

Turning a RPI 3 into a BLE peripheral and an iBeacon

Here our main target is to

1. Make the RPI into a BLE peripheral.
2. Make the RPI emit iBeacon.
3. Make the RPI scan for iBeacon.

With the help of the BLE and GPIO of the RPI we can control almost anything which is attached to it. Our first step was to make the RPI into a bluetooth peripheral so that it could be controlled by a smartphone over BLE. I followed a few tutorials for a start.

The tools used were as follows :

1. RPI 3
2. [Lightblue on IOS](#).
3. [Evothings Studio Software](#)

The tutorial I referred to mainly was as follows:

<https://www.hackster.io/inmyorbit/build-a-mobile-app-that-connects-to-your-rpi-3-using-ble-7a7c2c>

The initial part of the tutorial was installing Bluez. RPI 3 comes with inbuilt bluetooth and Bluez preinstalled but it is always good option to update the same. Also since the RPI compute module does not have bluetooth , it is required to install this package on the compute module. Next, follow the tutorial to install the Node.js application which make use of the Bleno library. Implement part one and two completely. Part 3 which is the mobile application is not implemented since I used LightBlue app to test the peripheral.

NOTE: While implementing the first part , when we are trying to stop the bluetooth system and then turning on hci0, there might be an error sometimes saying Cant init device hci0 : Operation not possible due to RF-kill (132).

If such an error occurs then implement the following steps:

```
rfkill list
sudo rfkill unblock bluetooth
rfkill list
sudo hciconfig hci0 reset
sudo hciconfig dev
```

after running these commands successfully the hci0 should be up and running.

If the tutorial is followed correctly the RPI should be able to advertise successfully by running the index.js file . This consists of three characteristics i.e load average, uptime and memory.

NOTE: Also refer to the following links for reference:

<https://tobiasturmm.de/2016/10/04/turning-a-raspberry-pi-3-into-a-bluetooth-low-energy-peripheral/>

<http://www.instructables.com/id/Control-Bluetooth-LE-Devices-From-A-Raspberry-Pi/>

The read and write to the peripheral was tested by running the code test.js under the folder node_modules bleno. This file has read , write and notify characteristics.

The second part would be to turn the RPI into an i-Beacon. When the RPI acts as an i-Beacon , it only advertises but it is not connectable. For behaving as an i-Beacon run the test-ibeacon.js file under the folder node_modules bleno. If everything works fine the RPI should be able to successfully advertise as an i-beacon.

I also tried to interface two bluetooth modules with the RPI. The approach in this would be to specify the bluetooth hci sockets. This would be done by passing in the device id's (0, 1 ..) into the bind methods.

Please refer to the following link : <https://github.com/sandeepmistry/node-bluetooth-hci-socket#bind>

pi3

```
sudo apt-get install git
cd ~
wget https://nodejs.org/dist/v5.9.1/node-v5.9.1-linux-armv7l.tar.xz
tar xfv node-v5.9.1-linux-armv7l.tar.xz
cd node-v5.9.1-linux-armv7l/
sudo cp -R * /usr/local/
```

pi2

```
wget https://nodejs.org/dist/v5.9.1/node-v7.9.0-linux-armv6l.tar.xz
tar xfv node-v7.9.0-linux-armv6l.tar.xz
cd node-v7.9.0-linux-armv6l/
sudo cp -R * /usr/local/
```

```
cd evothings-examples/examples/rpi3-system-information/rpi3-application
npm install
```