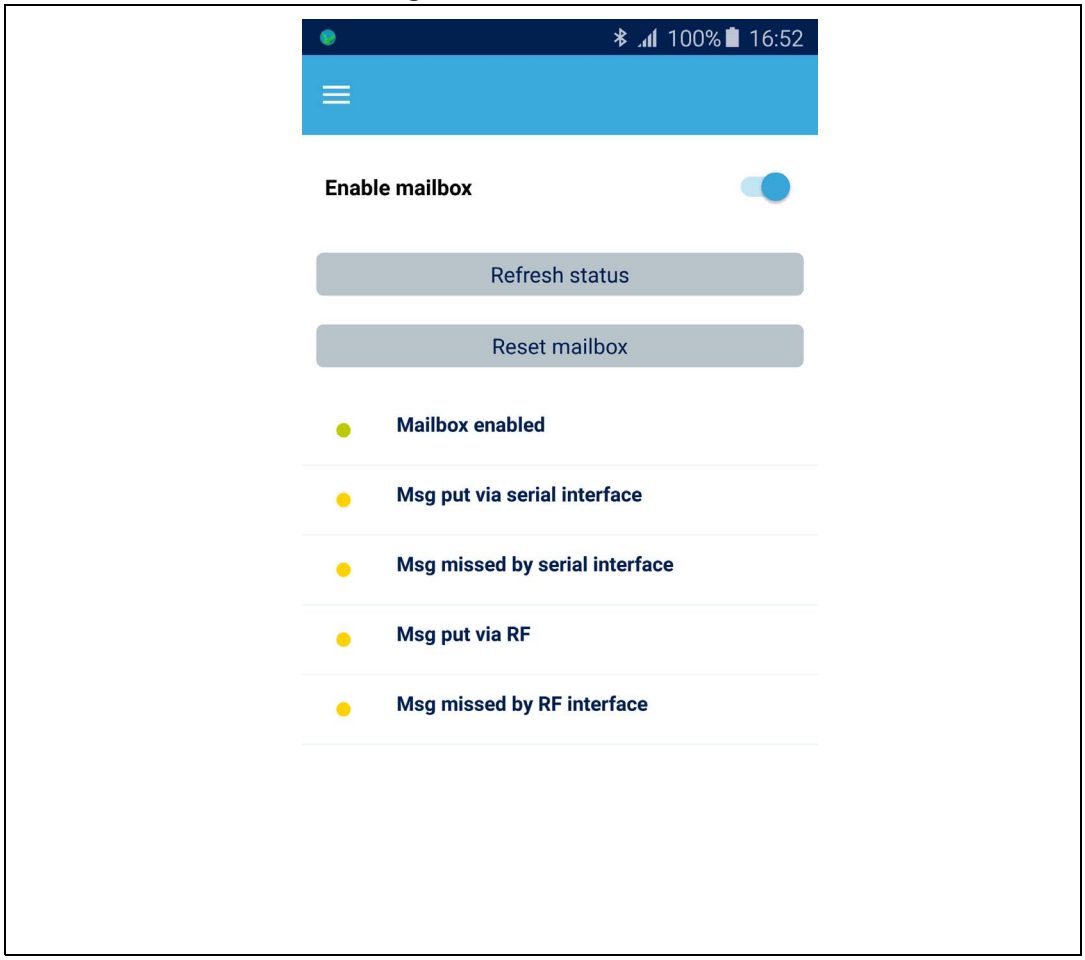
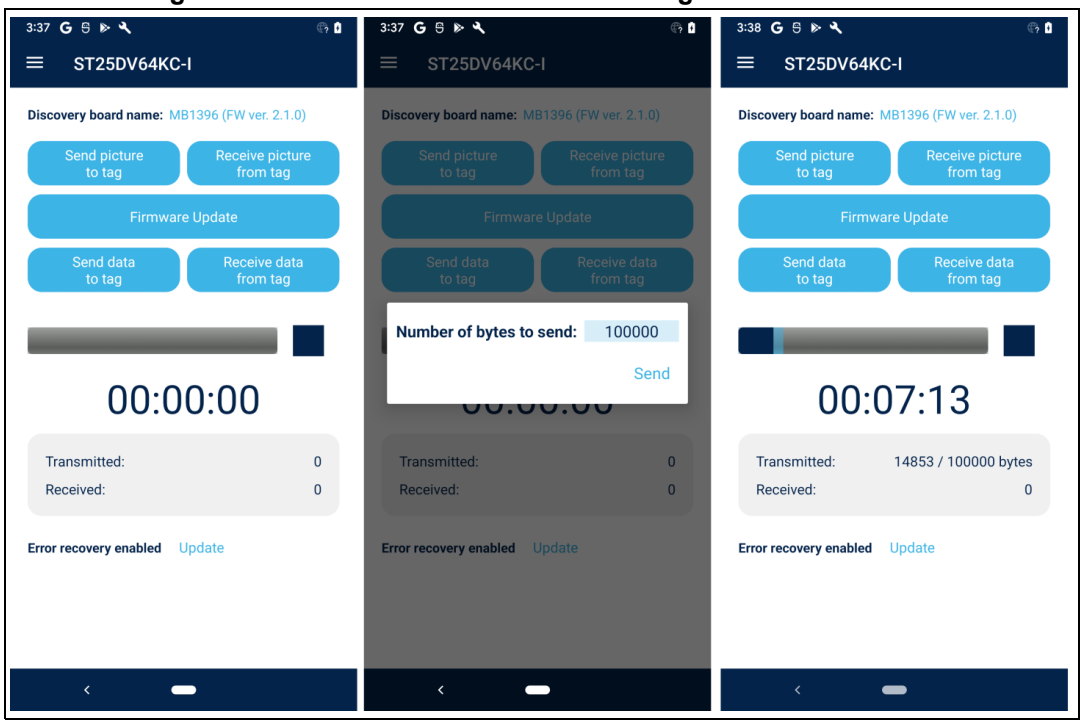


Figure 20. Data transfers from reader to tag of a random buffer



Note: According to data size selected, the corresponding payload data is initialized with random values on which a CRC is computed to be sure that host computes the same data after transfers.

Note: For every transfers done with the Android™ application, the user can chooce between two modes:

- Error recovery enabled. With this mode, the protocol is able to detect transmission errors and to recover (by retransmitting the faulty packets). This is the default (and recommended) mode.
- Error recovery disabled. Data transfer is done at max throughput. There is no recovery in case of transmission error.

Reader to host firmware upgrade

This function allows the user to send data (that is manipulated as binary) from the reader to the STM32F405. They are stored in the Flash memory and at the end of transfer, the following instruction code jumps to the new firmware location, demonstrating the capability to transfer many binary data. For this example, the new firmware is stored in a secondary area in the Flash memory and it is not intended to be used for new demonstrations. To come back to the right first firmware, press the reset button and it reboots on it.

The firmware upgrade transfer flow is similar to the one of data transfer (see *Figure 15*).

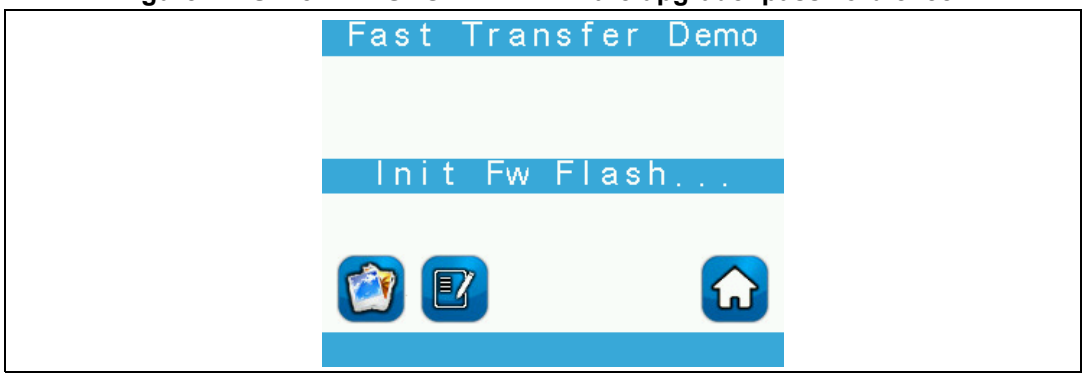
To prepare the ST25 Discovery for the firmware upgrade demonstration, power on the board then touch the FTM icon to start the fast transfer mode demonstration (*Figure 11*).

The kit is now waiting for a reader action (as can be seen on *Smart phone and firmware upgrade* and *PC software and firmware upgrade*) to continue the demonstration. For this demonstration the firmware dedicated to this demonstration can be found on the project

folder ST25DV-Discovery\Demonstrations\ST25DVDemo_FwUpgd\Binary. A demonstration firmware is embedded in the Android™ application and can be used straight away.

To be able to execute the firmware download the user first needs to send (from the reader) a correct password (for this demonstration the password is 0x12345678). During the password check, the Flash memory area where the firmware is stored is erased (*Figure 21*).

Figure 21. ST25DV-DISCOVERY firmware upgrade: password check



If the password is correct the user is authorized to start the transfer (*Figure 22*), if not, the user must enter a new password to continue (*Figure 23*).

