User manual

ST25DV-I2C out-of-band pairing demonstration

Introduction

This document shows how to run the ST25DV-I2C out-of-band pairing demonstration, using NFC to improve the security of Bluetooth® Low Energy (BLE) connections.

The ST25DV-I2C is a dynamic NFC Tag IC able to communicate with NFC readers and smartphones, and also with a microcontroller through an I^2C interface.

One of the most vulnerable phase is the one where the two Bluetooth® devices exchange keys and agree on the encryption key to use. This information is exchanged over the air so it is vulnerable to man-in-the-middle (MITM) attacks.

NFC provides a side communication channel with the advantage of being very short distance (a few centimetres) so the hackers cannot intercept this communication. The information, exchanged through NFC, is used to authenticate the cryptographic keys exchanged by the two Bluetooth® devices. Hence, the data transfer is safe.

The following packages are available on www.st.com for this demonstration:

- STSW-ST25DV004 firmware
- STSW-ST25005 Android™ application

1 General information

The application described in this document runs on to STM32WB55 Arm®-based devices.

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

1.1 Purpose and scope

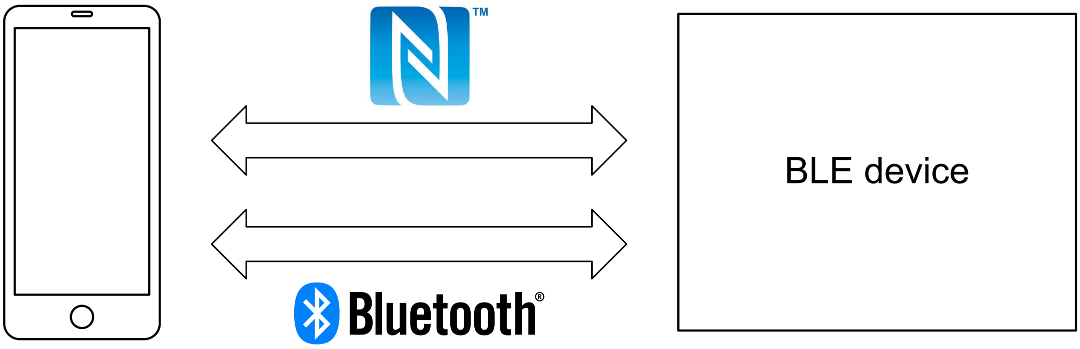
The ST25DV-I2C out-of-band pairing demonstration runs on the MB1355C board plus a X-NUCLEO-NFC04A1 shield, featuring a ST25DV-I2C tag connected to an STM32WB55 device through the I^2C bus.

The pairing with a Bluetooth® device is greatly facilitated by the use of NFC. Instead of selecting a Bluetooth® device in a list, the user can tap the NFC Tag. The Bluetooth® connection is automatically set up, avoiding the risk of selecting the wrong device in the list. This is the Bluetooth® handover over NFC.

NFC can be seen as a side communication channel between the Bluetooth® device and the Android™ phone. This extra communication channel is used to exchange some data, called out-of-band data, used during the Bluetooth® pairing.

These data contain a random number and a commitment value used to check the validity of the "Public Key" received during the "Public Key exchange" phase, preventing a MITM attack.

Figure 1. Out-of-band BLE pairing



1.2 Glossary and acronyms

Table 1. Glossary and acronyms

Glossary/Acronyms	Meaning
NFC	Near field communication
BLE	Bluetooth® Low Energy
ECDH	Elliptic curve Diffie-Hellman
MITM attack	Man-in-the-middle attack
NDEF	NFC Data Exhange format
OOB pairing	Out-of-band pairing

1.3 Hardware equipment

The following hardware is needed for this demonstration:

- MB1355C board plus X-NUCLEO-NFC04A1 shield
- An Android™ smartphone with at least the version 7.0 (Android™ Nougat)

