PSoC 4 BLE – Apple Notification Center Service

Objective

This example demonstrates the Apple Notification Center Service (ANCS) functionality using the PSoC 4 BLE family of devices.

Overview

Apple Notification Center Service (ANCS) enables a BLE capable device to connect to an iOS device and access notifications generated on the iOS device. This can be used to quickly design an accessory such as a smartwatch, which can pair with an iPhone.

This example showcases how PSoC 4 BLE can be used to support ANCS. The BLE Pioneer Kit is connected to an iPhone and a UART based console informs the user of the ongoing activity on the iPhone.

Incoming calls (with the caller name) are displayed on the terminal and can be accepted or declined by the user directly from the terminal. The number of missed calls, voicemails and emails is also shown.

For more information on ANCS itself, refer to the Apple Notification Center Service specification.

Requirements

Design Tool: PSoC Creator 3.1 SP1

Programming Language: C (GCC 4.8.4 – included with PSoC Creator)

Associated Devices: All PSoC 4 BLE devices

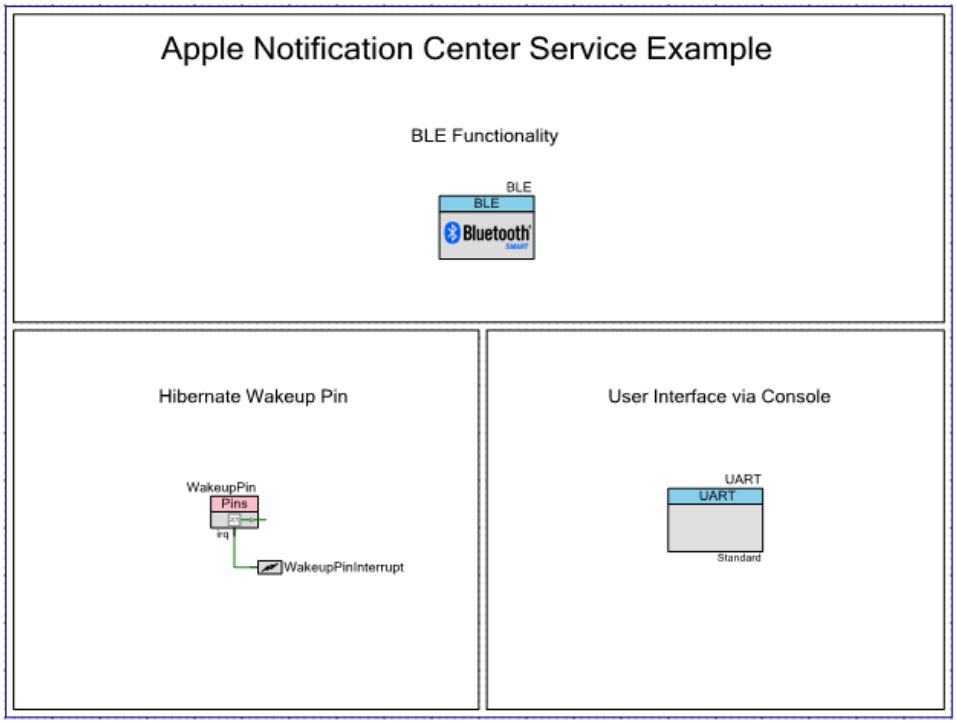
Required Hardware: CY8CKIT-042-BLE Bluetooth® Low Energy (BLE) Pioneer Kit

Hardware Setup

The BLE Pioneer Kit has all of the necessary hardware required for this lab. There is no special setup required.

PSoC Creator Schematic

Figure 1. PSoC Creator Schematic for ANCS



Project Description

This project implements a GATT Client and GAP Peripheral role. Being a Peripheral, the device advertises and waits for the iOS device to initiate a connection. Once connected, the device requests authentication, for which a passkey is displayed on the UART output, and the user is asked to enter this passkey on the iOS device.

Once the authentication is complete, the device initiates discovery of the ANCS service on the iOS device. After the discovery is done, the device subscribes to GATT notifications on the Notification Source and Data Source characteristics of the ANCS service.

When a new GATT notification arrives on the Notification Source characteristic, the device updates the relevant information on the UART output, which is seen on the terminal. There is a record maintained for the number of missed calls, voicemails and emails, as per the iOS device.

For an incoming call notification, the kit writes to the Control Point characteristic to request for the Attribute ID Title. This results in a GATT notification on the Data Source characteristic, from which the kit extracts the caller information. An alert is shown on the UART output, along with this caller information. The user is asked to Accept or Decline this incoming call. If the user enters ‘Y’ this call is accepted. If the user enters ‘N’ this call is declined.