Figure 22. ST25DV-DISCOVERY firmware upgrade: password OK

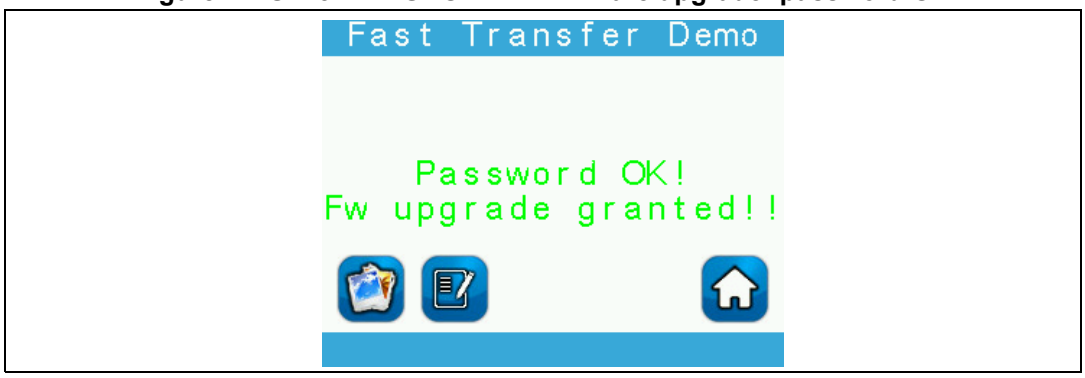
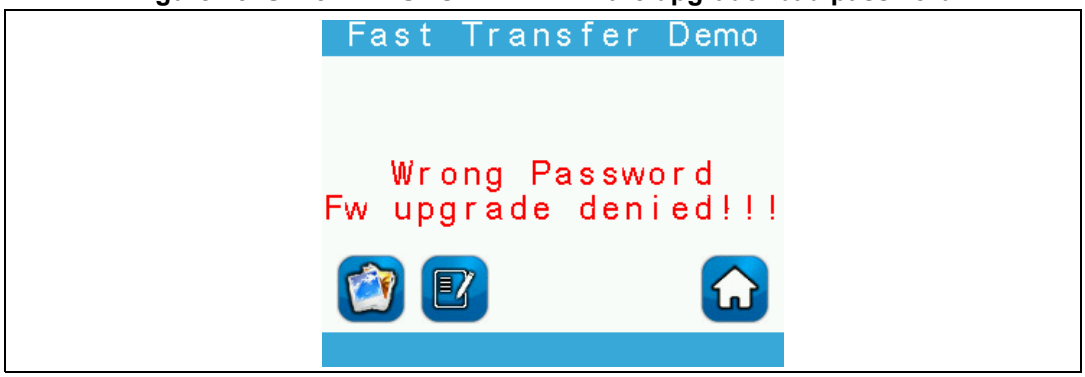
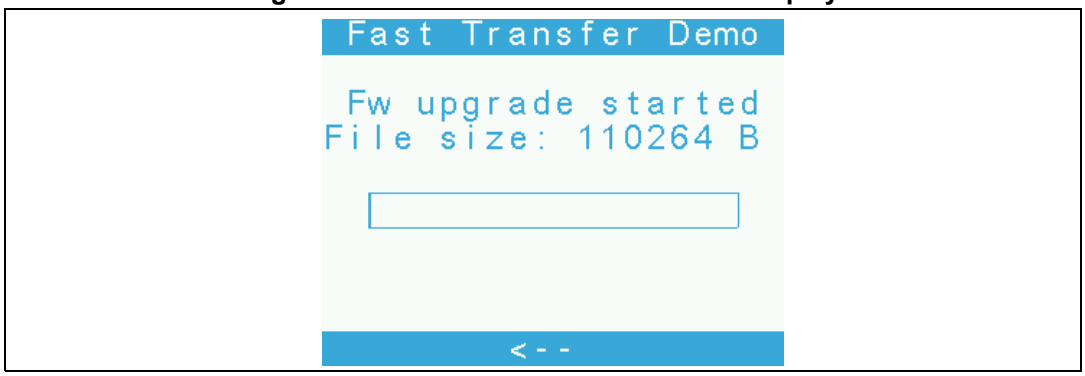


Figure 23. ST25DV-DISCOVERY firmware upgrade: bad password



During the transfer the full length of data transmitted is displayed, and a bar indicates the progress (see *Figure 24*).

Figure 24. Software end transfer status display



When the transfer is successfully completed, the computed CRC and the transfer time are displayed (see *Figure 25*). In case of failure an error message is displayed.

Figure 25. ST25DV-DISCOVERY firmware upgrade: transfer done

To start the new firmware, simply touch the screen.

Smart phone and firmware upgrade

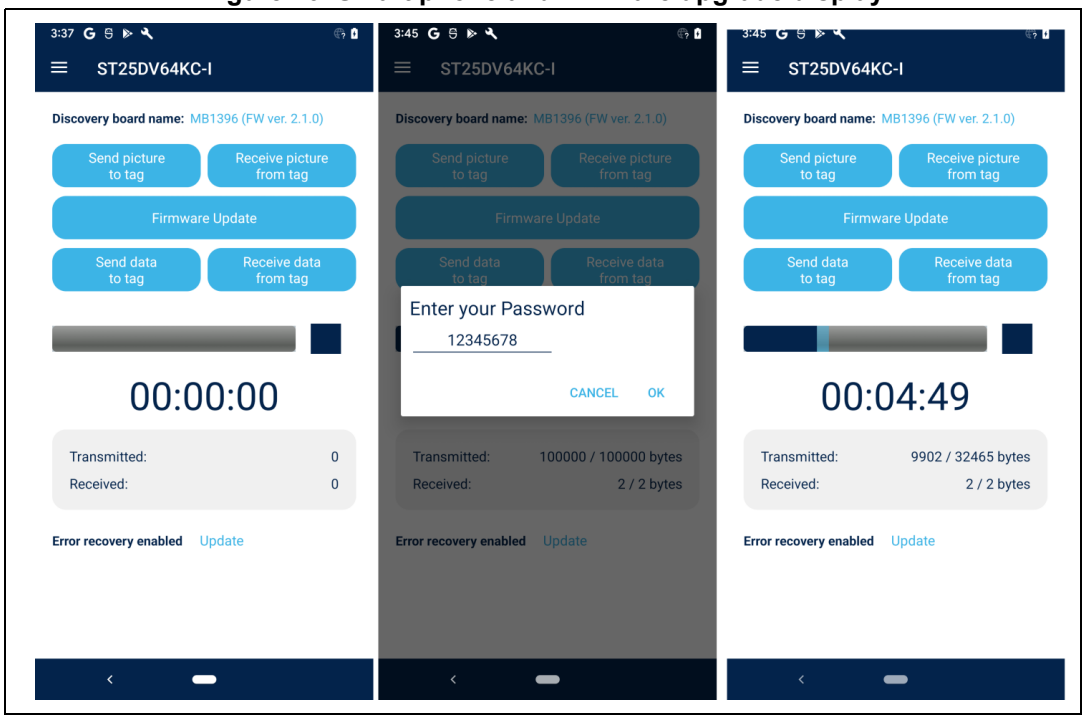
Features:

- Upgrade a new firmware to the ST25 discovery board

Demonstration steps:

- Check the mailbox status (Must be enable, withtout any pending messages in the mailbox).
- Select the “Firmware demos” in the drawer menu and click on “Firmware update button”.
- Select the file to be downloaded.
- A pop up asks for a password. By default, and for demonstration purpose the password is 12345678.
- Start download = Start transfer button.
- Smart phone and firmware download.

Figure 26. Smart phone and firmware upgrade display

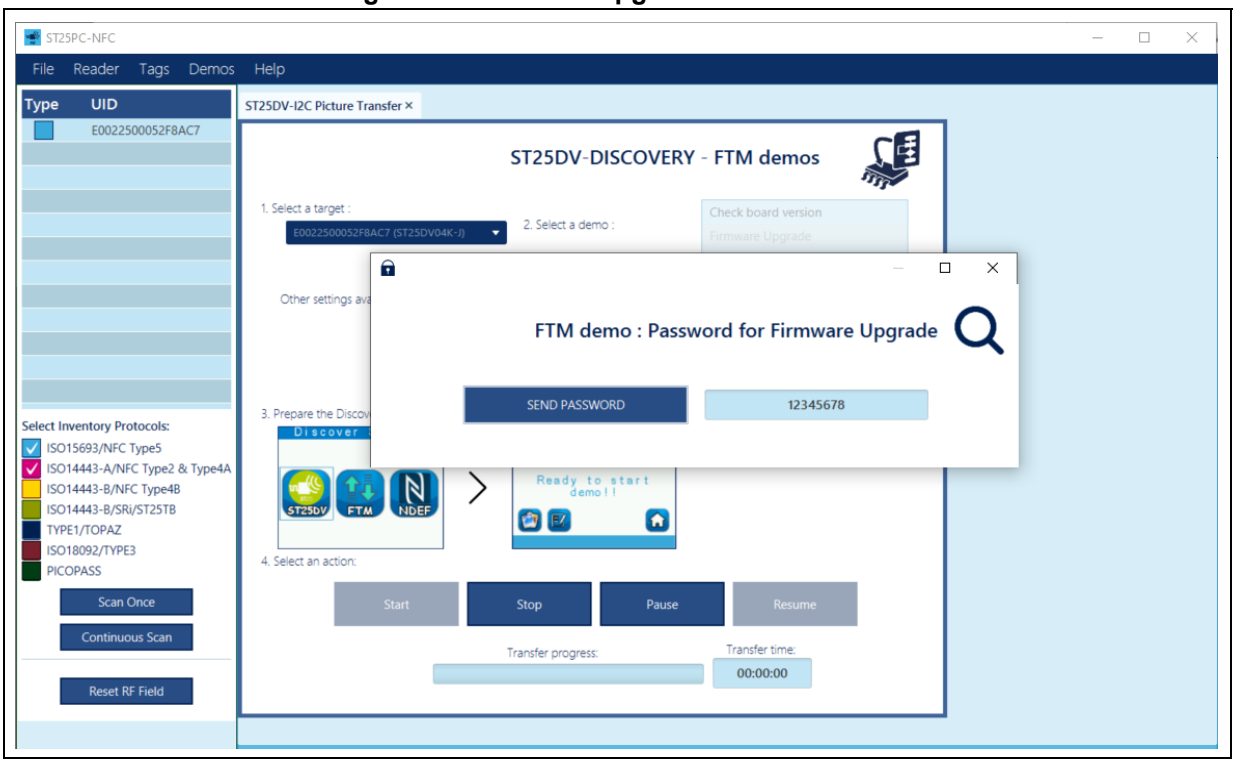


PC software and firmware upgrade

Select “Firmware Upgrade” demonstration, then select a binary file to be downloaded to the ST25DV-DISCOVERY board.

In order to start firmware upgrade, the user needs to select a password, then to click on “Start” button to start transfer to the board.

Figure 27. Firmware upgrade demonstration



Reader to host image download

This function allows sending data from the reader to the STM32F405. In this demonstration data are jpeg images, stored in the Flash memory and displayed when transfer is successful.

The image download transfer is he same as that the data transfer flow shown in *Figure 15*.

To prepare the ST25 Discovery for the R2H picture transfer demonstration, power on the board, then touch the FTM icon to start the fast transfer mode demonstration (*Figure 11*).

It is now waiting for a reader action, as can be seen on *Smart phone and picture transfers* and *Smart phone and picture download transfer* to continue the demonstration.

During the transfer the full length of data transmitted is displayed and a progress bar indicates the progress. When transfer is done and successful, computed CRC and transfer duration are displayed on screen. To display the downloaded picture, touch the screen. To return to the fast Transfer Mode demonstration simply touch the screen.

Displayed screens are similar to those of data transfer (see *Reader to host data transfer*).

Smart phone and picture transfers

Features:

- Send a picture to the tag.
- Receive a picture from the tag.

Picture upload steps:

- Check the mailbox status (c.f. drawer menus).
- Select “FTM demos” in drawer menu.
- Click on “Send picture to tag” Select a picture in phone’s memory.