Figure 2. **MB1355C board**

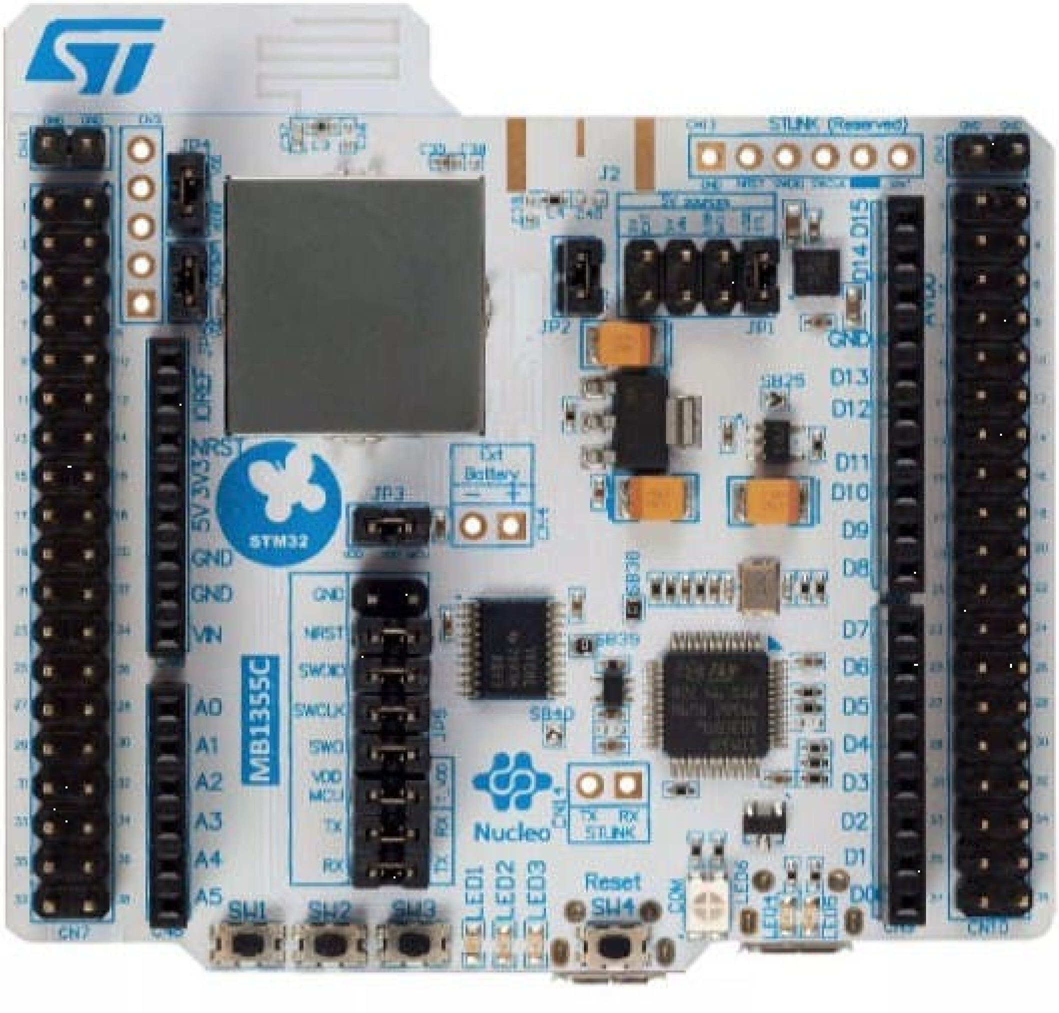
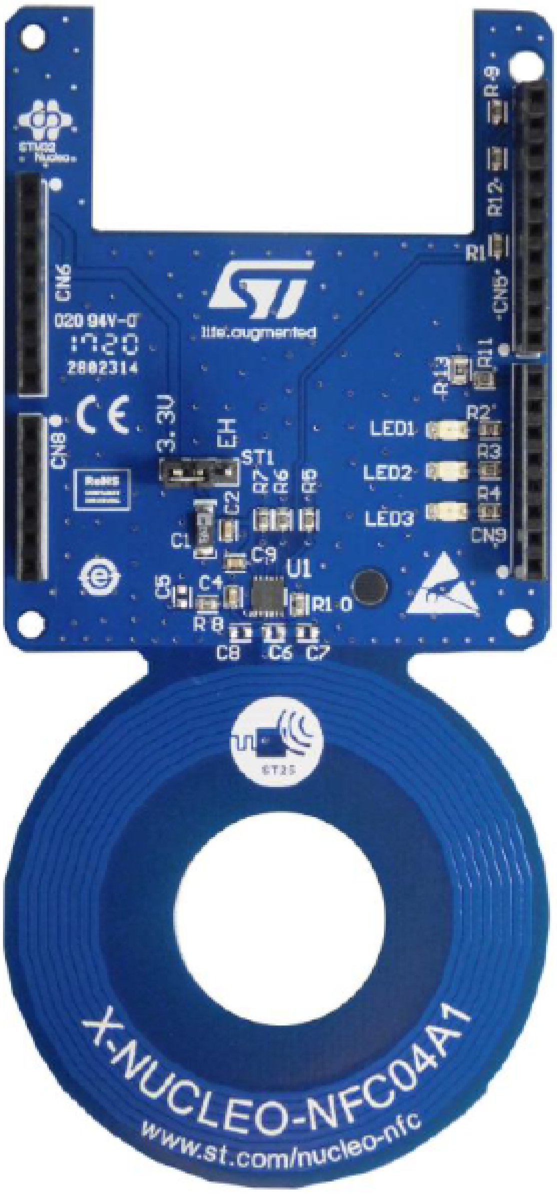


