Objective

This example demonstrates Out-of-Band (OOB) pairing between two PSoC 4 BLE devices using Near Field Communication (NFC).

Overview

This example demonstrates how to implement OOB pairing in PSoC 4 BLE device. OOB pairing achieves two things:

1. Automating the pairing process
2. Preventing Man-In-The-Middle (MITM) attack.

The central device acquires the OOB data such as the device address and the security key from an NFC tag and pairs with the peripheral device.

Requirements

Tool: PSoC Creator 3.1 SP1

Programming Language: C (GCC 4.8.4)

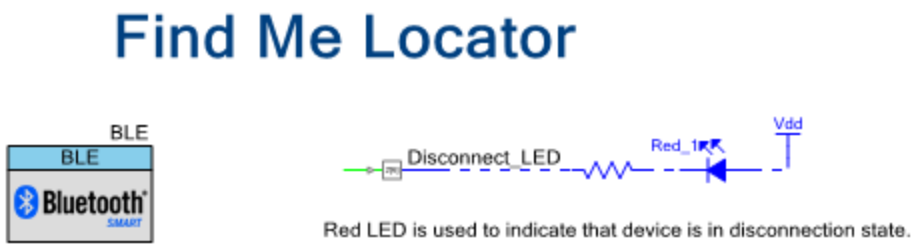
Associated Parts: All PSoC 4 BLE parts

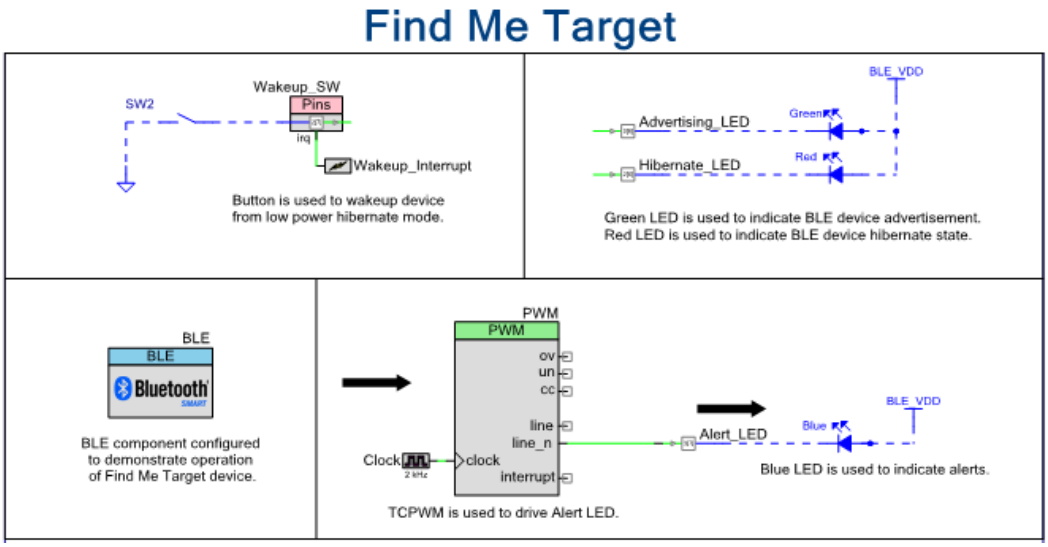
Required Hardware: CY8CKIT-042-BLE (2 numbers), Android phone with USB Host and NFC, NFC Tag, and USB On-The-Go (OTG) cable.

Some of the Android Smartphones that support both NFC and USB host are Nexus 4, Nexus 5, Samsung Galaxy S3, OnePlus One. Here is a list of phones that support NFC. An NFC tag is a small device with a chip that can store information and communicates with an NFC reader wirelessly within 4 cm (typical). These tags come in many forms such as key fobs, wrist bands, stickers etc. These tags are available at Amazon and eBay as cell phone accessories. An USB OTG cable allows a device to act as either USB host or as USB peripheral depending on the cable configuration. See this page to understand what makes a cable an OTG cable and to make one on your own.