Figure 28. Transfer from smart phone to tag

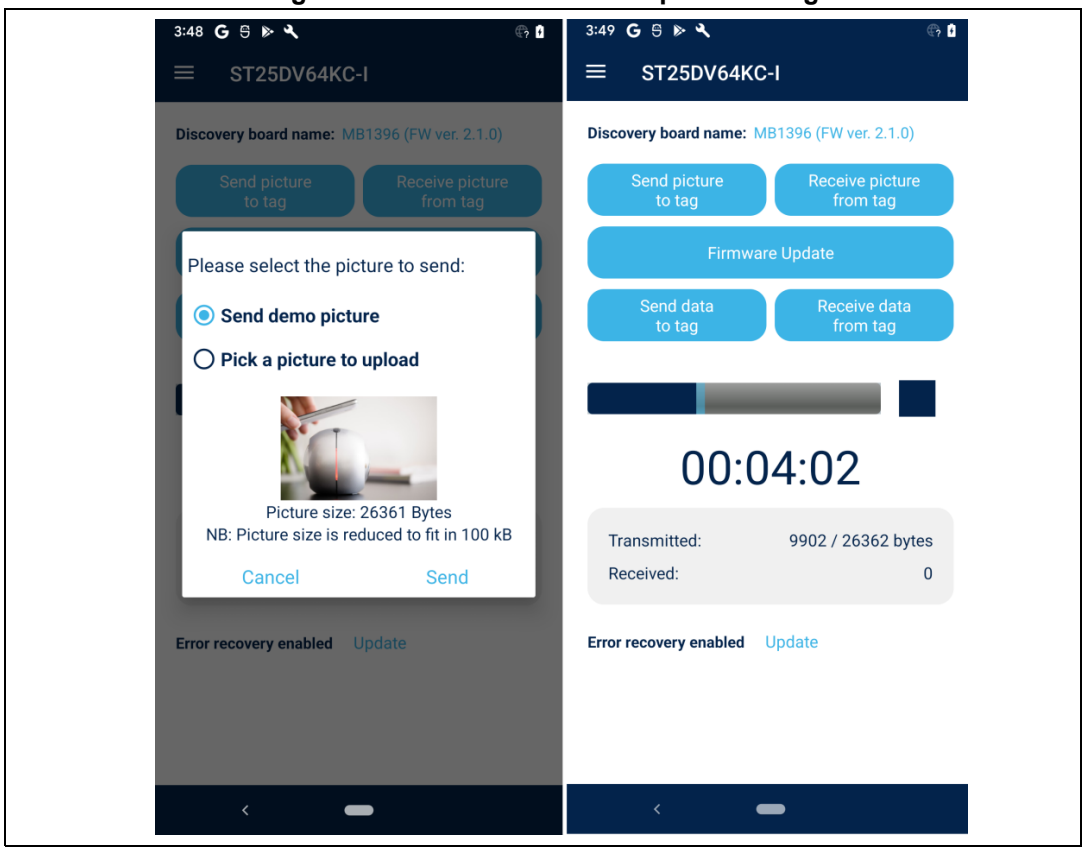


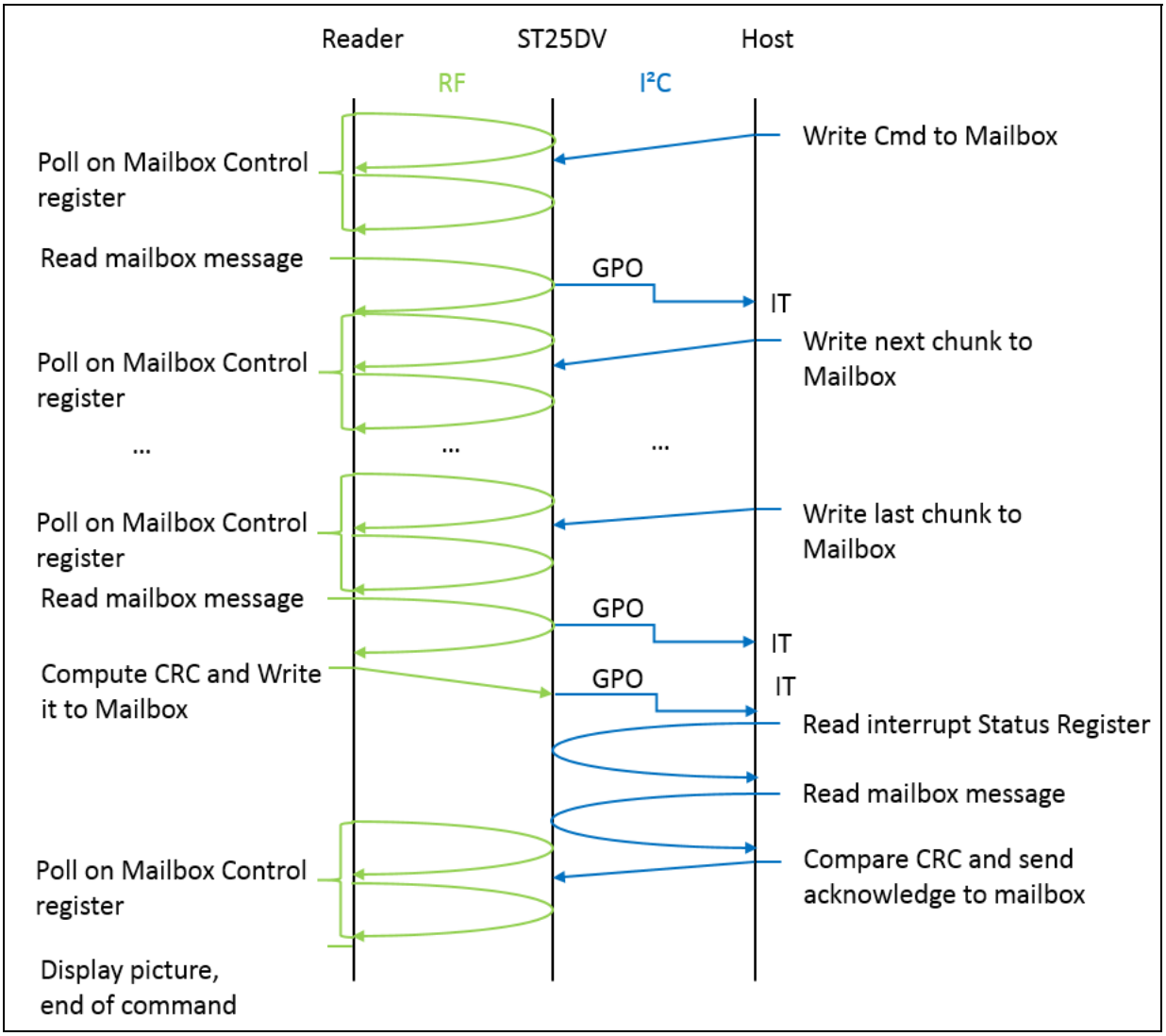
Figure 29. Image transfer demonstration: send a picture

Host to reader image upload

This function allows the user to send data from the STM32F405 to the reader (H2R).

In this demonstration data are jpeg images, data are written to the mailbox each time this is free, and the MCU is informed by the GPO when the mailbox message is read by the reader. *Figure 30* represents the transfer between the MCU and the reader.

Figure 30. H2R data transfer diagram



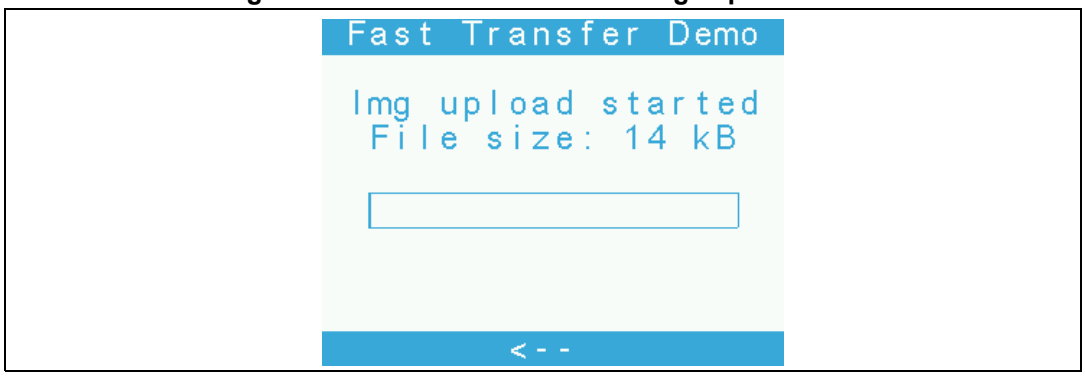
To start this demonstration the user needs to perform an action on firmware side.

The picture icon at bottom left allows the user to enter a new menu (see *Figure 31*), which lets the user choose between pictures to send to the reader. By touching the screen on the left or right border user see different pictures available for transfer, simply touch the image on screen to select it. To come back and cancel the action, touch the arrow at the bottom of the screen.

Figure 31. ST25DV-DISCOVERY Select picture to upload

After selecting the picture to transfer, the firmware writes the first chunk in the mailbox and waits for the message to be read. During the transfer a file size and a progress bar are displayed, as shown in *Figure 32*.

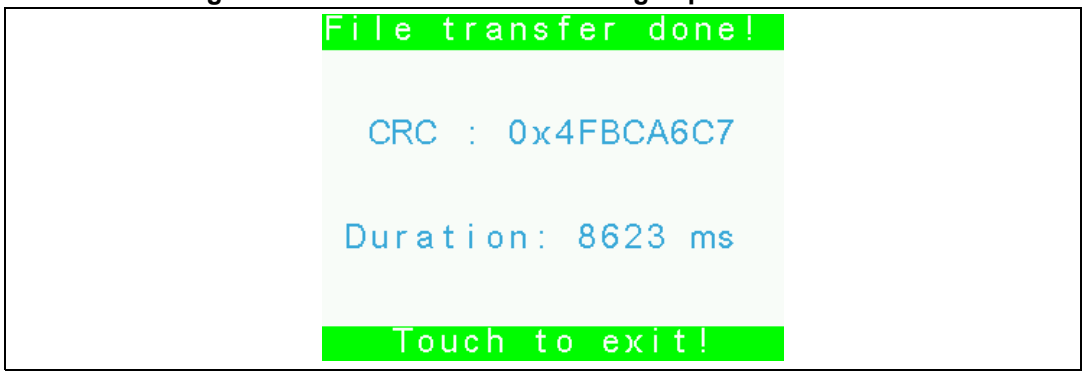
Figure 32. ST25DV-DISCOVERY Image upload start



The kit is now waiting for the reader to read the message on the mailbox (as can be seen in *Smart phone and picture download transfer* and *PC software and picture upload transfer*) to continue the demonstration.

When transfer is successfully completed, computed CRC and transfer time are displayed on screen (*Figure 33*). In case of failure an error message is displayed.

Figure 33. ST25DV-DISCOVERY Image upload successful



To return to the Fast Transfer Mode demonstration simply touch the screen.