Figure 3. **X-NUCLEO-NFC04A1 board**



1.4 Installation

This package includes the firmware for the MB1355C board and the Android™ executable (APK) to be used on Android™ smartphones.

1.4.1 MB1355C board installation

To program the MB1355C board, perform following steps:

1. Install the ST-LINK USB driver, available on www.st.com
2. Connect the MB1355C board to a PC with the ST-Link USB-micro
3. The MB1355C board icon appears in the computer directory
4. Drag-and-drop the firmware to the MB1355C board icon
5. Restart the MB1355C board

1.4.2 Android™ APK application installation

This application is not available on Google Play™ store so it must be installed manually. By default, Android™ prevents the installation of programs not coming from Google Play™ so go through the following steps:

1. Navigate to setting → Security.
2. Check the option "Unknown sources".
3. Tap “OK” on the prompt message.
4. Select "Trust".
5. Connect the Android™ phone to a PC with an USB cable and copy the APK to the phone’s internal memory.
6. With a file explorer, browse the memory of the phone and go to the folder where the APK is copied. Click on the file to install it.

Note: Depending upon the Android™ version and the phone model, the wording may vary.

Note: If there is no file explorer, the program "File commander" is available for free on Google Play™.

1.5 License scheme

The Android™ application and the associated firmware are provided under the SLA0052 software license agreement, available on www.st.com.

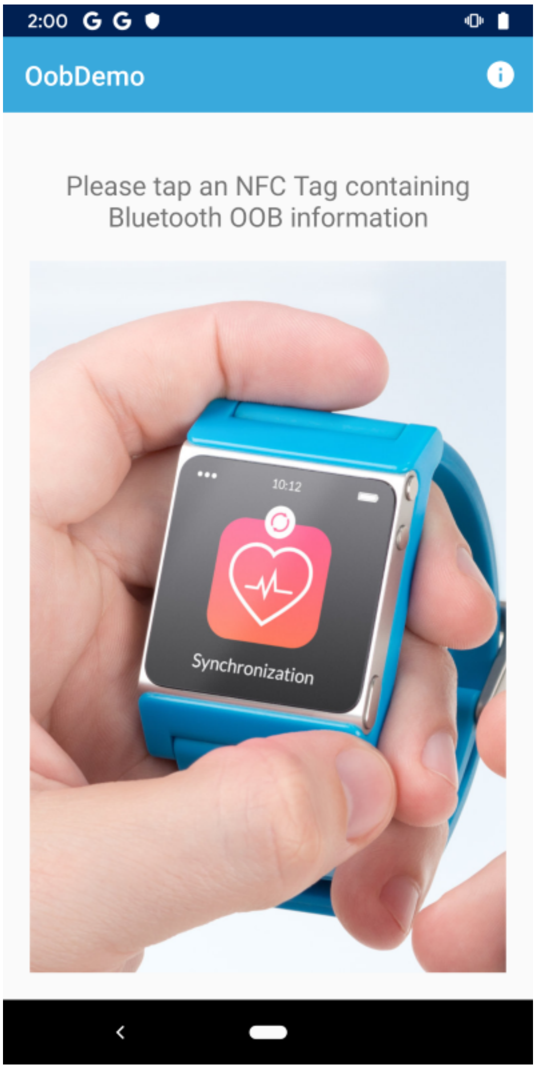
The software components provided in this package come with different license schemes, as shown in Table 2.

