BLE　Bluetooth Low Energy

低消費電力　ボタン電池で数年稼働

低コスト

低帯域幅

低複雑度

Bluetooth Low Energy：Android Developers

http://developer.android.com/intl/ja/guide/topics/connectivity/bluetooth-le.html

2.4GHz から 2.4835GHz までの 40 チャネル

無線変調速度　1Mbps

コネクション間隔（データをやり取りしてからアイドル状態になるまでの時間）　7.5ms~4s

最大スループット　5~10KB/s 程度

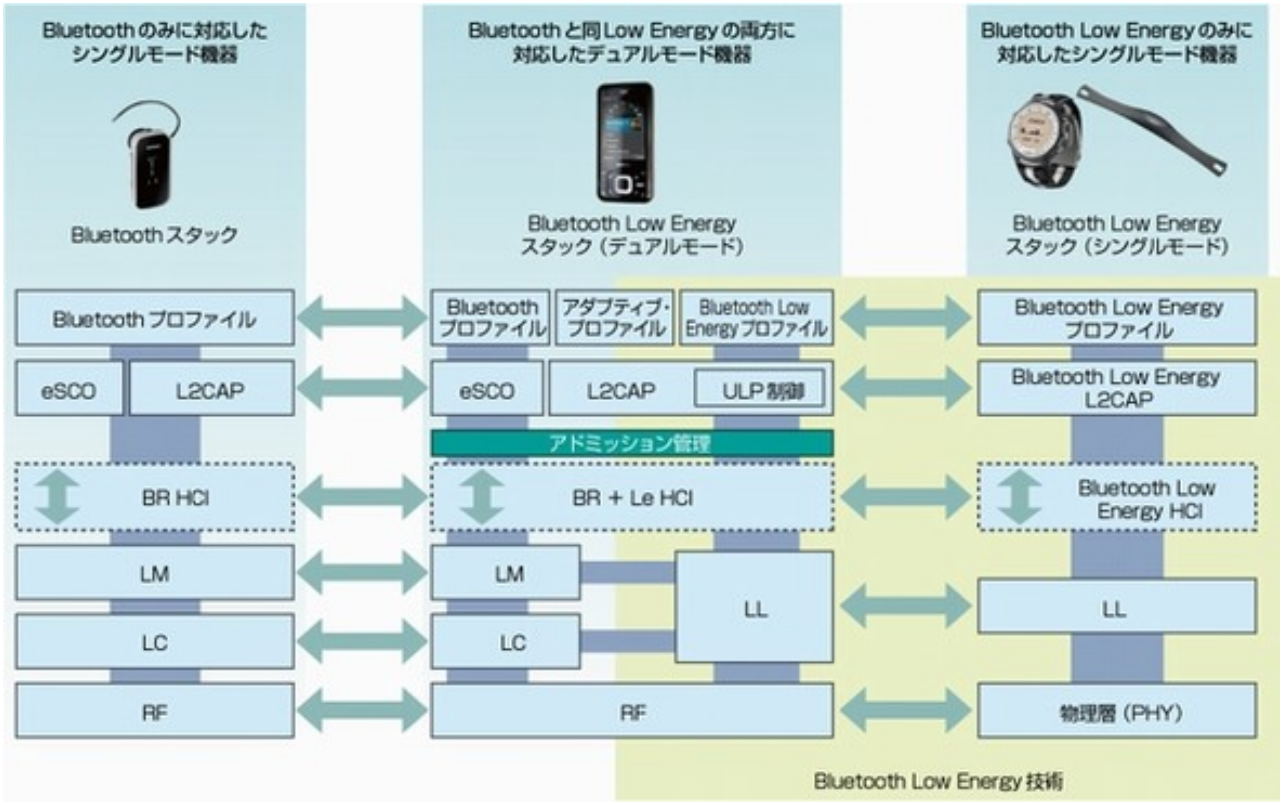


図 3　Bluetooth Low Energy では、それのみに対応するシングルモードの実装と、Bluetooth にも対応するデュアルモードの実装のいずれも可能　図中の略号は以下の通り。L2CAP：論理リンク制御および適合プロトコル、HCI：ホスト・コントローラ・インターフェース、LM：リンク・マネジャー、LC：リンク・コントローラ、LL：リンク層　（http://eetimes.jp/ee/articles/0906/30/news096.html）