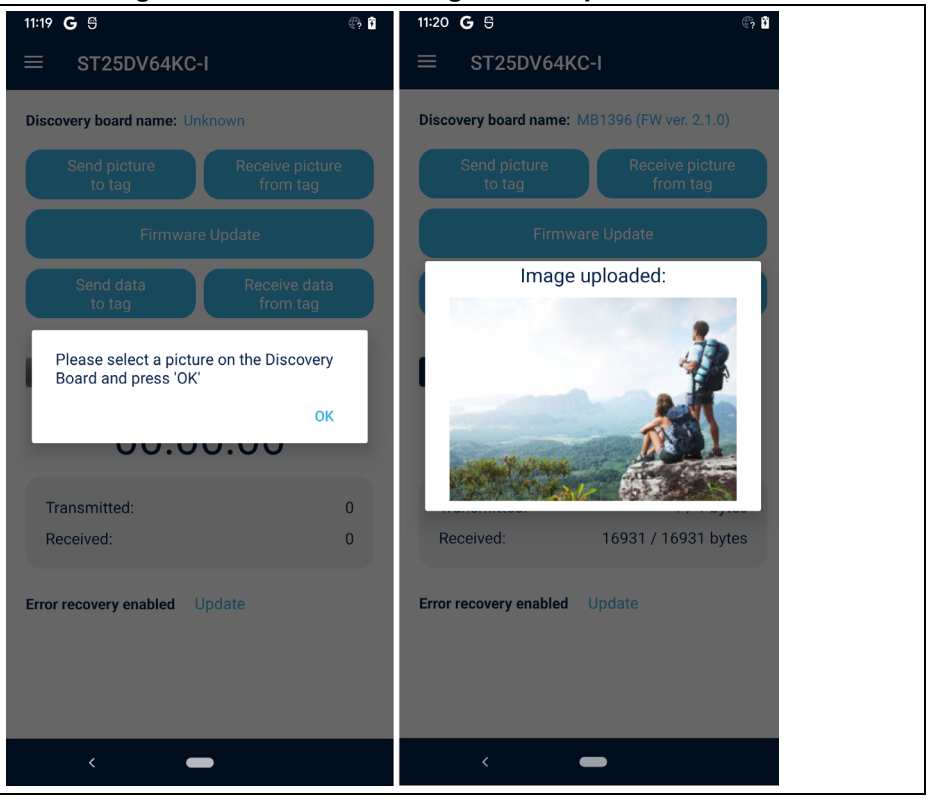
Smart phone and picture download transfer

Features: same as *Smart phone and picture transfers*.

Picture download steps:

- Check MB registers
- Select “FTM demos” in the drawer menu
- Received picture from tag button. Select a picture on the ST25 discovery kit. The transfer starts.

Figure 34. Transfer from tag to smart phone

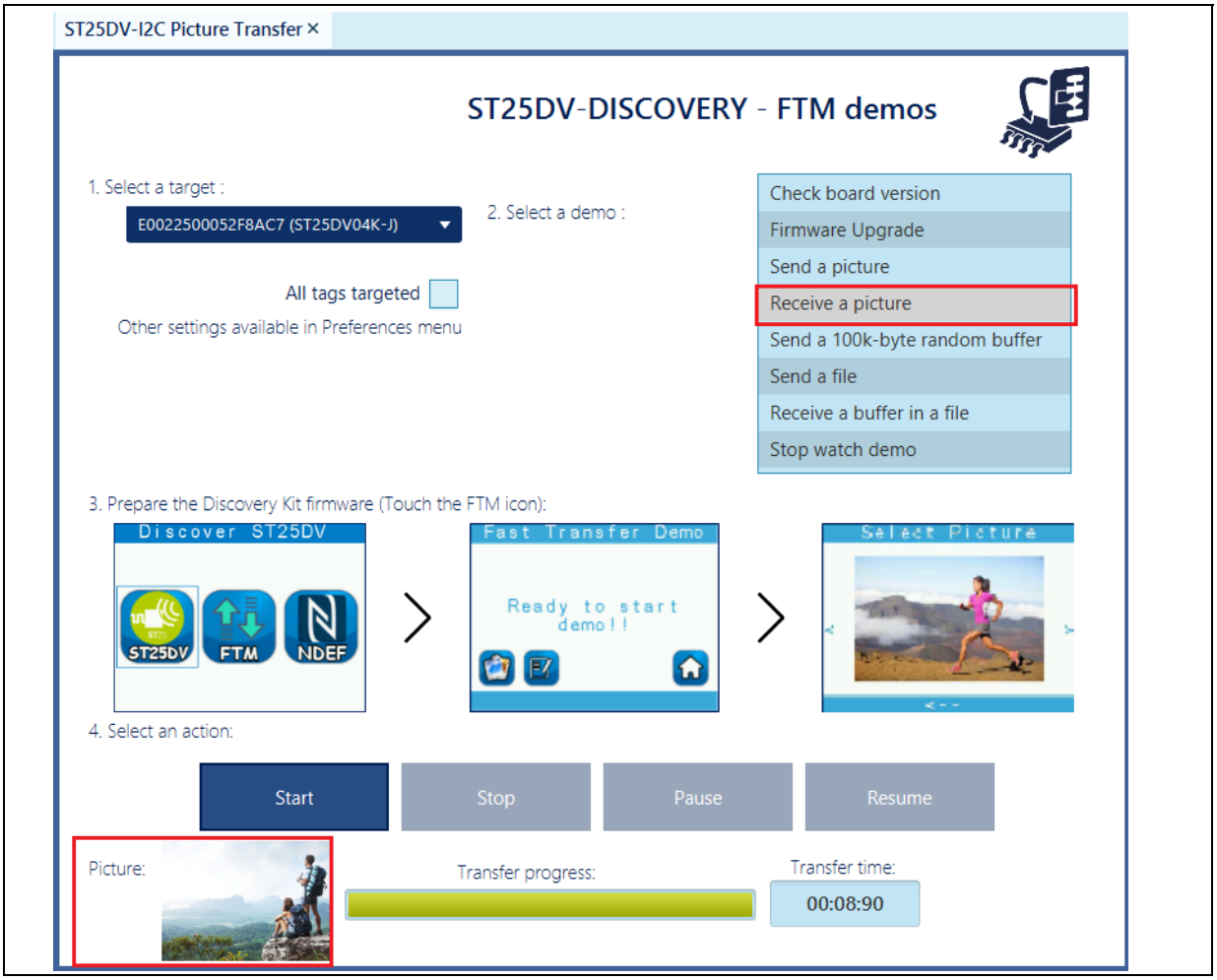


PC software and picture upload transfer

Select “Receive a picture” demonstration, after having selected a picture from ST25DV-DISCOVERY board user can click on “Start” button to start transfer from the ST25DV-DISCOVERY board.

The demonstration ends when the PC software receives the acknowledge and displays the image.

Figure 35. Image transfer demonstration: receive a picture



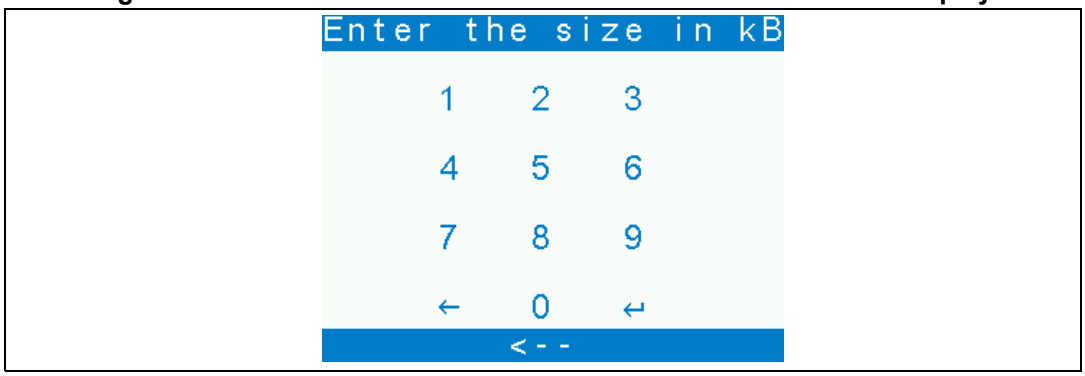
Host to reader data transfer

This function allows sending data from the STM32F405 to the reader. In this demonstration data are random binary data, data are written to the mailbox each time it is free, and the MCU is informed, via the GPO, when the mailbox message is read by the reader

Figure 30 shows the transfer between the MCU and the reader (the flow is identical to image upload).

To start this demonstration the user needs to perform an action on firmware side. The data icon at bottom left allows the user to enter a new menu, this menu allows selecting any size of data transfer (from the MCU to a reader) from 1 to 999 kB, using the keyboard shown in *Figure 36*. The arrow on the bottom left allows the user to correct its input. To cancel and return to the Fast Transfer Mode demonstration touch the arrow at the bottom of the screen. After having entered the correct size, touch the return icon.

Figure 36. ST25DV-DISCOVERY select H2R data size to transfer display



The kit is now waiting for a reader to read the message on the mailbox (as can be seen from *Smart phone and H2R data transfer* and *Smart phone and H2R data transfer*) to continue the demonstration.

When transfer is successfully completed, computed CRC and transfer time are displayed (see *Figure 33*). In case of failure, an error message is displayed.

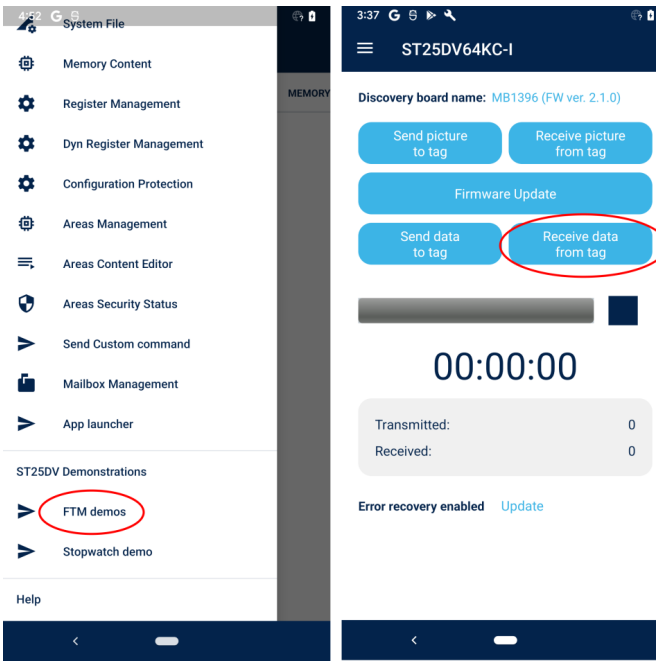
To return to the Fast Transfer Mode Demo simply touch the screen.

Smart phone and H2R data transfer

To start the demonstration after prerequisites setting checked, go to “FTM demos” menu.

After having selected the buffer size to send from ST25DV-DISCOVERY board, user can click on “Receive data from tag” button to start the transfer.

Figure 37. Smart phone and data transfers H2R with selected size



PC software and H2R data transfer

Select “Receive a buffer in a file” demonstration, after the selection of the buffer size to be sent from ST25DV-DISCOVERY board user can click on “Start” button to start the transfer of data from the board.

