



モバイルコンピューティング推進コンソーシアム  
Mobile Computing Promotion Consortium

**MCPC TR-028**

**Bluetooth LE Technical Reference  
for Interoperability**

**Version 1.00**

**2021 年 6 月 4 日**

**モバイルコンピューティング推進コンソーシアム  
技術委員会**

*CONFIDENTIAL*

## Draft Only Page

### Draft 変更履歴

日付	Version	変更内容
2020年11月18日	0.90	Draft Release
2021年2月17日	0.91	2021年1月度SWGのレビュー結果を反映
2021年3月18日	0.92	2021年2月度SWGのレビュー結果を反映
2021年3月30日	0.95	2021年3月度SWGのレビュー結果を反映
2021年5月19日	1.00	2021年4月度SWGのレビュー結果を反映

### Revision of This Template

日付	Version	変更内容

### ドキュメント発行者、および著作権者:

〒105-0011

東京都港区芝公園3-5-12 長谷川グリーンビル2F

モバイルコンピューティング推進コンソーシアム (MCPC)

電話: 03-5401-1935

FAX: 03-5401-1937

EMAIL: [office@mcpc-jp.org](mailto:office@mcpc-jp.org)

WEB SITE: <http://www.mcpc-jp.org>

### 機密保持について:

MCPC会則、MCPC IP Policyを遵守

### 免責について:

本ドキュメントはモバイルコンピューティングに関する標準仕様、推奨仕様などを提供するもので、モバイルコンピューティング推進コンソーシアム(以下MCPCとする)は、本ドキュメントを使用した結果発生した損害、第三者の特許、またはその他の権利の侵害に対して、一切の責任を負わない。また、本ドキュメントはMCPC、または第3三者が保持するいかなる権利のライセンスを許諾するものではない。

### 2進数、10進数、16進数の表記方法:

- 2進数は小文字”b”を付加する。(例: 10b)
- 2進数4桁以上は4桁ごとにスペースで区切る。(例: 1000 0101 0010b)
- 16進数は小文字”h”を付加する。(例: FFFFh and 80h)
- その他の数字表記は10進数とする。

### キーワード

- することができる  
してもよい  
(may) 推奨または要求に自由な選択肢を示す。
- すべきである  
(should) 必須ではないが強い推奨を示す。実施の際、必須ではないが考慮すべき。
- しなければならない  
(shall) 必須要求を示す。接続性、仕様準拠のために必ず実施しなければならない。

# Table of Contents

<b>1. Introduction</b> .....	<b>1</b>
<b>2. Bluetooth LEシステム/デバイスのIOP Issues</b> .....	<b>2</b>
2.1 ペアリング/接続 (Category-A) .....	2
2.2 データ通信 (Category-B) .....	5
2.3 アプリケーションその他 (Category-C) .....	7
<b>3. 推奨動作</b> .....	<b>8</b>
3.1 ペアリング/接続 (Category-A) .....	8
3.2 データ通信 (Category-B) .....	12
3.3 アプリケーションその他 (Category-C) .....	14
<b>4. デバイス実装における推奨値と対応</b> .....	<b>16</b>
<b>5. Terms and Abbreviations</b> .....	<b>19</b>
<b>6. References</b> .....	<b>19</b>

# 1. Introduction

近年、Bluetooth LE (Bluetooth Low Energy) を搭載した医療/ヘルスケア/スポーツ&フィットネス系のセンサやウォッチ等のデバイスが数多く出ており、同じくBluetooth LEに対応したスマホ/タブレットとの間でのIOP問題が報告されている。

本ドキュメントでは、MCPC互換性検証WS等を通じて抽出された問題点および技術課題等を記述したものである。また、Bluetooth SIG規定のBluetooth LEサービス/プロファイルの実装、Bluetooth LEシステム/デバイス固有の独自アプリの開発において、システム/デバイスのIOP上支障となり得る問題事象について、問題回避策も記述する。