PSoC Creator Schematic

This example is developed based on Find Me profile and has two projects; one for the Locator (GAP Central) and one for the Target (GAP Peripheral). The schematic for both the projects are given below. The UART component used in both the projects for debug purpose is not shown here.





Basics of Bluetooth Out-of-Band Pairing

Bluetooth spec has defined OOB pairing to ease the pairing process and to securely exchange the security key. One use case that eases the pairing process is just tapping the phone on an NFC-equipped Bluetooth speaker to connect quickly. This would otherwise require scanning for the Bluetooth devices in your phone, finding the speaker among the listed devices, and connecting to it. OOB pairing provides protection against passive eavesdropping and Man-In-The-Middle (MITM) attacks.

Refer to the following chapters in Bluetooth spec v4.1.

- Vol 1/Part A/Chapter 5.4 – LE Security
- Vol 3/Part H/Chapter 2.3 – Pairing Methods

Ideally, the OOB pairing should have been between just a phone and a BLE kit. But the pairing is done between two BLE kits in this demo because Android doesn't have API support for OOB pairing yet (API level 22). And this demo makes use of the phone as an NFC reader instead of using one in the form of an Arduino shield. See Figure 3 to understand the demo setup.

NFC is used as an OOB medium to exchange connection information. NFC forum has defined the data format to be used in OOB pairing and is available in a public document Bluetooth Secure Simple Pairing Using NFC. This document also defines all the terminologies involved in OOB pairing.

In a typical OOB pairing scenario, the central device (pairing initiator) sends a *Handover Request* message over NFC to get the OOB information such as target device's address, 128-bit security key etc. and the peripheral device (pairing responder) responds with a *Handover Select* message containing that information. This exchange is called as *Negotiated* Handover, shown in Figure 1. According to the Bluetooth spec for OOB pairing (Vol 3/Part H/Chapter 2.3.5 – Pairing Algorithms), the 128-bit key should be randomly generated at the time of handover, in which case both central and the peripheral should be equipped with NFC and the NFCs should be able to exchange data in peer-to-peer mode. This demo instead takes a simpler approach called as *Static Handover*, shown in Figure 2, in which the connection information including the security key is statically stored in an NFC tag. This demo intends to show how to use OOB APIs in PSoC 4 BLE. Therefore, an end-application should implement as per the guidelines from the Bluetooth spec.

The data stored in the tag follows the format in section 4.3.2 of the NFC Forum's application document Bluetooth Secure Simple Pairing Using NFC Rev 1.1. The data is stored in NFC Data Exchange Format (NDEF) and is a single NDEF record with MIME type *'application/vnd.bluetooth.le.oob'.*