Table 2. License scheme

Component	Copyright	License
Application projects	STMicroelectronics	SLA0052
LibNDEF		
Board support package (BSP)		BSD 3-Clause
STM32WB HAL/LL APIs		
STM32WB Bluetooth® HCI		SLA0044
STM32WB Bluetooth® stack		
STM32WB Bluetooth® profiles and services		
STM32WB Bluetooth®/Thread concurrent stack		
Cortex®-M CMSIS v4.5.0	Arm®	BSD 3-Clause

2 ST25DV-I2C "OobDemo" application screens

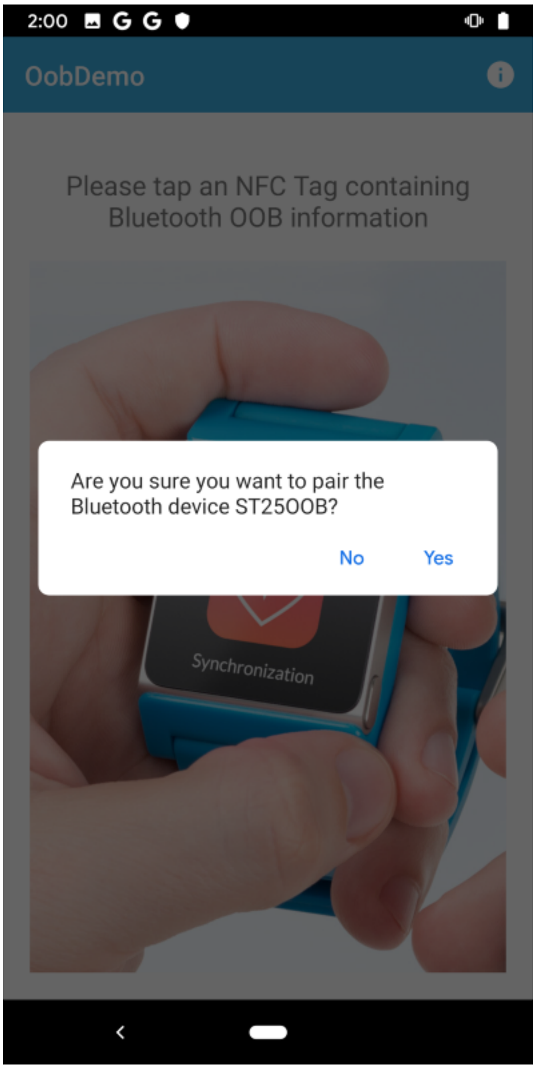
Figure 4. Home screen



Please tap an NFC Tag containing Bluetooth OOB information

Launch the application “ST25OobDemo” and tap the NFC Tag of the X-NUCLEO-NFC04A1 board.

Figure 5. Give confirmation

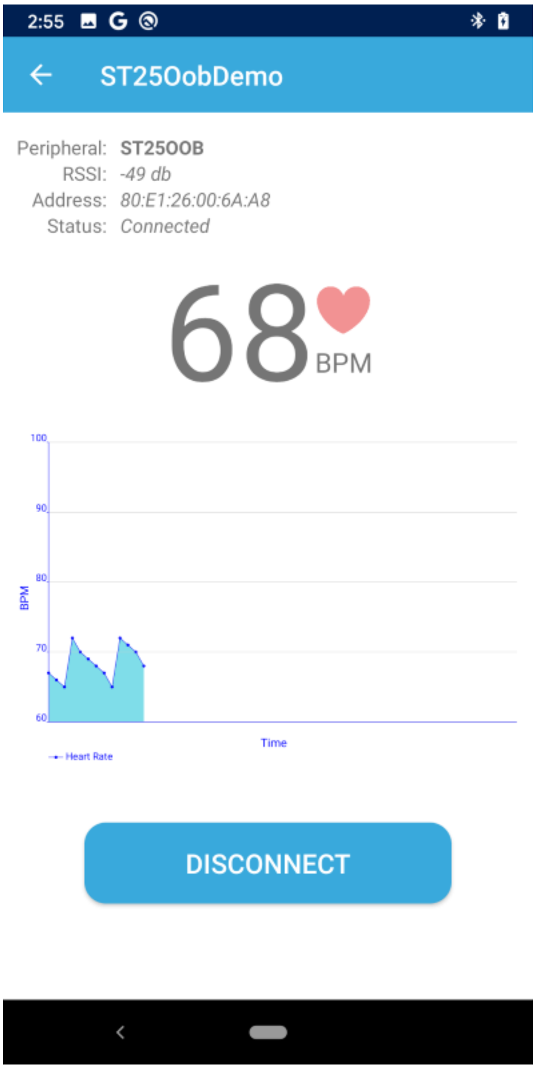


Please tap an NFC Tag containing Bluetooth OOB information

Are you sure you want to pair the Bluetooth device ST25OOB?