## 2. Bluetooth LEシステム/デバイスのIOP Issues

### 2.1 ペ어링/接続 (Category-A)

No.	問題点	推定原因
A1	Android 標準の設定アプリ (settings) でペ어링し、専用アプリで通信後に専用アプリを停止すると、ペリフェラル側がデータ送信を継続しセンタ側でデータが失われる	専用アプリの Indication 設定が有効になっており、設定アプリに対して送信し続けるため。
A2	Android で標準の設定アプリ (settings) でペ어링→接続→切断後、専用アプリから接続→データ受信した後に、専用アプリを停止すると、ペリフェラルから測定データが設定アプリに送信されてしまう	ペ어링実績のあるスマホアプリが複数存在し、ペリフェラルからは送信先のアプリケーションをサービス単位でしか指定できないため。
A3	ペ어링→接続完了後の初期手続きが完了しない	初期手続きの中でペリフェラル側の PXP コマンドをセンタ側がパラメータエラーと判定し、その後の処理が動作しない
A4	接続が完了しない	ペ어링によるリンクキーは生成しているが Encryption に失敗している
A5	パスキーエントリの Paring Random でエラーが出ている	原因不明
A6	SMP を Just Works で動かすと、エアログ上 Authentication まで問題は見られないが、Connect 後にセンタから必ず切断される	原因不明
A7	専用アプリでは見つかるが、Android のデフォルト Bluetooth画面では見つからない	Android のデフォルト Bluetooth 画面に認識させるには、Advertise Data にローカルデバイス名が必要
A8	専用アプリでは見つかるが、Android のデフォルト Bluetooth 画面では見つからない	Android のデフォルト Bluetooth 画面に認識させるには Advertise Data の AD Type で Flag を設定し、BR/EDR Not Supported と LE General Discoverable Mode の Bit を立てる必要あり
A9	ペリフェラルから SMP 認証を行ったが、Android から拒否された	原因不明
A10	Service 検索において、機器のサービス情報を取るタイミングは決まっていない。ペリフェラルが時計の場合に、接続時のサービス情報取得で、スマホから途中切断するとサービス情報が取得できず、つながらなくなる。スマホではアプリから切断されることがある	プロフィールではペ어링時にすべてのサービス情報取得を規定しているが、独自仕様においてはデバイス情報およびサービス情報の取得方法をアプリで規定する必要がある 切断される事象は、Android の設定アプリはリンクキー生成後に切断する動作仕様があり、その影響と考えられる。
A11	ペ어링中にリンクロスすると、動作が再開しない	リンクロス後の動作に関する規定なし 特に動作規定がないため、対向デバイスが接続可能な状態となるかは決まっていない。動作仕様として規定する必要がある。

A12	スマホの接続待ちとセンサ機器の接続開始のタイミングが難しい センサ機器操作だけでデータ送信する場合、スマホを見ないのであれば転送に時間がかかっても問題ないのではないか	センサ機器側から接続するシナリオでも、Central (スマホ側) から接続しなければいけない。プロフィールではAdvertising, Scanning のタイミングを規定している。
A13	接続に時間がかかる場合、デバイスとセンサの距離が届かなくなった時に送信失敗する可能性があるのでは？	可能性はある。接続タイムアウト、再接続に関する仕様を規定する必要がある。
A14	できるだけ早く接続したい	両方(スマホ側とセンサ機器側)にアクションがある場合、スマホ側のスキャンインターバルとセンサ機器側の Advertising の送信間隔を短くする。受け側はScan Interval が Advertising Interval の2倍はないと接続がすぐにできない。
A15	早く接続したい時と、早くつながらなくてもいい時がある	両方を操作したときに必ず接続・データ送信できるというユーザ保障がある方が好ましい。センサ側の操作が行われたときは、センサ側のAdvertisingを送信する間隔を狭くするよう送信間隔を決める必要がある。
A16	複数のセンサを1つのスマホで接続するときはどうするか？	問題点が不明
A17	複数デバイスと接続するときで1つのデバイスで複数サービスがあるときはどうするか？	不明
A18	Healthcare 機器のみをデバイスリストに表示するフィルタは可能か？	Advertising のデータにサービス UUID だけを取得できる仕組みがある
A19	Whitelist の使い方？接続相手を限定？	センサのユーザ ID 毎に接続先を仕分けするときに見える？
A20	デバイスが別機種とのペアリングを行おうとして、Undirected Connectable でAdvertisingをすると、ペアリング済みの機種と接続してしまう機種が有り、別のスマホとペアリングできない	Android の設定アプリでペアリングを行なった場合に起きる
A21	アプリケーションによって接続しやすいものとしにくいものがある。受信端末のBluetoothをOFFにしてから再度ON にすると接続しやすくなる	アプリによって Fast Connect 中の動作に違いがあり、Android は細かく制御する必要がある Bluetooth Off → ON でつながり易くなるのは、Fast Connection から始めるためと思われる
A22	Security Request 受信時、Central側に Key が残っていると Encryption を開始するが、Peripheral 側にキーがない場合 Encryption に失敗し自動切断され、接続確立は継続的に失敗する	Android 4.4.4 では、Peripheral の Security Request に対しCentral 側から再ペアリングを起動しないので、Central 側でキー削除を促し再ペアリングを要求する必要がある。Android 5.0.2 では、Central 側が再ペアリングを起動するのでキーを削除する必要なし。Qualcomm 製 Snapdragon を使用するスマホやQualcomm 製スタックをそのまま使用するスマホのうち Android 6.0.0 以降の機種では、Security 上の理由で再ペアリングが起動されなくなり、Android 4.4.4 と同様に Central 側でキー削除を促すアラートを出す必要がある