Figure 1 – Negotiated Handover

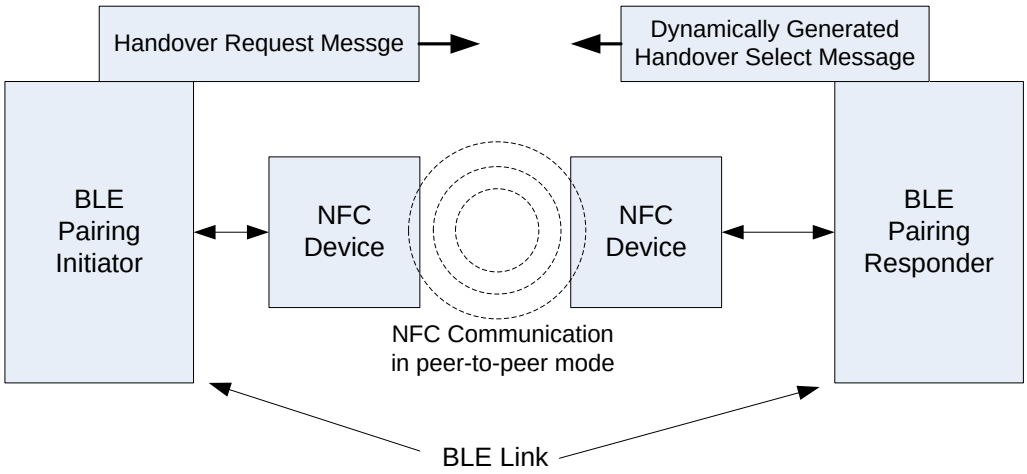
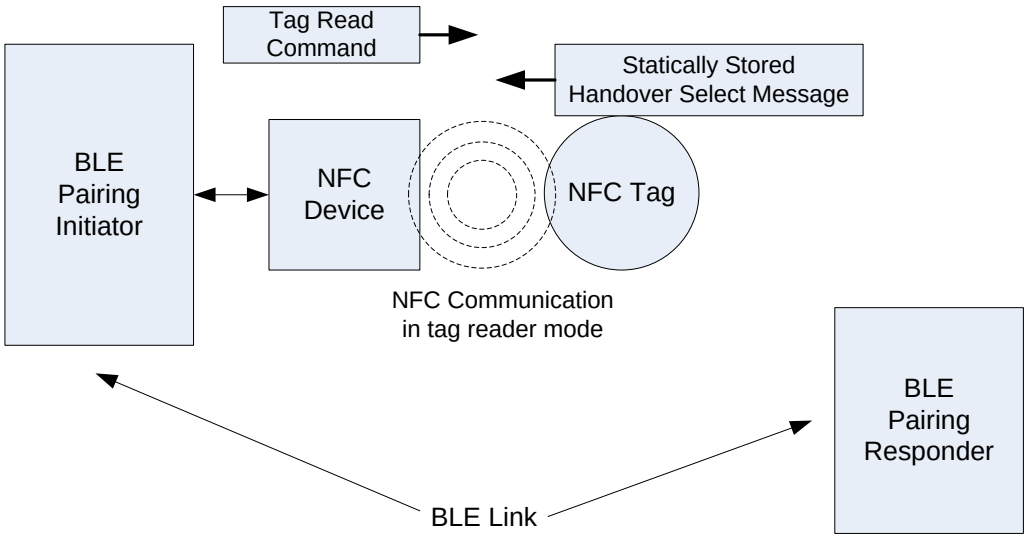


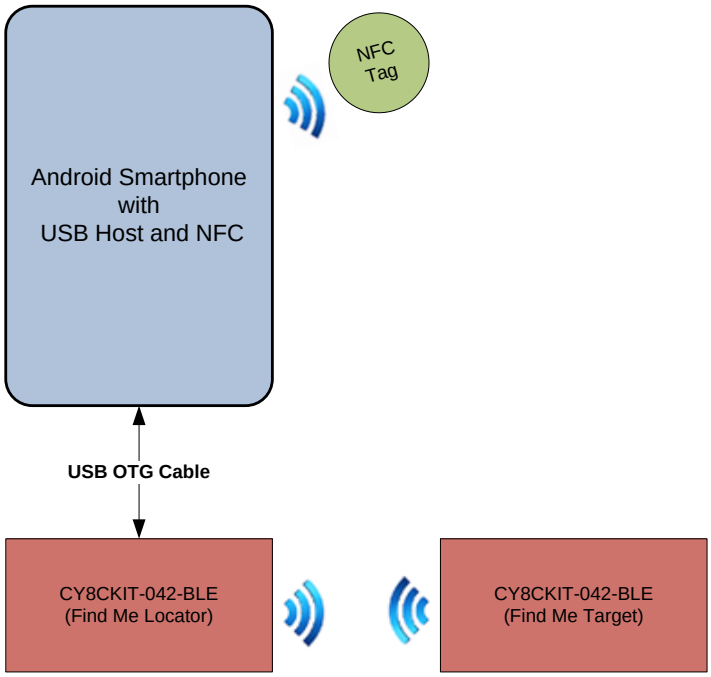
Figure 2 – Static Handover



Hardware Setup

Figure 3 depicts the hardware setup. This demo requires two CY8CKIT-042-BLE kits, an Android Smartphone that supports USB host and NFC, and an NFC tag loaded with the information about the target device. The Smartphone is connected to the FindMe Locator device using an USB On-the-Go (OTG) cable and is used as an NFC reader to get the information from the tag. The locator then uses the information received through the USB and pairs with the FindMe target device.