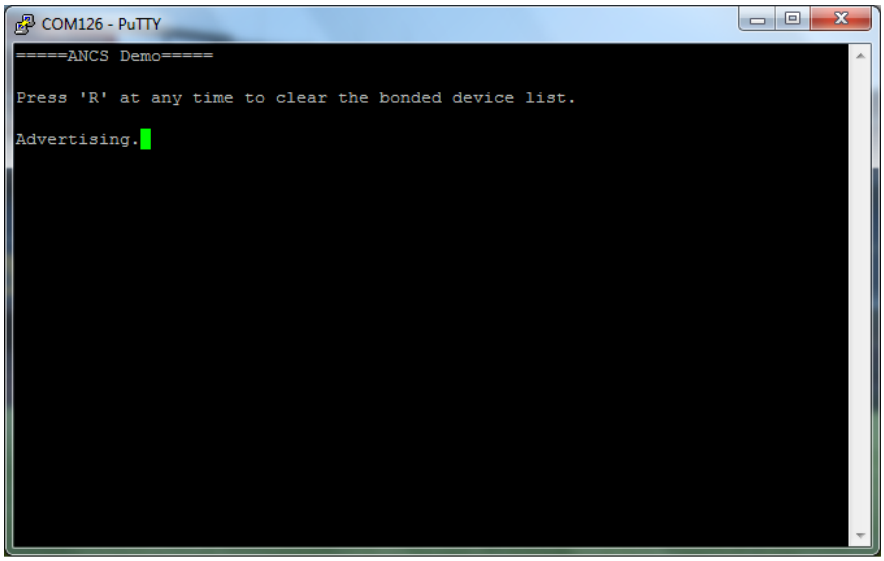
Bonding is implemented in this project, i.e. the encryption keys are stored to non-volatile memory (flash). This implies that a passkey exchange is required only once with the iOS device. The bonding information can be removed at any time by pressing ‘R’ on the terminal. Note that Bonding is called Pairing on iOS devices. If the bonding information is removed on the kit, the device should be unpaired on the iOS side as well, and vice-versa.

Expected Results

Build and program this example onto a BLE Pioneer Kit with the PSoC 4 BLE module (red colored module). Now you are ready to use this example. Follow these steps –

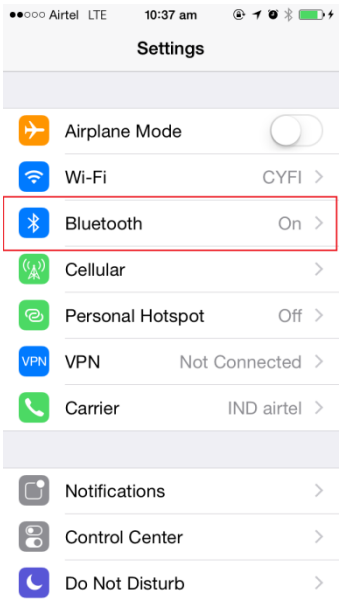
1. Open a terminal emulator such as Putty or Tera Term for the Pioneer Kit. The COM settings are: Baud rate – 115200 bps, Data bits – 8, Stop bits – 1, Parity – None.
2. Reset the device (by pressing the Reset switch) to see this output shown in Figure 2 on the terminal. The device is advertising at this point and waiting to be connected with an iOS device.

Figure 2. Terminal output



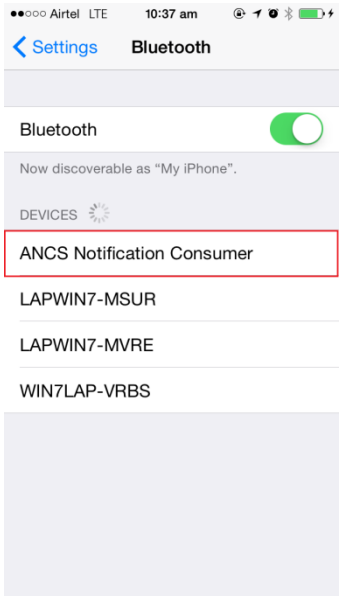
3. For this example, we are using an iPhone 5S. On the iPhone, navigate to the phone Settings as shown in Figure 3.

Figure 3. iPhone Settings



4. On the Settings page, enable Bluetooth. The BLE Pioneer Kit is listed as one of the devices nearby. Touch it to connect. See Figure 4.