A23	Phone 側でペアリング情報を削除したらペアリングできなくなった。繰返しペアリング要求したら、5回目に成功した	デバイス側にペアリング情報削除のスイッチがなく、何回か Pairing Fail を繰り返した後に成功する仕様としている
A24	Pairing 実行時に IO Capability が両者共 KeyboardDisplay の場合、Peripheral 側の Security Request に対して Pairing Fail を返す場合がある	原因不明 Central 側の問題と考えられる
A25	Pairing 後に Peripheral 側でキーを削除して再接続を試みると、Central 側で Connecting のままとなる	Report 送信後、設定値の期待する動作と異なる動作をしていた Report Map の調整により修正 (HOGP/HID) Central 側で明示的にキーを削除すれば回避可能、もしくは Central が Encryption に失敗したら Pairing 情報を破棄するよう実装されていれば回避可能 (GAP)
A26	Android 5.0.2 の IO Capability が DisplayYesNo になっていて、Passkey 接続ができない。Android 4.4.2 では DisplayKeyboard の設定で問題なかった	Core 4.0 から SMP で KeyBoardDisplay という IO Capability が規定されていて、DisplayOnly/DisplayYesNo デバイスで Passkey Entry になるためには、Initiator が KeyboardOnly/KeyboardDisplay であることが必要あるいは Slave からの Pairing の前に Security Request Command (MITM Protection required) を送る
A27	Android 5.0 で Scan Filter がサポートされ、複数のサービスについて Scan 設定が異なると問題になる	—
A28	Android 5.1 仕様で、Service Discovery に時間がかかり過ぎていたのを改善すべく、Service Discovery 時だけスマホから Connection Interval を 7.5ms に設定し一部の機器で失敗するケースがある Service Discovery が終了すれば、アプリが指定した Interval に戻される	Service Discovery 時の Connection Interval は Android で決めていて変更できない。デバイス側は SDP の最中に暗号化処理を並行して実施しており、CPU の処理負荷のため間に合っていないように見えるデバイス側で頑張るしかない
A29	Android でペアリング時にインストラクションがポップアップで表示されるケースと、画面上部に表示されるケースの二通りがあり、ユーザに分かり難い	Androidの仕様に依存する

表1. Bluetooth LE IOP Issues -ペアリング/接続-

## 2.2 データ通信 (Category-B)

No.	問題点	推定原因
B1	SMP 認証後、Android の GATT DB Write 要求に応答するが、その後 Android から GATT 要求が来ない	原因不明
B2	ペアリングに引続き、接続時に CCC の設定を行う必要があるか？ 暗号化前も OK？	ここではデータを送らないので CCC の設定は行わない。CCC の設定は受信側のデータ受信許可を意味し、データを受信できない時は、サーバに CCC=0をセットする
B3	デバイスのメモリ送信をどのように扱うか？ HealthCare のデータ管理は？	Bluetooth LEの基本動作は、データを持っている方がサーバとして動く。センサが複数のサーバと接続する場合、データ送信後センサのデータは消去し、同期はクラウドで行うのが望ましい(?)。Bluetooth LE センサの消費電力から、毎回全データの送信は難しいので、差分送信が好ましい
B4	Notification で送信する場合はリンクが切れた時、どこまで送ったかわからなくなる	すべての保存データを毎回送る(?) 保存データを消去しない(?)
B5	血糖計はデータを消去せずすべて保持する必要がある	血糖値計はスマホに送信後も、病院でも送信したいというユースケースがある。血糖値計はデータを送信しても消さずに全部持っている仕様になっている。
B6	活動量計を使う時に、どれくらいの頻度で送信するか、どの程度のデータを保存するか、消去する時はどうするか等	ユースケースはいくつかに落ち着かずだからプロファイルで決めればいいのか？
B7	ある程度大きいデータを送信する時は消費電力が大きくなるため、圧縮が必要	圧縮・伸長の方式も決める必要あり OTP/OTS v1.0 がリリースされているが、Core4.2が必要で、Classic でも動作することが求められるさらに、HPS についても考慮が必要 OTP を使用しても、GATT で無手順送信するのと大きな違いはないかもしれない また Bluetooth LE OTP でスループットを求めたり、Beaconによる割り込みが許容できないようなシステムにはお勧めできない
B8	リアルタイムデータ転送の最高通信速度は？	実装仕様に依存する
B9	リアルタイムデータ転送において測定後にセンサ機器から送信する時のセンサ側のインターバルを決める必要あり	スマホ側のアプリがフォアグラウンドにある時にはセンサ側の Advertising のインターバルの 2倍が必要？ Advertising の間隔は乱数で決まっているため、同期して受信できない動作を避けている？
B10	通信終了時は、Client と Server のどちらから行うか？(規定されていない？)	Client 側から切断、Server 側はtimeout を設定 プロファイルに規定されていないケースもあり、再接続時の動作についても検討が必要となる
B11	接続後すぐにデータを送信 (Indication) すると繋がらない場合がある。アプリから CCC の設定を毎回行う必要がある。CCC の設定タイミングをどうするか？	Android の場合、設定でペアリングする場合と、アプリでペアリングする場合で動作に差がある iPhone は Bluetooth LE ではアプリからのペアリングのみ