Figure 38. H2R data transfer demonstration

Stopwatch demonstration

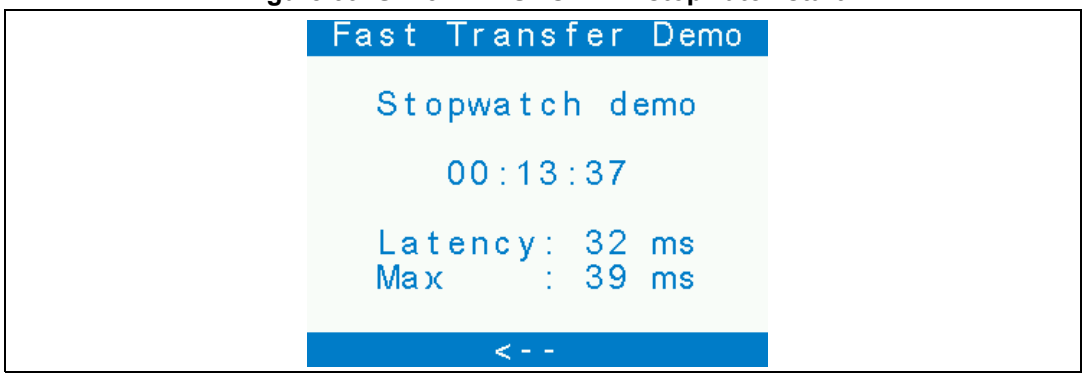
This demonstration shows how fast a smart phone can update the mailbox.

The smart phone sends, as fast as possible, stopwatch values (8-byte frames) to the MCU using the mailbox. The MCU displays these values on the LCD screen and computes the duration between two consecutive updates. The average and maximum inter-frame durations are refreshed and displayed after 20 frames have been received.

To run the demonstration

1. On the ST25 discovery board: enter the Fast Transfer Mode menu.
2. On the smart phone: open the ST25 NFC application, tap the ST25DV and go to the “Stopwatch Transfers” menu and click on Start button.

Figure 39. ST25DV-DISCOVERY stopwatch start



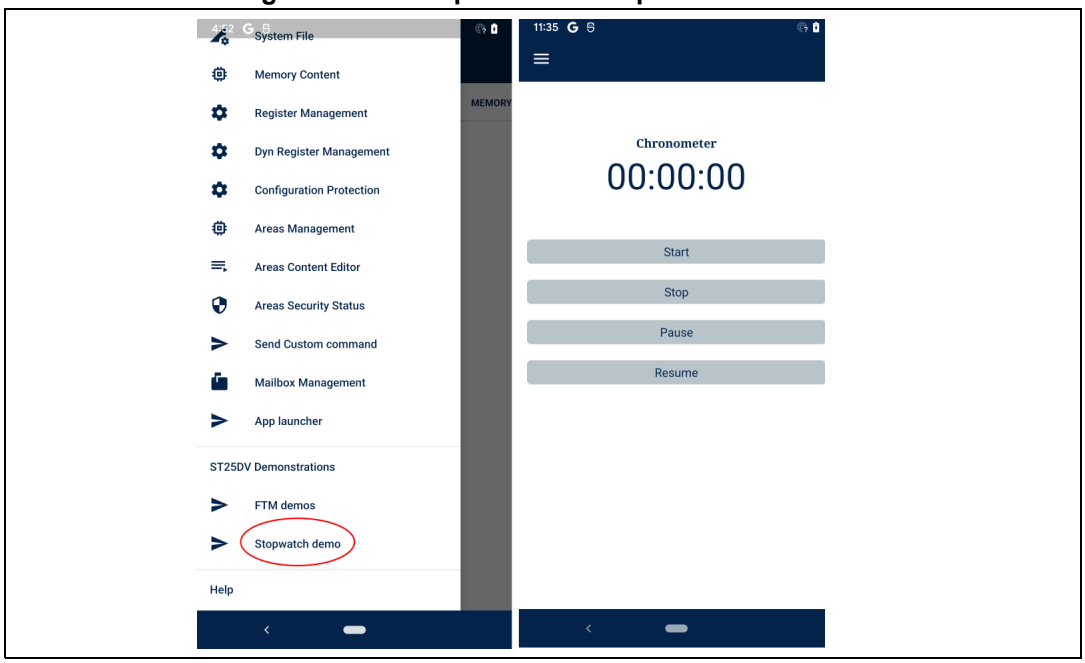
Smart phone and stopwatch transfers

In this demonstration the smart phone sends, as fast as possible, stopwatch values (8-byte frames) to the MCU using the mailbox. The main features are

- Configuration of the “Mail Box” register [Enable / Disable MB, Watchdog)
  - Selection of static or dynamic registers
  - Read Cfg to retrieve MB status
- Start: Starts the stopwatch, the display is updated accordingly, and frames are sent to the host if a tag is in the field. The display is updated, starting from 00:00:00 (minutes:seconds:milliseconds)
- Stop: Stops the stopwatch. When restarted, the stopwatch restarts from 00:00:00.
- Pause: Pauses the stopwatch
- Resume: resumes the stopwatch after a pause

On the smart phone open the ST25 NFC application, tap the ST25DV and go to the “Stopwatch Transfers” menu and click on Start button. Pause and Resume buttons can then be used.

Figure 40. Smart phone and stopwatch transfers



3.2.4 NFC NDEF demonstrations menu

The NFC NDEF demonstrations menu proposes several demonstrations concerning NDEF messages:

- URI NDEF message (URL and phone number)
- SMS NDEF message
- Email NDEF message
- vCard NDEF message
- Geo-location NDEF message
- Custom (MyApp) NDEF message
- Multi record NDEF message
- Bluetooth® Low Energy OOB NDEF message (requires an additional BLE module)
- Wi-Fi® OOB NDEF message (requires an additional Wi-Fi® module)

Figure 41. ST25DV-DISCOVERY NFC NDEF demonstrations menu display

URI demonstration

This menu shows how to manage URI content in the ST25DV.

With the MCU, the user can:

- Store an NDEF message containing an URL or a phone number (it can then be read with an NFC reader or a smart phone)
- Read any URI that has been stored in the ST25DV. Content of the URI is displayed on the screen.

Figure 42. ST25DV-DISCOVERY read URI display

Note: If, when the MCU reads the ST25DV memory, it does not contain a URI, an error message is displayed.

Smart phone and URI NDEF

Refer to *Reading and writing NDEF on a smart phone with the ST25 application*.

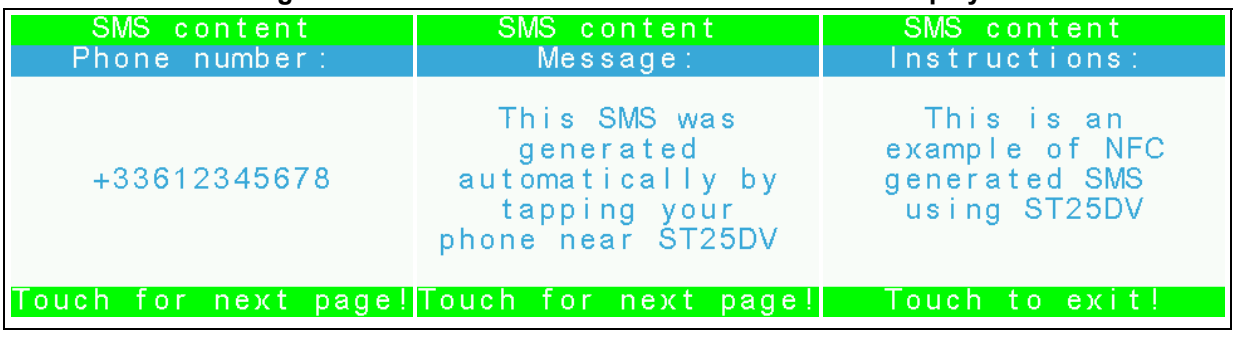
SMS demonstration

This menu presents how to manage SMS content in the ST25DV.

With the MCU, the user can

- Store an NDEF message containing a SMS (it can then be read with an NFC reader or with a smart phone)
- Read any SMS that has been stored in the ST25DV memory (the SMS is displayed on the screen)

Figure 43. ST25DV-DISCOVERY read SMS content display



Note: If, when the MCU reads the ST25DV memory, it does not contain a SMS, an error message is displayed.

Smart phone and SMS NDEF

Refer to *Reading and writing NDEF on a smart phone with the ST25 application*.

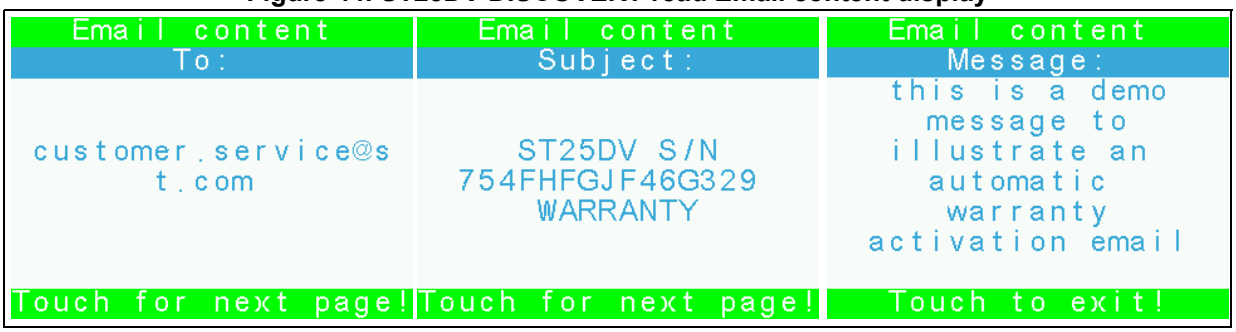
Email demonstration

This menu presents how to manage e-mail content in the ST25DV.

With the MCU, the user can

- Store an NDEF message containing an e-mail (it can then be read with an NFC reader or with a smart phone)
- Read any e-mail that has been stored in the ST25DV memory. The content of the e-mail is displayed on the screen.

Figure 44. ST25DV-DISCOVERY read Email content display



Note: If, when the MCU reads the ST25DV memory, it does not contain an e-mail, an error message is displayed.

Smart phone and e-mail NDEF

Refer to *Reading and writing NDEF on a smart phone with the ST25 application*.

vCard demonstration

This menu presents how to manage vCard content in the ST25DV.