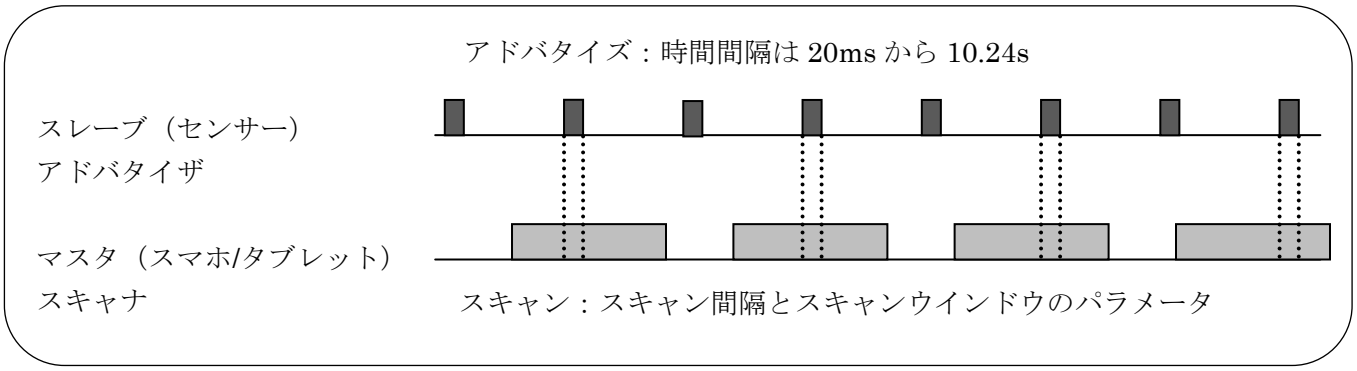
コネクション：マスターとスレーブ

コネクションを開始するデバイスがマスターとなり，可用生をアドバタイズしてコネクションを受け付けるデバイスがスレーブとなる。

①マスターがスキャンを開始して，コネクション要求を受け付けているアドバタイザを探す。

②マスターは目的のアドバタイザを検出したら，そのスレーブへコネクション要求を送信し，スレーブが応答すれば，コネクションが確立する。

通常は，スマホやタブレットがマスターで，センサーがスレーブとなる。スレーブはマスターのタイミングに従う。



BLE の 2 種類のパケット：アドバタイズ・パケットとデータ・パケット

アドバタイザ（広告者）
↓アドバタイズ・パケット
スキャナ

データはサービスでカプセル化され，サービスには 1 個以上の特性（characteristics）と呼ばれる単位から構成される

Bluetooth デバイスアドレス

パブリックデバイスアドレス（工場設定）とランダムデバイスアドレス（論理的）があり，ホストから指定された方を利用する。

汎用アクセス・プロファイル（GAP：Generic Access Profile）

役割

Broadcaster

関心のある任意のデバイスへ温度の測定値をブロードキャストするパブリックな温度計は，Broadcaster の好例である。Broadcaster はアドバタイズ・パケットでデータを送信するため，待ち受け状態にあるあらゆるデバイスで利用できる。

Observer

ブロードキャストしているデバイスからデータを収集する，受信にのみ特化した役割が Observer である。

Central

Central 役割はリンク層のマスターに対応する。Central は他のデバイスのアドバタイズ・パケットを待ち受け，次に選択したデバイスとコネクションを開始する。このプロセスを繰り返し，1 つのネッ

トワークに複数のデバイスを参加させる。

汎用アトリビュート・プロファイル (GATT:Generic Attribute Profile)

データの転送手順とフォーマット

Central vs. peripheral. This applies to the BLE connection itself. The device in the central role scans, looking for advertisement, and the device in the peripheral role makes the advertisement.