B12	CCC は初回接続時に行われ、デバイス側が覚えていて、次回 Indication で接続時に直ちに送信すると受取れない場合がある。デバイスは Delay が必要？	CCC 設定は受信側のデータ受信許可を意味するデータを受信できない時は、サーバに CCC=0 をセットする？
B13	設定でペアリングした場合、デバイス側から Indication でデータを送るとアプリが立ち上がっていても、データを受け取ってしまうさらに、Confirmation の応答を送ると、データが失われてしまう (Android)	専用アプリの Indication 設定が有効になっており、設定アプリに対して送信し続ける
B14	Android 5.0 で、Indication のデータがアプリに届かない。Confirmation は返っている (以前にもあった問題)  Android 側でOS 起動の自動サービス検索中に Indication/Notification を送信すると、応答は返るが、スマホアプリにまで通知されない Android5.1 で解消されたが、6.0 で再現した	接続の度に Group type request が Android から要求され、Indication のデータがアプリに届く場合と届かない場合があり、機種によって動作が異なる Confirmation が返る場合と返らない場合の両方のケースがある。 5.0 の問題は、5.1 でスマホメーカーがペアリング後に自動サービス検索が起動されないよう修正した 6.0 で再現したのは Android の問題かもしれない。接続時にペアリングまで実行し、スマホアプリからサービス検索すれば OS が実行する自動サービス検索は回避できる。なお、検索結果に変更のない場合コールバックが返らないケースもあるので、GetServices() を使用するのが好ましい
B15	血糖値計で、サーバへリアルタイムにデータ送信する場合、介在するネットワークのタイムアウトが問題になる インターバルを長くして (10min) アプリがリアルタイムに応答できるか？	データをクラウドのサーバに上げてから血糖値計のデータを削除する構成にすると、タイムアウトが問題になる。スマホがサーバとの通信を保証し、擬似サーバとして機能するシステムにすべき？
B16	スマホから Peripheral 側のデバイスに連続してパケット送信 (Write Command) すると、時折パケット送信されないことがある	Android SDK で API をコールする処理は成功しているように見えるが、Android の Snoop Log では送信の処理間隔を空けないと時折パケットが送信されない

