

Assignment#1-1: BLE (Bluetooth Low Energy) Host Programming

Due: 10/01/2015

Goal: After this assignment, you will have a basic understanding and skills to be able to make an host(master/central/client) application that can establish connection and communicate with most BLE devices(slave/peripheral/server).

Content Detail:

1. Scan, discover, and connect to Hamster via BLE only if RSSI > -50 dBm.
 - This will ensure that you need to bring your hamster close (5~10cm) to the BLE source.
 - Make sure to stop scanning after the successful connection.
2. Discover services, characteristics, descriptors and print out values/properties.
3. Read from "Sensors" and print out data (reading 20 bytes) in hex.
4. Write to "Effectors" with left LED green and right LED blue: writing 20 bytes where 1 byte is 00 (== 0x00)
Ex) <00 00 10 00 00 02 03 00 00 00 00 00 00 00 00 00 00 00 00 00>
5. Using notification, continuously get data (20 bytes) from "Sensors" and print out data in hex.
 - continuously: each connection interval notified by "Sensors"
6. Using #5, continuously write to "Effectors" with varying wheel speeds, LED colors, musical notes.
Ex) left and right wheels: 50% speed:
<00 00 10 32 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00>
7. Measure the communication rate(in hz) (== 1 / connection interval) between host and device.
8. Error handling code.

Hints:

In OS X, CentralManager == host/client/computer, Peripheral == slave/server/device(Hamster).

File: "hamster_class_api.py"

Open a terminal in Mac OS X.

Run the main app.

>python hamster_class_api.py

***** Note: if you get error about PyObjC, you need to install Xcode.

File: "hamsterAPI_ref.py"

Look at the required methods for delegates and available class methods in "hamsterAPI_ref.py".

Notice that this file is not a correct python file. This file lists relevant APIs that you should use.

--You need to implement the methods for both CBCentralManagerDelegate and CBPeripheralDelegate protocols.

--And you need to use CBCentralManager class methods and CBPeripheral class methods in delegate methods.

***** Note: this is event driven programming such that once you call the CentralManager class methods or Peripheral class methods, corresponding delegate methods are being called automatically.

Reference:

Mac OS X's Core Bluetooth Framework Reference

https://developer.apple.com/library/prerelease/ios/documentation/CoreBluetooth/Reference/CoreBluetooth_Framework/index.html#//apple_ref/doc/uid/TP40011295

PyObjC that is a bridge between Python and Objective-C.

<https://pythonhosted.org/pyobjc/core/intro.html>