Figure 4. iPhone listing the BLE Pioneer Kit



5. Once connected, enter the passkey on the iPhone, which is shown on the UART output. See Figure 5 and Figure 6.

Figure 5. Passkey Displayed on Terminal

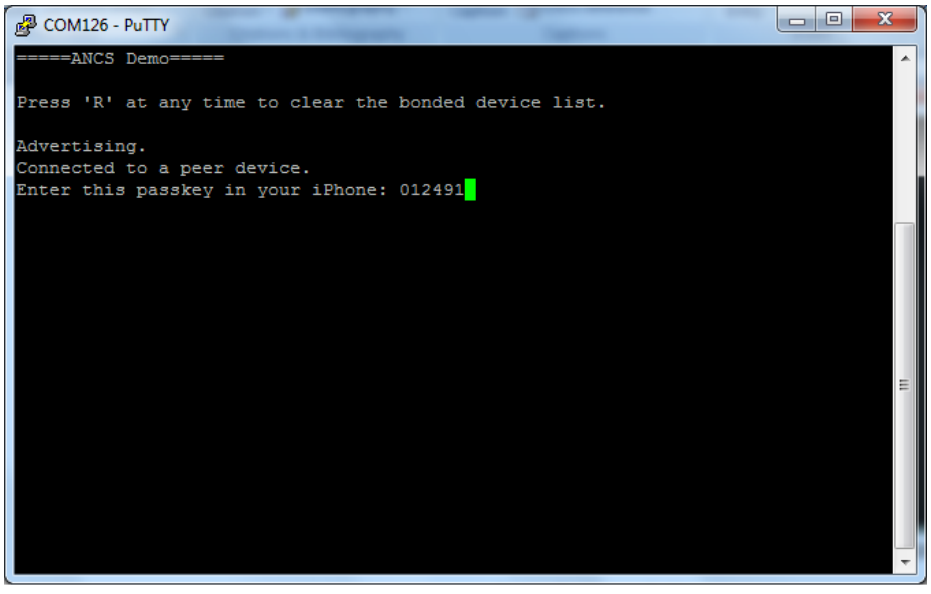
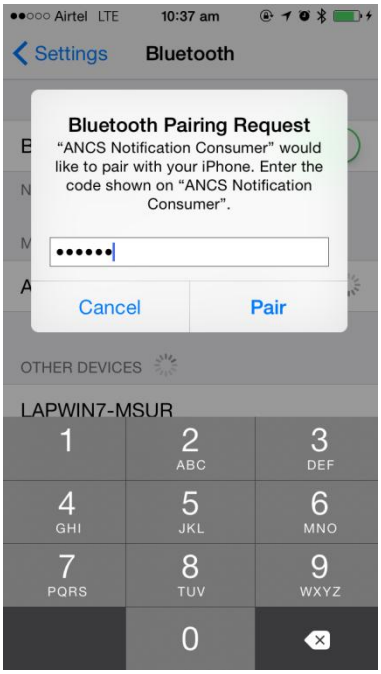
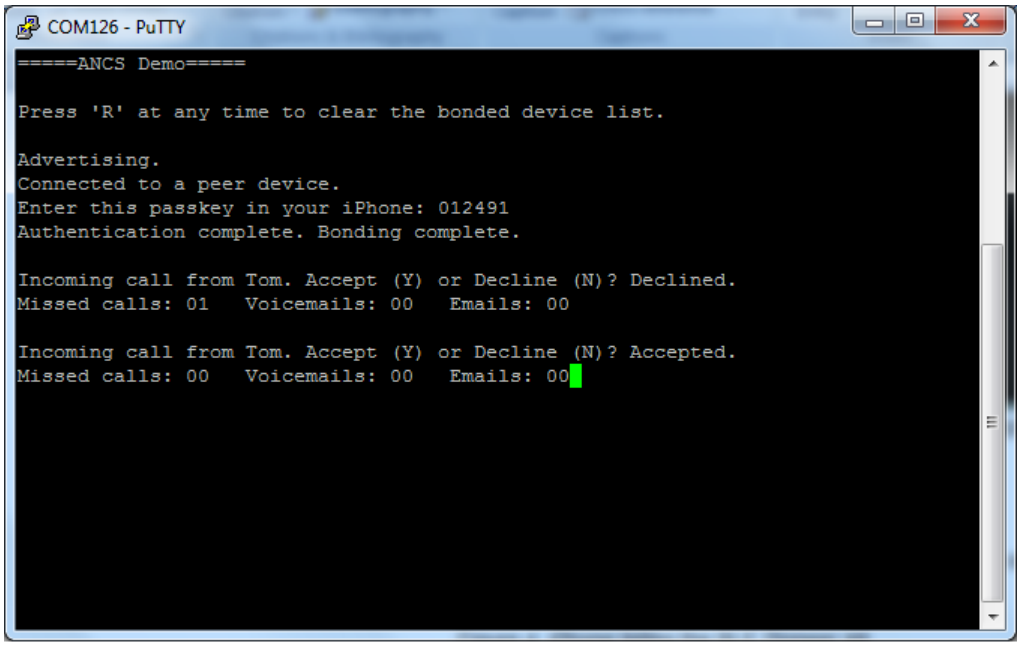