表2. Bluetooth LE IOP Issues -データ通信 -

## 2.3 アプリケーションその他 (Category-C)

No.	問題点	推定原因
C1	MCPC 互換性検証 WS 中にテスト用の通信アプリをスマホに設定するのに時間がかかり、テストに十分な時間を割くことができない	スマホアプリに関する情報が事前にスマホ側に伝わっておらず、テストセッションの最初に情報を聞き出して設定している
C2	デバイスを操作するコマンド、サービス定義が必要。メモリクリア、アンペアリング、メモリの有無設定等、センサに関する基本設定をアプリから行う機能。	データを消す時にはユーザ確認がないと担保という意味で消しづらい。ユーザにデータの消去確認を行うコマンドを定義しておく方がよいかも。コマンドで全メモリ送信、差分送信を選べるようにしたらどうか。
C3	一つのデバイスに対して複数のユーザが使用するときはどうするか？	アプリ毎に接続相手を規定する ペアリングは複数行うとして、アプリ毎にWhite listを設定する(デバイス側で対応)
C4	複数ユーザ ID 有り、マルチユーザ対応デバイス	ID 毎にペアリングする
C5	複数ユーザ ID なし、シングルユーザ対応デバイス	個人を特定できないので、複数ユーザが使う前提の場合はメモリデータなどを持たない方がよい 誰かが測定してしまう場合は、受信側、クラウド側でデータ削除をする必要がある
C6	複数のアプリが動いている場合のアプリとデバイスの接続、ペアリングにおいてAdvertisingからアプリの特定ができるか？	実装依存する
C7	スマホ OS のメジャーアップデートに伴いスマホアプリが全く動かなくなるケースがある	API 仕様が変更され、互換性が保たれない
C8	HOGP サポートデバイス同士で、特定のロールしか持たないデバイスはそれ以外のロールのとは動作しない	サービスでは何がつながるかわからず、接続してはじかれて初めて判断できる
C9	FMP クライアントで、自機がサポートしている Service をすべて立ち上げていると Alert が鳴らない	自機のサービスをすべて停止すると Alert が鳴るので、何かの Service と競合していることが考えられる(サーバ側の実装の問題)
C10	バックグラウンドで動作しているアプリを含めアプリ毎の設定が可能だが、API 仕様を詳細に検討する必要がある	問題点が不明

表3. Bluetooth LE IOP Issues -アプリケーションその他-

### 3. 推奨動作

前章で提示した課題を解決するための推奨動作を以下に示す。

#### 3.1 ペ어링/接続 (Category-A)

Recommendation A1:

##### 【問題事象】

Android標準の設定アプリ(settings)でペ어링し専用アプリで通信後、専用アプリを停止すると、ペリフェラル側がデータ送信を継続しセンター側でデータが失われる。あるいは、Android標準の設定アプリでペ어링→接続→切断後、専用アプリから接続→データ受信し専用アプリを停止すると、ペリフェラルから測定データが設定アプリに送信されてしまう。

##### 【想定原因】

複数アプリでUUIDが衝突し、意図したアプリとは異なるアプリがペリフェラル機器からのデータを受信してしまうことが原因。ペリフェラル機器の実装によるがデータを送信したことを契機にペリフェラル内のデータを削除する実装にしていれば、申告にあるようにデータが失われる。

##### 【対策】

Android 標準の設定アプリ "Bluetooth 設定" はなるべく使用せず、スマホの専用アプリでペ어링から処理する。あるいは、Bluetooth 設定で作成したペ어링情報を専用アプリへ移し、Bluetooth 設定の登録情報をマスクする。

##### 【関連する課題】

A2,

Recommendation A2:

##### 【問題事象】

Advertising データには、ローカルデバイス名を追加し、AD Type の Flag で BR/EDR Not Supported と LE General Discoverable Mode のビットを立てる。Privacy を考慮する場合はその限りでない、初回のペ어링時には必要である。ただし、スペック 10.7.2 には Privacyに関する記述がある。

Motivation:

Android 標準の設定アプリ "Bluetooth 設定" でデバイスが検索できない。

##### 【関連する課題】

A7、A8

Recommendation A3:

Bluetooth LE機器と初回接続する場合、クライアント側がデバイス情報とサービス情報を取得するようアプリで規定する  
非ストリーミング機器と接続するときに操作が入った場合にはインターバルの間隔を短くする。

Motivation:

プロファイルではペアリング時にすべてのサービス情報取得を規定しているが、独自アプリではデバイス情報およびサービス情報の取得方法を動作仕様で規定する必要がある。

【関連する課題】

A10

Recommendation A4:

ペアリングエラーの発生時、再ペアリングを促す処理を実装する。Central側は再ペアリングと同時にペアリング情報もクリアする。

Motivation:

ペアリング中のリンクロス等に対し、エラー発生後の処理が規定されていないため動作が再開しない。

【関連する課題】

A11

Recommendation A5:

センサ機器側から接続するケースや接続するタイミングを制御したいケースにおいて、GATT 上の独自アプリで Fast Connect と Reduce Connect の使い分けを明確にする。  
Note. 設定値を明記する(ユースケース毎に分ける)

Motivation:

プロファイルでは、Indication、Advertising、Scanning のタイミングやインターバルを規定しており、独自アプリについてもこれを参照して実装する必要がある。

【関連する課題】

A12、A14、A15、A21

Recommendation A6:

接続タイムアウトおよびリンクロス後や切断後の再接続に関する標準手順を規定する。

Motivation:

Bluetooth LE では BR/EDR の Supervision Timeout に相当する動作仕様が規定されていない。  
→ Supervision Timeout の規定はあるが、再接続の手順は規定されていない  
(BR/EDR も同様)  
→ 「切断後に双方のデバイスが Advertise と Scan を実行すること」といった記載を追記するか？

【関連する課題】

A13

Recommendation A7:

特定の機器だけをデバイスリストに表示したい場合、Advertising データのサービス UUID だけを取得する仕組みを利用する。ただし、信頼できる結果が得られるかどうか、独自アプリでも適用可能かは検証が必要。  
独自サービスで 128bit UUID を使用すると Advertising データに1個の UUID しか載せられない

Motivation:

BR/EDR における Class of Device と同様の使い方を想定

【関連する課題】

A18

Recommendation A8:

センサの ID 毎で接続先を仕分けるケースでは、Whitelist を使用する

Motivation:

接続相手を限定したい場合を想定

【関連する課題】

A19

## Recommendation A9:

Android 4.4.4 以前で Central 側に Key が残っていて Peripheral 側にキーがない場合、Central は Security Request 受信に対しユーザアラートでキー削除を促す。ただし、Android 5.0.2 以降では Central が再ペアリングを起動するのでキーを削除する必要なし。Android 6.0.1 でセキュリティ上の理由により再ペアリングを起動しない仕様に戻ったため、Android 4.4.4 と同様にユーザアラートでキー削除を促す必要がある。

## 【修正案】

ペアリング(暗号化を含む)を実施すると、セントラルであるスマートフォンのOS、アプリ(実装依存)、ペリフェラルであるデバイス(実装依存)にそれぞれペアリングに関するキーなどの情報(以降ペアリング情報)が残るため、別のスマートフォンと新規ペアリングを実施したい場合や、同じ組み合わせでも何らかの要因によって再度ペアリングを実施したい場合は、各実装にあわせてそれぞれが保持しているペアリング情報を消去する必要がある。また、OSからのペアリング情報の消去はアプリから消去するAPIは準備されていないため、ユーザーがOSのBluetooth設定から消去操作する必要がある。

## Motivation:

Bluetooth LE には Peripheral から再ペアリングを起動することができず、Central 側だけにキーが残っているケースではCentral が再ペアリングを起動するよう仕様で規定されている。しかしながら、Android 4.4.2 以前とiOS においては仕様に準拠した実装ができていないので、Security Request 受信後の接続に連続して失敗する場合 Central はキー削除のアラートをユーザに示す必要がある。

## 【修正案】

ペアリングを実施すると、セントラルであるスマートフォンのOS、アプリ(実装依存)、ペリフェラルであるデバイス(実装依存)にそれぞれペアリングに関するキーなどの情報(以降ペアリング情報)が保存される。UIを持たない機器や、小型でメモリ等のリソースが少ない機器では保存できる情報が少ない。このため、ある機器と一旦ペアリングした後他の機器とペアリングすると前の情報が消去され、次回接続時には異なるペアリング情報に書き換わるために、再度ペアリングが必要になる。しかし、Bluetooth LEの仕様で、ユーザの介在なしにOSがペアリング情報を消去することができないために再度のペアリングに失敗するため、ユーザにペアリング情報の削除を促す仕組みが必要なる。

Bluetooth Core Specにおいて、ユーザの介在なしのリンクキー削除を禁止する記載が追加されているため、Androidの実装でもプログラマブルにキーを削除する実装になっていない。このため、アプリケーションがキーを削除する実装が必要となる。【例、カシオさんの実装を記載させて頂くか？】

## 【関連する課題】

A22

## Recommendation A10:

Central 側でペアリング情報を削除した後に再ペアリングする場合、Peripheral 側では再ペアリング要求の受信によりペアリング情報を削除して再ペアリングに応答する

## Motivation:

ペアリング動作は初回も再ペアリング時も変わりがないので、再ペアリング要求の受信でペアリング情報を削除しても問題なし

## 【関連する課題】

A23

### 3.2 データ通信 (Category-B)

#### Recommendation B1:

- ① 受信アプリは接続毎にCCCDをEnable設定し、例えば血圧計側はCCCDのEnable設定を受信してから血圧測定値を送信する。
- ② ①だけの対応だとCCCD設定値を記憶するという仕様を満たしていないため、前回接続時にCCCD設定がEnableの状態でも切断され、かつ今回接続時にCCCDをEnable設定しなかった場合は、血圧計のアプリケーションレベルのタイマーのタイムアウトによる切断前に血圧測定値を送信する。

