Bluetooth Low Energy: Smartphone App – Device interfacing

by **Satya K Vivek** | December 23, 2018

In the era of Internet of Things (IoT), user controls/ manages different systems using Smartphones. Bluetooth Low Energy (BLE) plays a major role mainly due to BLE interface availability on smartphones and its low power requirements. In this white paper, we will be dealing with technical intricacy in interfacing BLE to different smart devices from iOS and Android.

The concept of Internet of Things is continuing to revolutionize the way we interact with the devices that we use in our daily life. The devices that we use in our kitchen, bedroom, rest room or car, and the wearable devices that we use for fitness care or health care monitoring, everything is now going to be digitally connected to us. One of the key advantages is that we can connect, monitor and/or control these devices from our smartphones. Bluetooth Low Energy (BLE) is the most popular technology which enables users to connect to devices directly from any modern smartphone.

The biggest advantage of BLE is that all modern smartphones support BLE in both hardware and software. All modern mobile operating systems like Android, iOS etc natively support BLE and provides API layers for easy development of Apps. This allows device vendors to develop innovative Apps that can interface with their devices, receive data and provide users with the data in myriad of ways providing maximum user experience.

Another advantage of BLE is that the technology allows devices to be power efficient and hence battery operated. For example, your smart thermostat can work for several years on battery and at the same time and allow users to directly interact with their smartphone via BLE. In the past, this type of devices used ZigBee or Z-wave, which required a Hub to talk to Smartphone as smartphone does not support these technologies natively.

Smartphone App – BLE Device interfacing

In the IoT space, many of the devices will require a connectivity only when the user is in the direct vicinity. Devices used for healthcare monitoring, fitness care, home automation, climate monitoring etc are examples for these. These devices will mostly report some information to the user and/or provides options to control and configure the operation. Some of these devices also need occasional access to internet to upload usage statistics, logs etc to the cloud. Using Bluetooth Low Energy technology, all these use cases can be implemented using Apps running in your smartphone.

Device roles in BLE connection

Central and Peripheral roles

There are two different roles for devices in BLE as per the Bluetooth specification. In the classic Bluetooth, we are aware of the master and slave roles of devices for establishing a link. The phone or PC will be act as master which can scan for slave devices like Bluetooth earphone.

In BLE, there is a similar concept of central and peripheral roles for establishing link level connection. A peripheral can advertise its presence to other devices. A central device can scan for advertisement and initiate a connection with a peripheral device. When interacting with BLE devices, the smartphone will be in the central role and devices in peripheral role. This will enable the smartphone to connect and communicate with multiple devices at the same time using multiple Apps.

The peripheral can also enforce a secure connection by using the pass key mechanism. It can enforce a pass key which the central has to provide while establishing a pairing with the device. The user has to enter this pass key while connecting the device for the first time from the App.

Client and Server roles

Server is the device that provides data and client is the device that reads the data from server. In Smartphone App – BLE device interfacing, the client and server roles can be run on any device as per the use cases.

it is most common for a Peripheral to be a Server and a Central to be a Client. For example, a BLE Heart Rate monitor device is a server in a peripheral device providing the Smartphone App with the Heart Rate data.

It is also possible for a central device to run a server by providing a service. One example is the Time server which is run in a smartphone. A peripheral BLE device can access this service to get the real time from the phone. In many cases, the peripheral device will run both server and client roles simultaneously. For example, the Heart Rate device will provide the heart rate data service while running a client to get the time from the smartphone to sync its internal clock.

BLE service and profiles for App – device communication

The BLE specification defines a specific structure on the way data is exchanged between devices. The data is represented as a collection of state variables, such as battery level, temperature, heart rate etc. We can group state variables into services based on functionality. The Heart Rate Service,

for instance, is a collection of state variables including heart rate measurement and body sensor location.

The technical term for these state variables is “Characteristics”. For the sake of interoperability, each characteristic also holds a description of the value’s type. The characteristics supports data communication in both ways. I.e. the App can write a value to a characteristic in the device to control or configure it.

Services can be bundled together into a profile. For example, the Heart Rate Profile includes two services - Heart Rate and Device Information.