With the MCU, the user can

- Store an NDEF message containing a vCard. Depending on the size of the ST25DV, this vCard may or may not embed a picture
- Read the ST25DV content

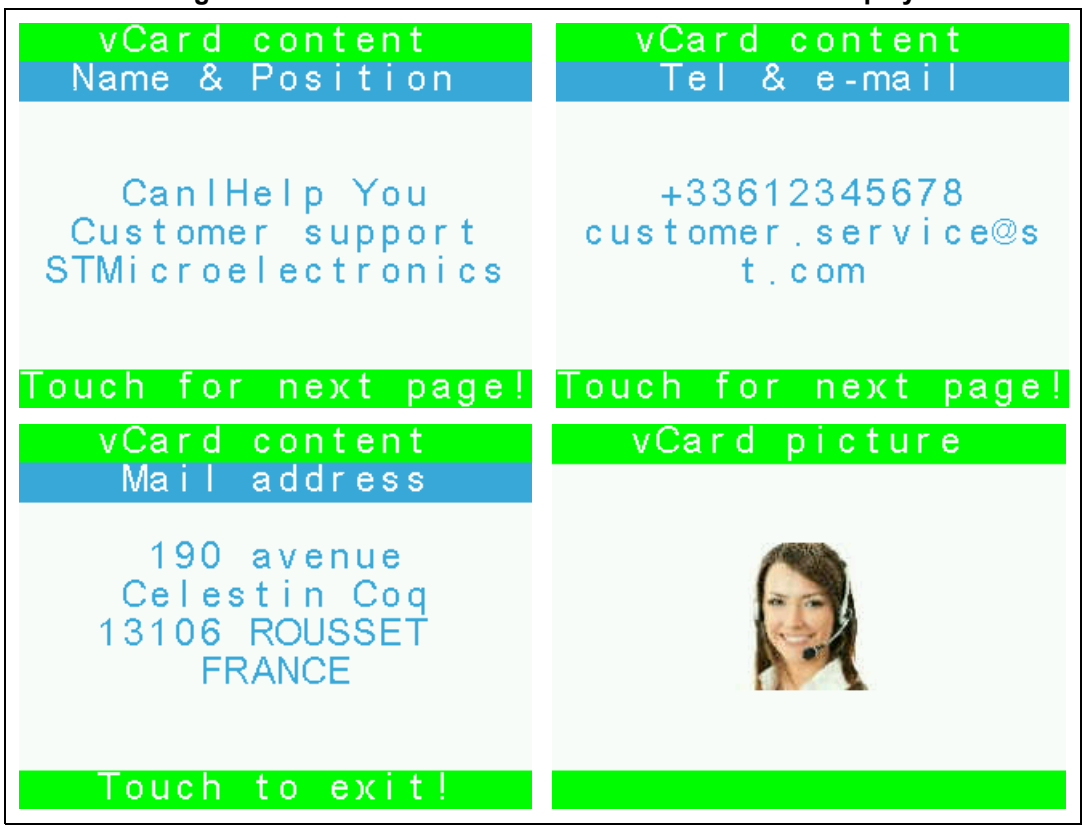
If the vCard has been previously stored, the following information is displayed on the board screen: name, title, organization, cell phone number, work address and work e-mail.

Note: The vCard with an embedded picture needs around 3 kB of memory and cannot be stored in the smaller memory size ST25DV versions, the ST25DV64 (64 kbits) must be used.

Note: If, when the MCU reads the ST25DV memory, it does not contain a vCard, an error message is displayed.

Note: Only vCard 2.1 has been implemented in this firmware version. If user stores a vCard 3.0 in the ST25DV, the firmware issues an error message.

Figure 45. ST25DV-DISCOVERY read vCard content display



Smart phone and vCard NDEF

Refer to *Reading and writing NDEF on a smart phone with the ST25 application*.

Geolocation demonstration

This menu presents how to manage the geolocation content in the ST25DV.

With the MCU, the user can

- Store an NDEF message containing a geolocation (it can then be read with an NFC reader or with a smart phone)
- Read any geolocation that has been stored in the ST25DV memory, the content is displayed on the screen

Figure 46. ST25DV-DISCOVERY read Geolocation display

Note: If, when the MCU reads the ST25DV memory, it does not contain a Geolocation, an error message is displayed.

MyApp record demonstration

This demonstration shows how to use a proprietary NDEF record to control an application running on the host MCU. As the format of the message follows the NFC Forum NDEF standard, it can be written or read by any device supporting this format.

The MCU waits until a “Proprietary configuration” record is written to the tag, then it updates the LCD display and controls a LED as described in the record.

This demonstration uses the ST25DV GPO to detect that the ST25DV has been written. In the ST25 NFC App, the user shall enter the “compose NDEF” menu, and then select the “Proprietary configuration” record.

From there, the user can:

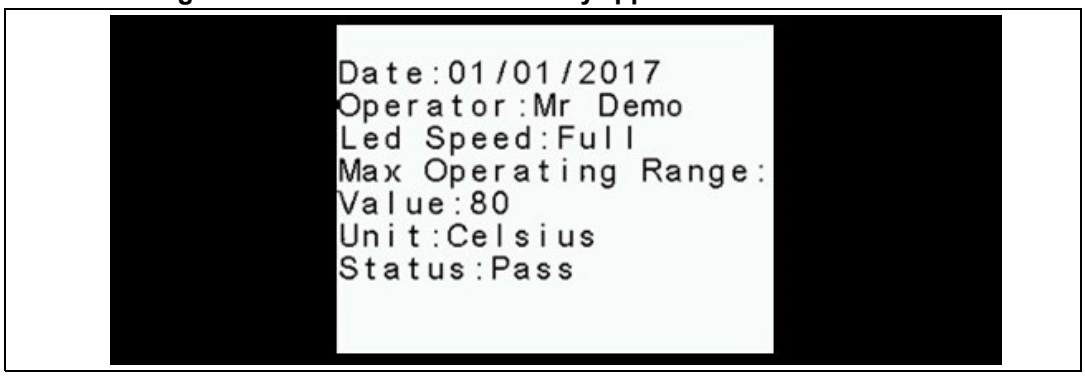
- Select if the LED1 must blink and configure the blinking period.
- Enter the text that is displayed by the MCU.

Note: The “Proprietary configuration” record allows the user to control up to four LEDs. As the ST25 Discovery board has only one LED, configuring LED2, LED3 or LED4 won’t have any effect.

After changing the setting in the Compose “Proprietary configuration” record menu, the user must click on “Write to tag” and tap the ST25DV. As soon as a new message is written in the ST25DV memory, the MCU reads it and if it finds the expected message, it configures the LED and the LCD as specified in the message.

If the data written to the ST25DV is not a valid “Proprietary configuration” record, the MCU displays an error message and continues waiting for a valid “Proprietary configuration” record to be written.

Figure 47. ST25DV-DISCOVERY MyApp demonstration screen



Multi record demonstration

This menu presents how to manage a multi record NDEF in the ST25DV.

With the MCU, the user can

- Store an NDEF message containing the ST25 URI (it can then be read with an NFC reader or a smart phone)
- Add the ST25 Android™ application record to the existing NDEF message.

For this demonstration the user is requested to read the URI with a smart phone, and check that its preferred browser is open with the ST25 URL (this is the default behavior when URI are natively supported by the smart phone).

Then, after adding the ST25 AAR, when the ST25DV is read with a smart phone, the browser is not open anymore, while the ST25 application is. The URI appears as the first record in the application.

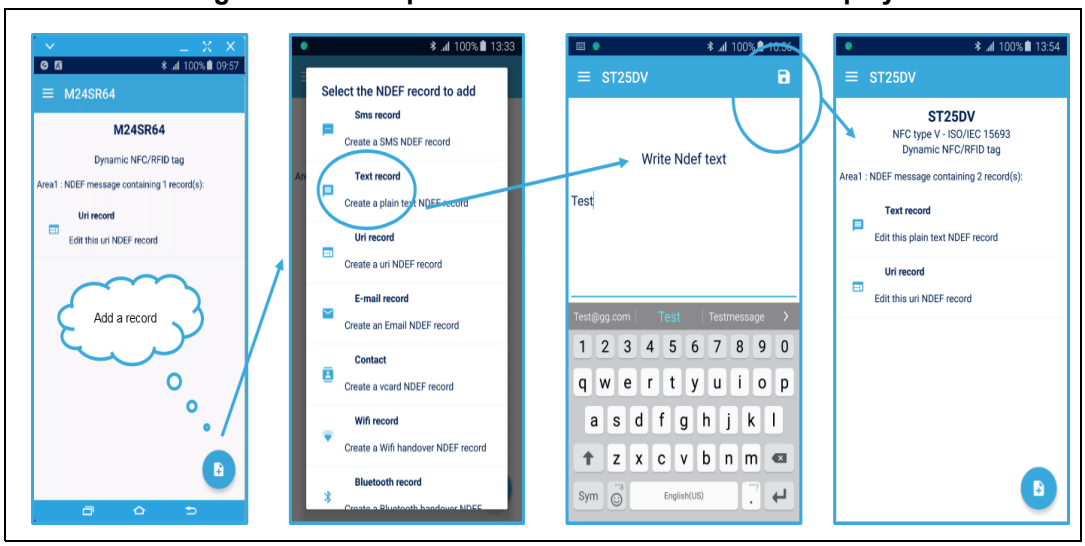
Smart phone and multi records NDEF

This section provide dedicated indications to build multi records NDEF (see *Figure 48*).

How to build each record:

- From “Ndef Editor” menu, click on the ‘+’ at the bottom right corner to add a record. A popup appears to allow you to choose the record type that you want.
- Fill the record and click on the Floppy icon, in the top right corner, to save the record to the tag.
- Proceed in the same way to add some more records.

Figure 48. Smart phone and Multi records editor display



Bluetooth® Low Energy pairing demonstration (requires an additional BLE module)

This menu indicates how to manage a Bluetooth® Low Energy (BLE) pairing content in the ST25DV. By default, the ST25-DISCOVERY board doesn't feature the BLE module, but the ST BlueNRG module can be added on it.

For this demonstration it is required to read the ST25DV content with a smart phone having the Bluetooth® Low Energy OOB support (e.g. Android™ 6.0).

With the MCU, the user can

- Store an NDEF message containing the BLE out-of-band (OOB)
- Read a Bluetooth® or BLE OOB previously stored in the ST25DV memory

When the user reads the ST25DV with a smart phone having the BLE support, the smart phone automatically asks confirmation and connects to the BlueNRG module presents on the demonstration board. The MAC address provided by the smart phone is displayed on the screen of the demonstration board as soon as the pairing is completed.

The BlueNRG is configured to act as a Human Interface Device (HID); it is displayed on the smart phone with this name: HID.

The HID protocol is natively supported by Android™ 6.0 smart phones, and this demonstration allows the user to control a pointer on the paired phone, by touching the screen of the ST25DV-DISCOVERY board. The screen acts as a touch-pad. The user button on the ST25DV-DISCOVERY board acts as a select button.

Touch the bottom line of the LCD to exit this demonstration.

The user can select the “Change BLE Address” command to prevent spurious connections from previously paired smart phones.

This changes the BLE module device address, with the effect that it is seen by smart phones as a different device (however the device name - HID - is left unchanged).

Note: Some smart phones require a specific action from the user to enable the HID control (this setting can be located in the Bluetooth® menu of the smart phone).

Smart phone and Bluetooth low energy

Please refer to *Reading and writing NDEF on a smart phone with the ST25 application*.

Wi-Fi® pairing demonstration (requires an additional Wi-Fi® module)

This menu presents how to manage a Wi-Fi® pairing content in the ST25DV. By default, the ST25DV-DISCOVERY board doesn't feature the Wi-Fi® module, but the ST SPWF01 module can be added on it, acting as a mini-Wi-Fi® access point.