Figure 3 – Demo Setup



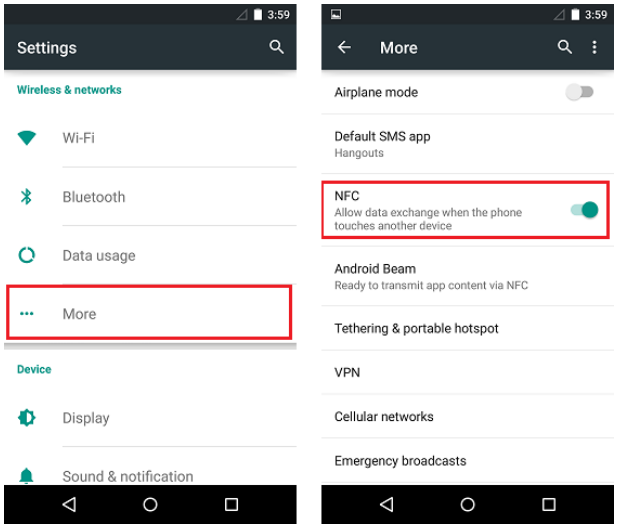
An Android app has been developed for this purpose which is provided along with the projects. Following are the instructions to work with the app.

- Plugging the USB automatically launches the app (for all the kits that has KitProg). The app may not be launched if the phone doesn't support USB host.
- A pop-up asks the user for permission to open the app every time the kit is connected unless the '*Use by default for this USB device*' is checked at the first time the USB is plugged.
- If the kit is already connected to the phone before opening the app, unplug and re-plug the cable once.
- If the phone doesn't have NFC, a pop-up asks the user whether to terminate the app. Note that this demo can't be experienced without NFC.
- A new tag can be configured with the target details using ***Configure Tag*** menu in the app.
- The app presents a console to display the status from the app and the messages received from the Locator. Use the ***Clear*** menu to clear the text in the console.

Operation

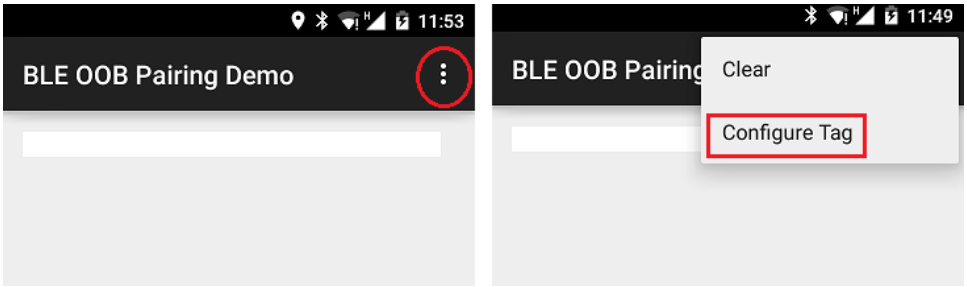
1. Install the ***BLE OOB Pairing Demo.apk*** app in an Android phone that has both USB host and NFC.
2. Configure a new NFC tag.
    a. Enable NFC on your Android phone in settings → More → NFC. Figure 4 Shows where to find this in Android version 5.0 (Lollipop).

Figure 4 – Enabling NFC in Settings



b. Open the app and choose “Configure Tag” from the Menu. The menu is located on top right corner in phones with latest Android versions as shown in Figure 5.

Figure 5 – Configure Tag Menu



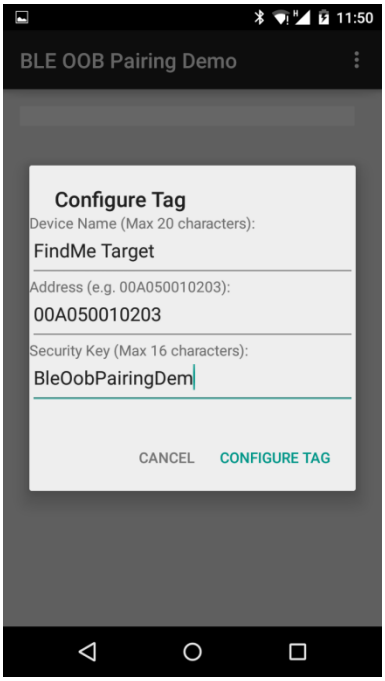
c. Place the phone over the NFC tag. Usually NFC is locates behind the phone.

d. In the app, enter the device name, Bluetooth device address (12 hex characters without any delimiter), and security key (16 characters) as shown in Figure 6 and click the “CONFIGURE TAG” button.