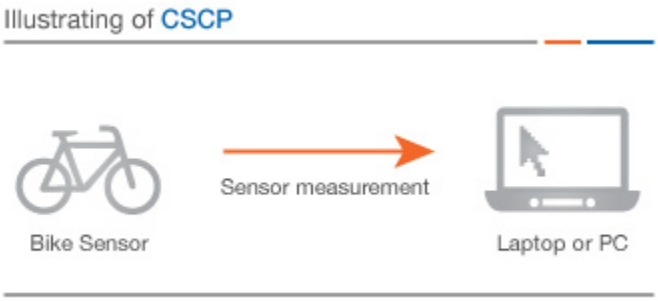
GATT server vs. GATT client. This determines how two devices talk to each other once they've established the connection.

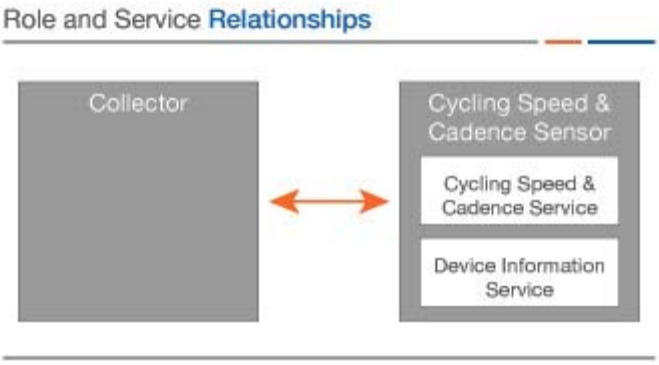
Cycling Speed and Cadence Profile

CSCP Cycling Speed and Cadence Profile 1.0 21 August 2012

CSCS Cycling Speed and Cadence Service 1.0 21 August 2012

Sensor : GATT server Collector : GATT client.





BLE 通信アプリ

①Setting Up BLE

device の BLE を利用可能状態にする

②Finding BLE Devices

startLeScan()

特定の peripheral タイプの scan する場合は以下のメソッド

startLeScan(UUID[], BluetoothAdapter.LeScanCallback)

③Connecting to a GATT Server

mBluetoothGatt = device.connectGatt(this, false, mGattCallback);

④Reading BLE Attributes

⑤Receiving GATT Notifications

onCharacteristicChanged()

device で特定の characteristic が変化したら知らせる

⑥Closing the Client App

close()

```
// Setting Up BLE：
// 1.Get the BluetoothAdapter
//Initializes Bluetooth adapter.
private BluetoothAdapter mBluetoothAdapter;
final BluetoothManager bluetoothManager = (BluetoothManager) getSystemService(Context.BLUETOOTH_SERVICE);
mBluetoothAdapter = bluetoothManager.getAdapter();
// 2. Enable Bluetooth
// Ensures Bluetooth is available on the device and it is enabled. If not,
// displays a dialog requesting user permission to enable Bluetooth.
if (mBluetoothAdapter == null || !mBluetoothAdapter.isEnabled()) {
    Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
}
// Finding BLE Devices
/**
 * Activity for scanning and displaying available BLE devices.
 */
public class DeviceScanActivity extends ListActivity {
    private BluetoothAdapter mBluetoothAdapter;
    private boolean mScanning;
    private Handler mHandler;

    // Stops scanning after 10 seconds.
    private static final long SCAN_PERIOD = 10000;
```

```
    ...
    private void scanLeDevice(final boolean enable) {
        if (enable) {
            // Stops scanning after a pre-defined scan period.
            mHandler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    mScanning = false;
                    mBluetoothAdapter.stopLeScan(mLeScanCallback);
                }
            }, SCAN_PERIOD);

            mScanning = true;
            mBluetoothAdapter.startLeScan(mLeScanCallback);
        } else {
            mScanning = false;
            mBluetoothAdapter.stopLeScan(mLeScanCallback);
        }
        ...
    }
...
}
```