For this demonstration, it is required to read the ST25DV content with a smart phone with Wi-Fi® support.

With the MCU, the user can

- Store an NDEF message containing a Wi-Fi® Protected Setup (WPS) out-of-band descriptor.
- Read a WPS OOB previously stored in the ST25DV memory.

When the user reads the ST25DV with a smart phone having the Wi-Fi® support, the smart phone automatically asks a confirmation and connects to the SWPF_AS01 Wi-Fi® network. The MAC address provided by the smart phone is displayed on the screen of the demonstration board as soon as the connection is completed.

Note: For the demonstration no Internet access is provided by the board mini-Access Point.

Note: If, when the MCU reads the ST25DV memory, it does not contain a WPS OOB, an error message is displayed.

Smart phone and Wi-Fi pairing

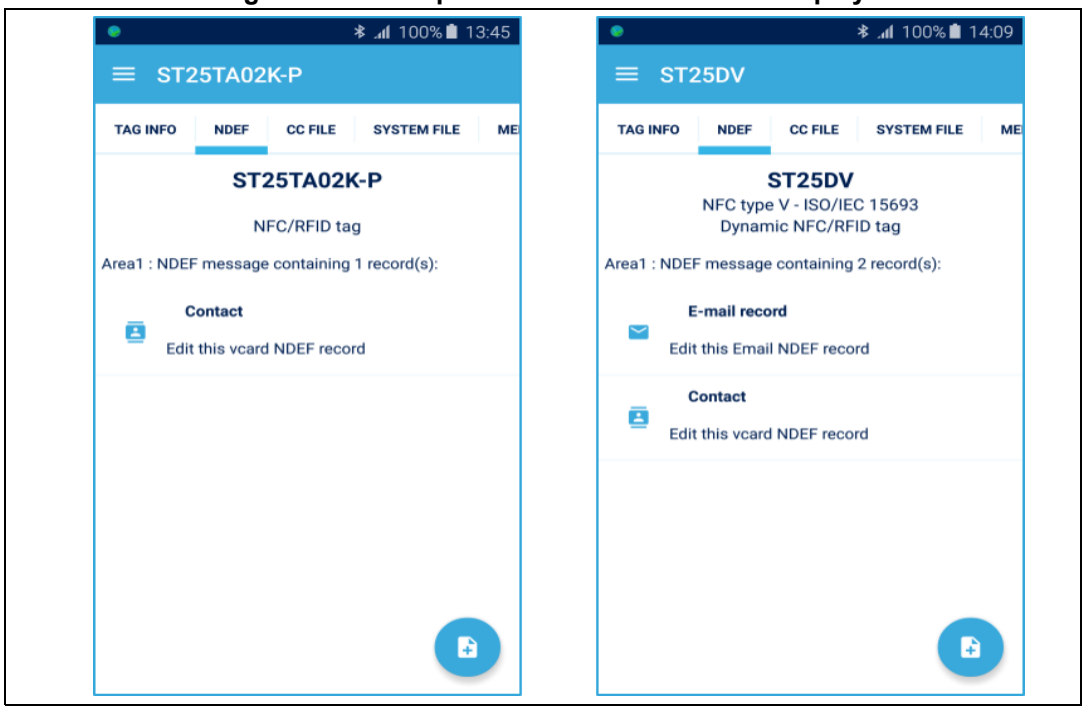
Please refer to *Reading and writing NDEF on a smart phone with the ST25 application*.

Reading and writing NDEF on a smart phone with the ST25 application

The ST25 NFC application enables to read or write NDEF messages.

Read: Tap the tag, if a managed NDEF is discovered, content is accessible from the NDEF tab or NDEF Editor (refer to Android™ application user manual for more details):

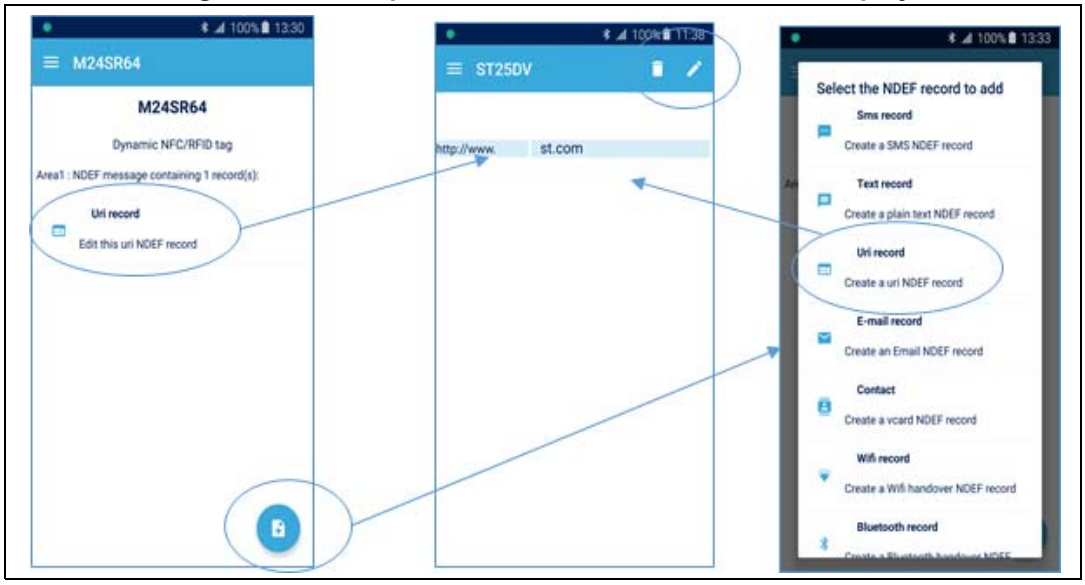
Figure 49. Smart phone and NDEF discover display



Write an NDEF: Select “NDEF editor” and choose the NDEF type you would like to write.

An editor according to the selected type is displayed.

Figure 50. Smart phone and URL NDEF write menu display



Clear EEPROM

This menu allows the user to clear the EEPROM content. The three choices are:

1. Write an empty NDEF
2. Erase the CCFile (replace the first four bytes of memory with 0xFF)
3. Clear EEPROM (fill the whole memory with 0xFF).

Appendix A FTM protocol details

A.1 Introduction

The new FTM protocol has been introduced to bring the following benefits:

- Protocol integrated into the ST25SDK so that every application (Android™, iOS®, PC) can use it (previous protocol was fully managed by the applications).
- Simplify the use of the Fast Transfer Mode: A single API function takes care of the transmission (whatever the data size) so users do not have to bother anymore to split the data and to acknowledge them.
- Short overhead in order to optimize the data rate.
- Robust in case of errors or if the tag is moved out and back in the RF field. The transmission should be able to restart and to complete without error.

The NFC Reader (ex: a smart phone) always is the initiator of all communications.

The new protocol is available on MB1286 “ST25DV-Discovery board” and on MB1396 “ST25DV64KC-DISCO board”.

A.2 Operating modes

The protocol is able to operate in two modes:

- A mode with error recovery: This mode is robust in case of transmission error or packet lost.
- A mode without error recovery. This mode gives the priority to the transfer speed but is not able to recover in case of error.

For both modes, the data to transfer should be split to fit in the ST25DV-I2C’s Mailbox (which can contain up to 256 bytes). «Packet» is called the amount of data fitting into the ST25DV’s mailbox.

Depending of the size of the data to send, there are one or more «packets» to transmit.

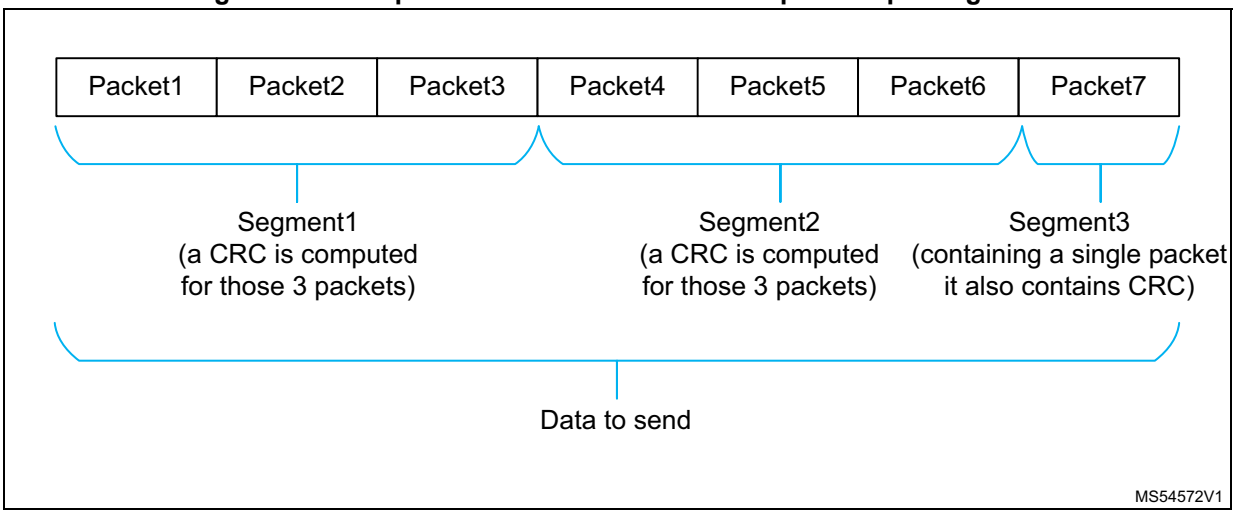
When sending big data files (like images or firmware data), it is interesting to detect transmission errors early and to be able to resent only the failing packets. The user could use a CRC for the whole file but in case of error user have to resend all the file. User could use a CRC to acknowledge each packet but this would have a big impact on performance. Instead a CRC can be used to acknowledge a group of packets. This group of packets is called “Segment”.

A CRC is computed on all the data of the segment. If the CRC is correct, all the packets of the segment are considered as valid. Otherwise, an error is sent to the transmitter and all the packets of the current segment have to be resent.

Depending of the use case, the application has the possibility to change the number of packets per segment. In the ST25SDK, the default setting is 20 packets per segments (= 4900 Bytes per segment).