   Note that the security key entered here should be exactly the one set through the global variable *securityKey* in main.c of FindMe Target project.

e. If the tag configuration is successful, the message “APP: Tag Configured Successfully” appears on the app screen.

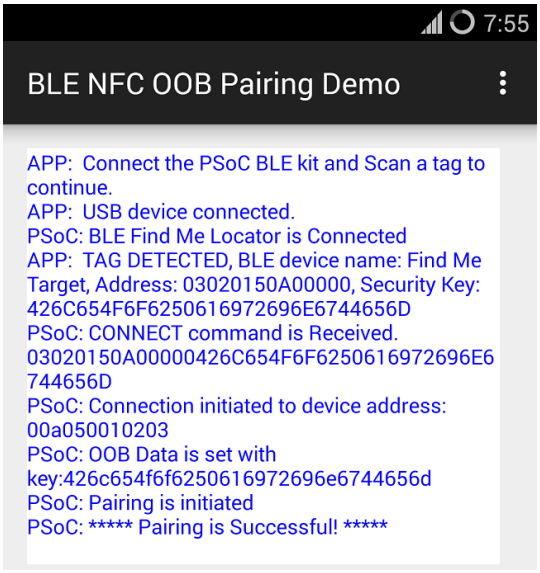
Figure 6 – Configure Tag Dialog



3. Load the BLE_FindMe_Locator.hex into one of the kits using PSoC Programmer.
4. Load the BLE_FindMe_Target.hex into the other kit.
5. Connect the FindMe Locator kit to the phone using an USB OTG cable.
6. The app should be automatically launched upon plugging the USB. If not, your phone may not support USB Host.
7. If a pop-up appears asking for permission to open the app, check the "*Use by default for this USB device"* box and click OK.
8. A pop-up asks whether to terminate if the phone doesn't have NFC. Ensure NFC is enabled in the settings. This demo can't be experienced without NFC.
9. Now, the console displays 'App: *USB device is connected'* and *'PSoC: BLE Find Me Locator is Connected'*. The messages shown by the app are preceded with *'App:'* and the messages received from PSoC are preceded with *'PSoC:'*.

   Note that second message is sent 3 seconds after the kit is powered through USB. It may not be displayed sometimes if the app takes more than 3 seconds to open. In that case, you can press the reset button on the kit.
10. Ensure the target device is ready to accept a connection. The RGB LED should be Green otherwise press SW2 to wake-up the device.
11. Now read the tag by placing the phone on the tag and the pairing is initiated from the locator device. '*Pairing is Successful!'* is displayed upon successful pairing. Figure 7 shows the entire message sequence on the console.
12. Note that the Target device also sends debug messages through the on-board USB-to-UART. The setting is 57600 baud rate, 8 data bits, 1 stop bit, and no parity bit.

Figure 7 – App Console



Demo

Figure 8 below shows a picture of the demo. The BLE kit closer to the phone is connected to the phone using USB OTG cable and the other kit is powered through USB from a PC USB port.

Figure 8 – BLE OOB Pairing Demo

Related Documents

Table 1 lists some of the relevant links. Visit PSoC 4 BLE web page to access all application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

Table 1. Related Documents

Document	Title	Comment
AN91267	Getting Started with PSoC 4 BLE	Provides an introduction to PSoC 4 BLE device that integrates a Bluetooth Low Energy radio system along with programmable analog and digital resources.
AN91445	Antenna Design Guide	Provides guidelines on how to design an antenna for BLE applications.
NFCForum-AD-BTSSP_1_1	Bluetooth Secure Simple Pairing Using NFC	Defines NDEF formats for Bluetooth pairing
Android NFC Guide	Android NFC developer guide	Provides overview of NFC technology and how to develop for NFC using Android APIs.