Figure 6. Passkey entered on iPhone



6. After the passkey is entered, the authentication is complete, followed by bonding.
7. Now when an incoming call is seen on the iPhone, the same is displayed on the UART terminal. You can accept the call by pressing 'Y', or can decline the call by pressing 'N'. See Figure 7. The number of missed calls, voicemails and emails are also shown.

Figure 7. ANCS – Incoming Calls, Missed Calls, Voicemails and Emails



8. To remove the bonding information, press ‘R’ on the terminal to clear it on the kit side, as shown in Figure 8. To clear it on the iPhone side, touch the radio button next to the device name and select Forget Device, as shown in Figure 9.

Figure 8. Clearing the bonded device list on the BLE Pioneer Kit

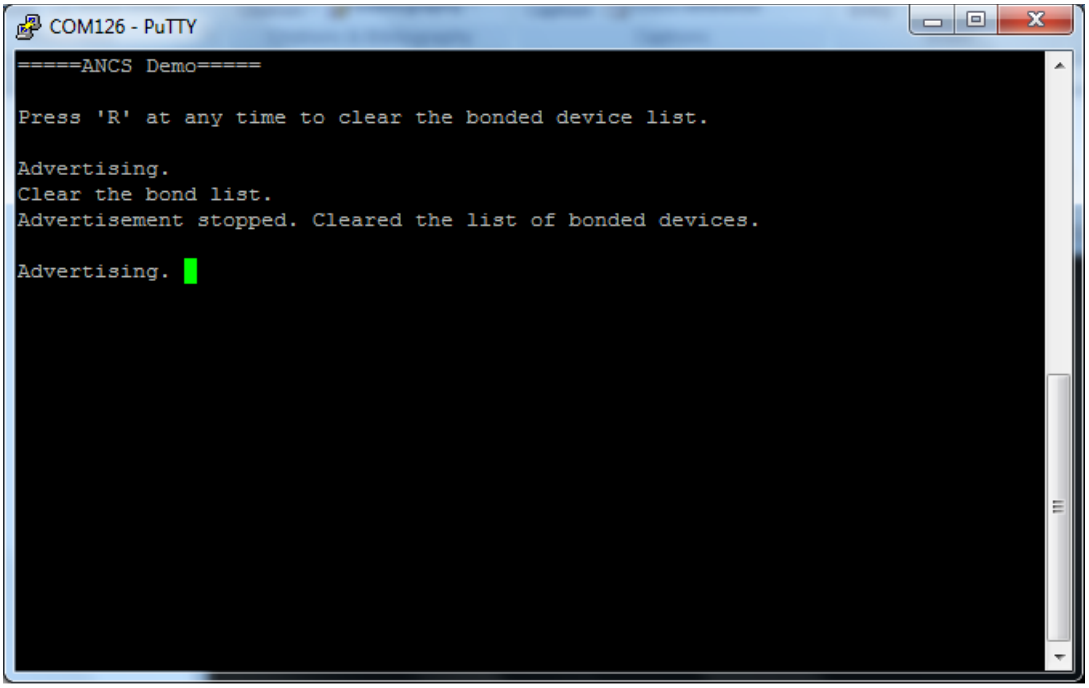
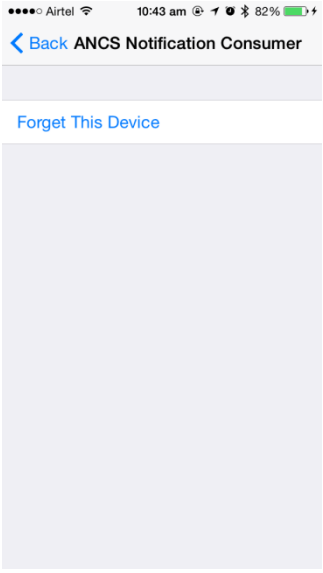


Figure 9. Clearing bonded device (called pairing) on iPhone



Related Documents

Table 1 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component / user module 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.