Example of transmission with three packets per segment:

Figure 51. Example of transmission with three packets per segment



A.3 Low-level protocol

A low-level protocol has been defined to:

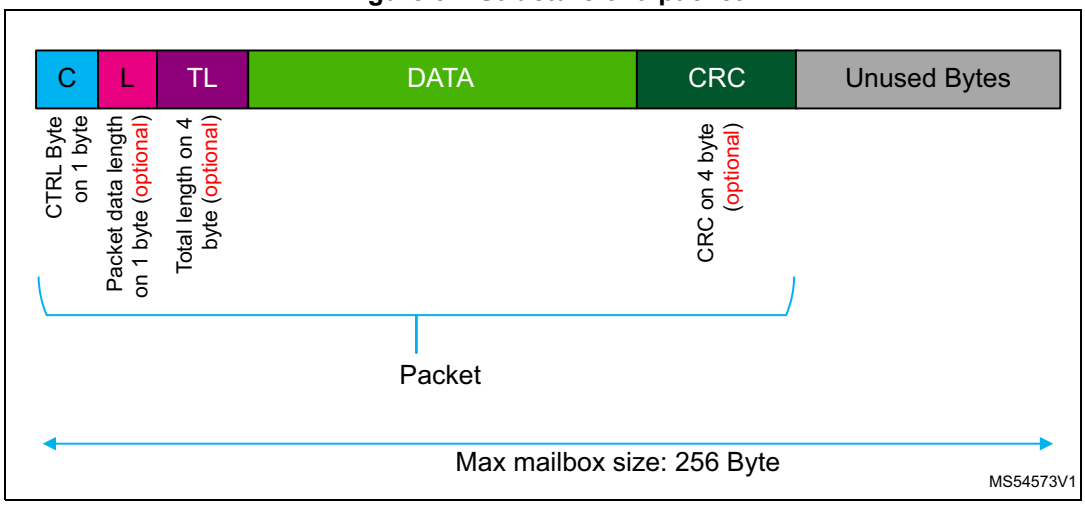
- Split the data to send into packets (a small header is added to indicate the content of the packet)
- Ensure their transmission
- Possibly: Check the validity of the data received and acknowledge them

This protocol allows user to send data of any size, both ways.

Each packet starts with a “Control Byte” or a “Status Byte”:

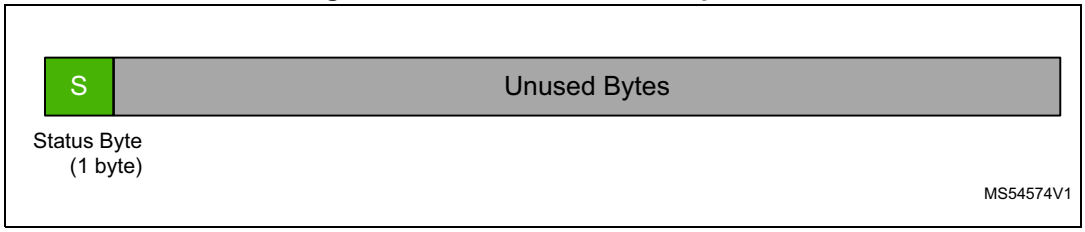
1. A “Control Byte” is used to indicate the bytes present in the packet (packet data length, total, data length, CRC presence…etc).

Figure 52. Structure of a packet



2. A “Status Byte” is a single Byte sent in response to one or more received packets to indicate if the transmission was successful or not.

Figure 53. Structure of a status packet



The **control byte** has the following content:

- b7: This bit should be set to 0 for a control byte (it is used to distinguish control bytes and status bytes).
- b6: Bit indicating if a packet data length field (on one byte) is present. It indicates the length of the data present in this packet (including the CRC bytes, if any). If no packet data length field is present, the receiver considers that the packet has the max possible size.
- b5: Segment End. When this bit is set, a CRC is present in this packet. The CRC is calculated on all the data present between “Segment Start” and “Segment End”. An acknowledge should be sent by the receiver.
- b4: Segment Start.
- b3-b2: Bits used to indicate data split on several packets. They have the following meaning:
  - b00: Data fitting in a single packet.
  - b01: First packet of a serie of packets. In that case, a total length field (on 4 bytes) is present. It indicates the length of the data that has been split on several packets.
  - b10: Middle packet (there can be more than one).
  - b11: End packet finishing the transfer of the data.
- b1: Parity bit toggling for every new segment. It allows to detect the re-transmission of the same segment.
- b0: Bit indicating if the packet belongs to a segment.

The **status byte** has the following content:

- b7: This bit should be set to 1 for a status byte (to avoid confusion with control bytes).
- b6-b4: RFU.
- b3-b0: Status of the last packet(s) received:
  - 00: SEGMENT_OK
  - 01: SEGMENT_ERROR (Ex: CRC error, packet lost…etc)

So a status byte with the value 0x80 corresponds to SEGMENT_OK and 0x81 corresponds to SEGMENT_ERROR.

A.4 ST Light Protocol

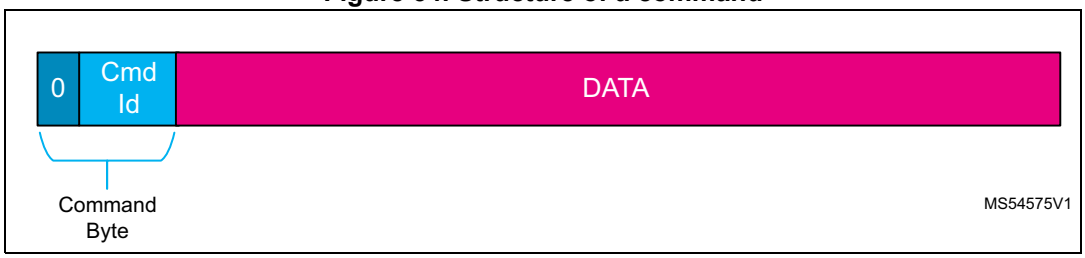
On top of previous low level protocol, a light protocol is used to indicate the content of the data sent. This light protocol is used to send commands that can be understood by the receiver.

It is not mandatory to use this light protocol. Users can use their own protocol if they prefer.

In this protocol, the data sent are called “Commands”.

A command starts with a byte called “cmdId” and can be followed by some data:

Figure 54. Structure of a command

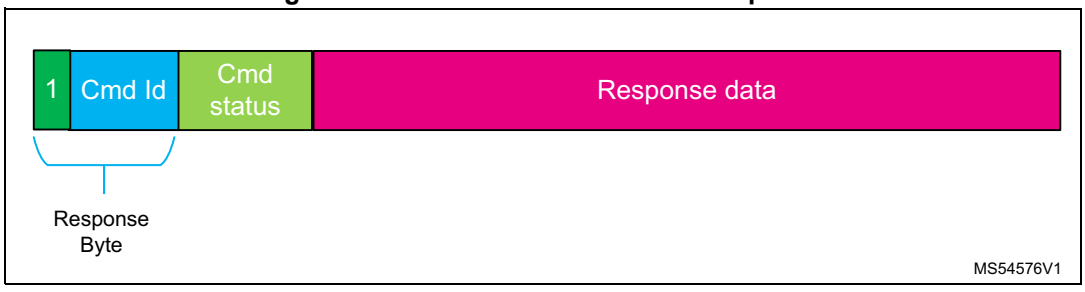


Warning: **Warning: The most significant bit of the command_id should be set to 0.**

Response format:

- The response should start with a response byte that is (0x80 | cmdId). It allows user to check that the response really matches the command sent. The most significant bit is set to 1 to indicate that it is a response and not a command.
- Then there is a command status:
  - CMD_OK (=0x81) in case of success
  - CMD_ERROR (=0x82) in case of error
  - CMD_INTERNAL_ERROR (=0x83) in case of internal error (not due to the protocol).
  - CMD_UNKNOWN (= 0x84)
  - CMD_NOT_ALLOWED (= 0x85): Command not allowed
- Then there might be some “Response data”

Figure 55. Structure of a command response



Note: Beware to confusion between "status bytes" used for acknowledging (SEGMENT_OK = 0x80, SEGMENT_ERROR = 0x81) and Command Status (CMD_OK = 0x81, CMD_ERROR = 0x82).

The following commands are currently used:

Table 1. FTM commands list

Cmd Id	Command name	Data	Response data
0x00	FTM_GET_BOARD_INFO	Data should contain 2 bytes (0x00, 0x00)	The response data contains 4 bytes: one indicating the board name and 3 indicating the firmware version
0x01	FTM_SEND_PICTURE	Data should contain the picture	Null
0x02	FTM_READ_PICTURE	Null	The response data contains the picture read
0x03	FTM_STOPWATCH	Data should contain the stopwatch data	Null
0x04	FTM_FW_UPGRADE	Data should contain the firmware to send	Null
0x05	FTM_SEND_DATA	Data should contain the data to send	Null
0x06	FTM_READ_DATA	Null	The response data contains the data read
0x07	FTM_SEND_PASSWORD	Data should contain the password to send to grant the Firmware upgrade	Null
0x0F	ECHO (used only for debug)	The first byte indicates if the STM32 should send a response with or without error recovery (0x00 = without error recovery, 0x01 = with error recovery). The other bytes can be chosen by the issuer	The response data contains the same bytes has in the command data

4 Revision history

Table 2. Document revision history

Date	Revision	Changes
20-Feb-2017	1	Initial release.
30-May-2017	2	Updated: – Replaced in all document STM32F415 product by STM32F405. – *Figure 2: ST25DX-DISCOVERY motherboard MB1283-B*, *Figure 7: Smart phone GPO register configuration*, *Figure 10: Smart phone and area display*, *Figure 11: ST25DV-DISCOVERY Fast Transfer Mode display*, *Figure 18: ST25DV drawer menu (Mailbox management, FTM demonstration, Stopwatch demonstration)*, *Figure 20: Data transfers from reader to tag of a random buffer*, *Figure 26: Smart phone and firmware upgrade display*, *Figure 28: Transfer from smart phone to tag*, *Figure 37: Smart phone and data transfers H2R with selected size*, *Figure 37: Smart phone and data transfers H2R with selected size*, *Figure 40: Smart phone and stopwatch transfers*, *Figure 48: Smart phone and Multi records editor display*, *Figure 49: Smart phone and NDEF discover display*, *Figure 50: Smart phone and URL NDEF write menu display* – *Section 3.1: Prerequisite*, *Smart phone GPO management*, *Smart phone and multi areas*, *Section 3.2.3: “Fast Transfer Mode Demo” menu*, *Smart phone and Fast Transfer Mode use cases*, *Smart phone and FTM data transfers*, *Smart phone and firmware upgrade*, *Smart phone and picture transfers*, *Smart phone and stopwatch transfers*, *Smart phone and multi records NDEF* Added: *Figure 19: Mailbox Status* Removed *Figure 8: Smart phone GPO interrupt*, *Figure 58: Smart phone and Multi records discovered display*
11-Oct-2021	3	Update: – *Section 3.2.3: “Fast Transfer Mode Demo” menu* – *Figure 12: Smart phone and Fast Transfer Mode use cases display*, *Figure 26: Smart phone and firmware upgrade display*, *Figure 28: Transfer from smart phone to tag*, *Figure 34: Transfer from tag to smart phone*, *Figure 37: Smart phone and data transfers H2R with selected size*, *Figure 40: Smart phone and stopwatch transfers* Replaced block “protocol definition” in *Section 3.2.3: “Fast Transfer Mode Demo” menu* with *Appendix A: FTM protocol details*

IMPORTANT NOTICE – PLEASE 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 acknowledgement.

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, please 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.