A pop-up requests confirmation to do a Bluetooth® pairing with this device.

Figure 6. Setup launching



2:55

ST25OobDemo

Peripheral: ST25OOB
RSSI: -49 db
Address: 80:E1:26:00:6A:A8
Status: Connected

68 BPM

If user clicks ‘Yes’, the BLE connection is installed. The smartphone then starts receiving some fictive heart rate data from the Bluetooth® device and starts displaying them as a graph.

Click on the "DISCONNECT" button to stop the connection with this device (otherwise, the device remains connected).

3 Technical information

The NFC Tag contains an NDEF record with Bluetooth® handover information and OOB data:

- Record type name : “application/vnd.bluetooth.le.oob"
- Bluetooth® device name: e. g. « ST25OOB »
- Bluetooth® device address: e. g. 80:e1:26:00:6A:a8
- LE secure connections "Random Value" (16 bytes)
- LE Secure connections "Confirmation Value" (16 bytes)

The last two fields are optional and correspond to the OOB data.

When taping the NFC Tag, this NDEF message is read natively by the NFC service of the Android™ phone. A pop-up is displayed to ask the user to confirm if it really wants to do a pairing with this Bluetooth® device. The Bluetooth® pairing starts if the user clicks "Yes".

If the OOB data are present, it is used during the pairing process.

The Android™ application uses a "Broadcast Receiver" to be notified when a Bluetooth® connection is done. This "Broadcast Receiver" checks if the connected Bluetooth® device has the expected device name. If this is the case, the application starts displaying the data received.

Note: In this demonstration, the user is not authenticated. In a real product, the device checks if the user has the permission to read the data. This is out of the scope of this demonstration, which shows how the NFC facilitates the pairing with a Bluetooth® device.

3.1 Secure simple pairing

The used pairing method is called “Secure simple pairing” (refer to §7 of [1]), performed in five steps:

1. **Public key exchange**: The devices exchange their public keys and compute a shared secret information thanks to Diffie-Hellman protocol.
2. **Authentication Stage 1**

   The OOB protocol is used during this phase. The smartphone uses the random value and the commitment value received from the Bluetooth® device. The protocol checks if the "Public Key" received during the "Key Exchange" phase is really the one of the device. If a hacker has substituted the "Public Key" during the phase 1, the verification of the commitment value fails. This is the protection against MITM attack.

Note: The NDEF data contained in the tag must be protected in write so that the data cannot be modified through the RF interface and cannot be manipulated.

3. **Authentication Stage 2**

   This stage confirms that both devices have successfully completed the exchange.
4. **"Link Key" calculation**

   Calculation of the "Link Key".
5. **LMP Authentication and Encryption**

   The final phase in simple pairing consists in authentication and generation of the encryption key.

4 Known issues

The current implementation has some issues, detailed in the table below. The following tickets have been opened on Android™ "Bug Tracker".

Table 3. **Issues description**

Ticket number	Short description
142410444	OOB pairing does not work if user puts only the “Heart Rate” profile. As a workaround, user must add also an “HID profile“ (handled natively by Android™)
143335473	After Bluetooth® handover through NFC, the NFC of the phone cannot be used for about 15 seconds.
143939066	If Bluetooth® is not enabled on the phone when user taps the NFC Tag containing Bluetooth® handover the NFC service automatically starts the Bluetooth® but it is running in a special mode called “*Quiet mode*”, which has a bug. As a workaround, the Bluetooth® is enabled by default when the application starts.

5 Reference documents

Table 4. Reference documents

Reference	Document title	Document url
[1]	*Bluetooth Core Specification*	http://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth_5.2_Feature_Overview.pdf
[2]	*Bluetooth Secure Simple Pairing Using NFC*	https://members.nfc-forum.org/apps/group_public/download.php/18688/NFCForum-AD-BTSSP_1_1.pdf

Revision history

Table 5. Document revision history

Date	Version	Changes
16-Apr-2020	1	Initial release.

Contents

List of tables

List of figures