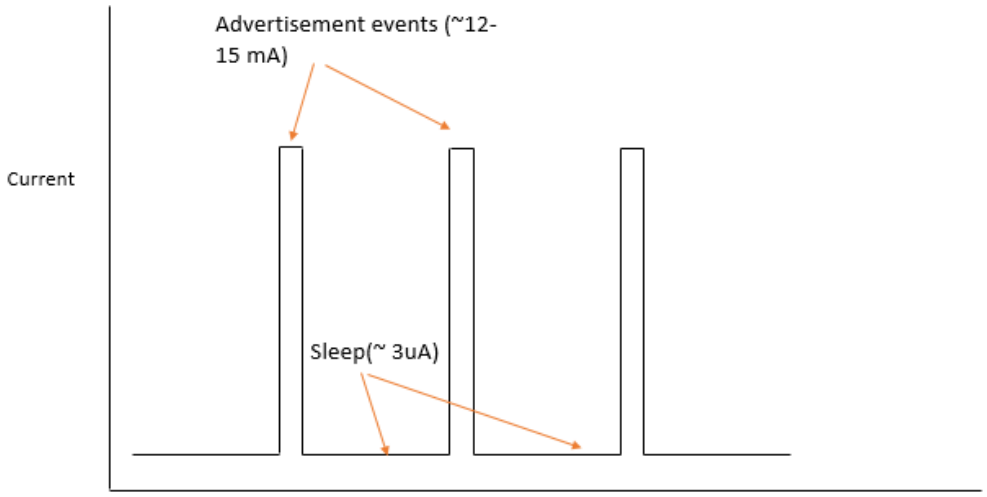
The Bluetooth Special Interest Group has defined some standard profiles for popular devices in the market. Some of the popular profiles are Blood Pressure Profile (BLP), Current Time Service (CTS), Cycling Speed and Cadence Profile (CSCP), Glucose Profile (GLP), Health Thermometer Profile (HTP), Heart Rate Profile (HRP), Weight Scale Profile (WSP) etc. Many of the BLE chip vendors like Nordic, Ti etc provide example implementations of these standard profiles. We can adopt these implementations if we want to develop such a device.

In most cases, a custom BLE profile need to be developed for a device. Let us consider developing a device for giving nerve stimulation therapy to patients. The user can connect to the device using an App to configure the therapy settings, get status of therapies, access logs of past therapies etc. Based on these requirements, we can define characteristics that represent the therapy settings of the device, status of therapies etc.

Fine Tuning of Advertising interval and Connection parameters for Low power operation

A BLE peripheral device advertise its presence to central devices by sending advertising packets periodically. This period is called Advertising interval. The current consumption during the advertising events will be high and between the advertising events the BLE device will be switched to sleep state to save power.

The advertising interval should be between 20 milliseconds and 10.24 seconds as per the Bluetooth specification. A higher advertising interval means the device saves more power. But this will also result in a delay for the Smartphone App to discover the device. So, the advertising interval should be set based on use case. For example, if you want the device to advertise indefinitely and run on a battery for a long time, the advertising interval should be set to an optimum value keeping power consumption in mind. But if the device advertises only when the user pushes a button, a fast advertising will not be a problem.



Another parameter that determines the power consumption is the connection interval. Connection interval defines how frequently a phone and device exchange data after establishing a connection. Between the connection event, the BLE chip will go sleep state.

A higher value for connection means less power is used, but this will also reduce the effective data transfer rate between the peripheral and central. A lower connection interval result in higher data transfer rate, but this also leads to higher current consumption in the peripheral device. This means, setting connection interval is a trade-off based on the use cases. But in most cases, the data rate required for BLE devices will be less, so a higher connection interval will be fine.

Challenges

While developing iOS Apps for Bluetooth, there will be some challenges in defining use cases depending upon whether the App is in foreground or background. An iOS app behaves differently in the background than in the foreground, because system resources are more limited in background.

By default, many of the common Bluetooth tasks are disabled while your app is in the background or in a suspended state. But you can declare your App to support the Bluetooth background execution modes to allow the app to be waken up from a suspended state to process certain Bluetooth-related events.

But still the App can't run forever in background. At some point, the system may terminate your app to free up memory for the current foreground app—causing any active or pending BLE connections to be lost. iOS Core Bluetooth stack provides options to save state information for the peripheral connections and restoring the state when the App is re launched.

In short, the interactions with the BLE device need to be carefully designed while developing iOS Apps.

Conclusion

Bluetooth Low Energy is the most popular technology that provides low power, low latency and low cost connectivity to devices in IoT. The biggest advantage of BLE is that all modern smartphones support BLE which enables users to interface with the devices directly from their phones. Device manufacturers can design Apps that will interact with the devices in the most seamless way to give the best user experience. Gadgeon has vast experience in developing BLE enabled devices and Android/iOS Apps that communicate with the devices for various applications like Personal health care, Home automation, Personal security systems, Climate monitoring, Smart mirrors etc. Gadgeon can provide BLE solutions using any of the major hardware platforms available in market like Nordic nRF, TI CC26xxx, Cypress PROC, Raspberry Pi etc.