#### Motivation:

Peripheral, CentralともにCCCD設定や例えば血圧測定値送信タイミングの規定がない  
→仮に接続直後に血圧測定値を送信した場合に、サービス検索中と重なってしまいCentral側のアプリケーションで受信できないことがある

※ 接続後3秒や5秒のウェイトを設けて血圧値を送信してみたが、機種によってサービス検索に要する時間が異なるため、ウェイト時間の定義が難しい

#### 【関連する課題】

B2、B11、B12

#### Recommendation B2:

センサデバイスのメモリ送信では、基本的に送信成功の場合 Indication に対する Confirmation 受信でメモリをクリアする。ただし、データの種別によりデータを保持するケースやデータを送ったか否かを記憶しておく必要があり、詳細はプロファイル & サービスとして規定する必要がある。

#### Motivation:

ヘルスケアのデータ管理、血糖値計のデータ送受信方法等で個別に検討する必要があるが、原則としては実装依存

#### 【関連する課題】

B3、B5

#### Recommendation B3:

Notification でデータを送信する場合、すべてのデータを毎回送信し、保存データは消去しない。

#### Motivation:

データ送信時にリンクが切断すると、どこまで送信が完了したかわからなくなる。

#### 【関連する課題】

B4

Recommendation B4:

活動量計のユースケースは、将来 Physical Activity Monitor Profile (PAMP) として規定されるので参照する

Motivation:

どれくらいの頻度でデータを送信し、その程度のデータを保持し、いつ消去するかが不明。

【関連する課題】

B6

Recommendation B5:

大きめのデータ送信においては、圧縮・伸長方式を検討する。また、Object Transfer Profile & Service を利用することもできるが、Core4.2 が前提となることに加えて Classicでも動作する必要がある、HPS (HTTP Proxy Service) の実装を考慮する必要がある。

Motivation:

ある程度大きなデータを送信するケースでは、送受信時間と消費電力が懸念される。

【関連する課題】

B7

Recommendation B6:

バッテリーの保ちと接続の信頼性に基づいて、データのインターバルによりリンクをつないだままにするか、再接続するかを決める。

Motivation:

センサ機器の測定結果をリアルタイムデータ転送する場合、スマホアプリがフォアグラウンドにある時はセンサ側 Advertising Interval の2倍が必要。また、Advertising の間隔は乱数で決まるため、同期して受信できないケースを避けている。

【関連する課題】

B9

Recommendation B7:

通信終了手順は、原則として Client 側から切断し Server 側は Timeout を設定する。ただし、プロファイルによって不明な場合があり、再接続時の動作と合わせて検討が必要。

Motivation:

通信終了手順が規定されていないケースがある。

【関連する課題】

B10

**Recommendation B8:**

接続時にペアリングまで実行した後スマホアプリからサービス検索をすることにより、Android が起動する自動サービス検索を回避する。サービス検索結果が前回の結果から変更のない場合、コールバックが返らないケースもあるので、GetServices() を使用することが望ましい。

**Motivation:**

Android 5.0 でペアリング後に OS が自動サービス検索を実施し、検索中の Indication/Notification に対しては応答が返るもののスマホアプリに通知されない問題があり、Android 5.1 でスマホメーカーが独自に自動サービス検索が起動されないよう修正した経緯がある。

Android 6.0 で再び同様の問題が発生したため、スマホ側では自動サービス検索を回避する動作仕様とする必要がある。

**【関連する課題】**

B14

**3.3 アプリケーションその他 (Category-C)****Recommendation C3:**

ひとつのデバイスを複数のユーザが使用するケースでは、ペアリングをユーザ毎に複数回実行し、アプリ毎にデバイス側で Whitelist を設定して接続相手を規定する。マルチユーザ対応デバイスで複数のユーザIDが規定されているケースでは、ID 毎にペアリングする。規定済みの User Data Service (UDS) で対応可能

**Motivation:**

ユーザの識別は、アプリで対応する必要がある

**【関連する課題】**

C4

**Recommendation C4:**

シングルユーザ対応デバイスでユーザ ID が規定されていないケースでは、アプリによりマニュアルで個人を識別する

**Motivation:**

個人を特定できないので、複数ユーザが使う前提の場合はメモリデータなどを持たない方がよい。誰かが測定してしまう場合は、受信側、クラウド側でデータ削除をする必要がある。

**【関連する課題】**

C5

Recommendation C5:

複数のアプリが動作している場合のアプリとデバイスの接続、ペアリングを特定するために Advertising の機能拡張を検討する。

Motivation:

Android標準の設定アプリと専用アプリとの競合や、同じプロファイル/サービスを実現する異なるアプリを特定可能な方法が必要。

【関連する課題】

A1, A2, A10, A20, B13, C6

Recommendation C7:

HOGP/HIDS に対応するデバイスのテストにおいて、HIDS がサポートするデバイス(ロール)の情報を事前に確認する。

Motivation:

例えば、Keyboard のロールしか持たないデバイスと Mouse のロールしか持たないデバイス同士では接続してはじかれる。

【関連する課題】

C8

## 4. デバイス実装における推奨値と対応

Recommendation A9. Re-Connection & Encryption after Peripheral removes Key

【Android 5.0.2 のケース】

Central のスマホ (Master) と Peripheral のデバイス (Slave) との間でペアリングした後、別のスマホとペアリングさせ最初にペアリングしたスマホの情報を削除 (本デバイスは1台のペアリング情報しか記憶できない)。ペアリング情報が残っている最初のスマホから本デバイスと接続しようとした時のログを以下に示す。

<b>Slave</b>	<b>SMP Security equest</b>		
<b>Master</b>	<b>LL_ENC_REQ</b>	Random vector (Rand): 0xa7b84170aefd9a6a Encrypted diversifier (EDIV): 0xdcd7 Master session key identifier (SKDm): 0x9bca1664e4b9c010 Master initialization vector (IVm): 0xcf22e8db	
<b>Slave</b>	<b>LL_ENC_RSP</b>	Slave session key identifier (SKDs): 0x348fc9c741792dc8 Slave initialization vector (IVs): 0x07a6cbd3	
<b>Slave</b>	<b>LL_REJECT_IND</b>	Error Code: PIN or Key Missing	
<b>Master</b>	<b>LL_TERMINATE_IND</b>	<b>Error Code: Remote User Terminated Connection</b>	<b>ただちに切断</b>
<b>Slave</b>	<b>SMP Security Request</b>		<b>再接続</b>
<b>Master</b>	<b>Pairing request</b>	<b>Bonding Flags: Bonding</b>	<b>MITM: MITM Protection: Yes</b>
<b>Slave</b>	<b>Pairing Response</b>		
<b>Master</b>	<b>Pairing Confirm</b>	0x9188cf846938c315227693126927bed5	
<b>Slave</b>	<b>Pairing Confirm</b>	Confirm Value: 0x0b76aaba4f1dd0a981b51d68f0910c55	
	<b>Pairing Random</b>		
	<b>Paring Random</b>		
	<b>LL_ENC_REQ</b>	Random vector (Rand): 0x0000000000000000 Encrypted diversifier (EDIV): 0x0000 Master session key identifier (SKDm): 0xe4c3cc5b3f372c5f Master initialization vector (IVm): 0x20bdf9bf	
	<b>LL_ENC_RSP</b>	Slave session key identifier (SKDs): 0x4399df7063cecd28 Slave initialization vector (IVs): 0x8db37ee6	
<b>Slave</b>	<b>LL_START_ENC_REQ</b>		
<b>Master</b>	<b>LL_START_ENC_RSP</b>		
<b>Slave</b>	<b>LL_START_ENC_RSP</b>		

1回目の接続において、Peripheral (Slave) が "Key Missing" で切断しているため、2回目の接続では Central (Master) が Pairing Request を送信し、キーが更新される。

Android 4.4.4 以前では2回目の接続でも Central が Pairing Request を送信しないので、1回目と同じシーケンスが繰り返される。これは Core Spec v4.2 Vol3, Part C, p2066 以降に規定されている、

#### 10.3 AUTHENTICATION PROCEDURE

##### 10.3.1 Responding to a Service Request

##### 10.3.2 Initiating a Service Request

および、"Bluetooth Accessory Design Guidelines for Apple Products" 3.9 Pairing の規定に準拠していないためである。iOS でもこれと同様の実装であるため、対策としてスマホアプリ側で接続失敗が複数回連続する場合にユーザアラートでキー削除を促す等の処置が求められる。

なお、Qualcomm 製 Snapdragon を使用するスマホや Qualcomm 製スタックをそのまま使用しているスマホのうち、Android 6.0.0 以降の機種では Security 上の理由で2回目の接続時に Pairing Request を送信しない仕様となったため、Android 4.4.4 と同様ユーザアラートでキー削除を促す等の処理が必要になる。

#### Recommendation B8. Service Discovery

##### 【Android6.0でIndication/Notification が受信できない問題】

###### <症状>

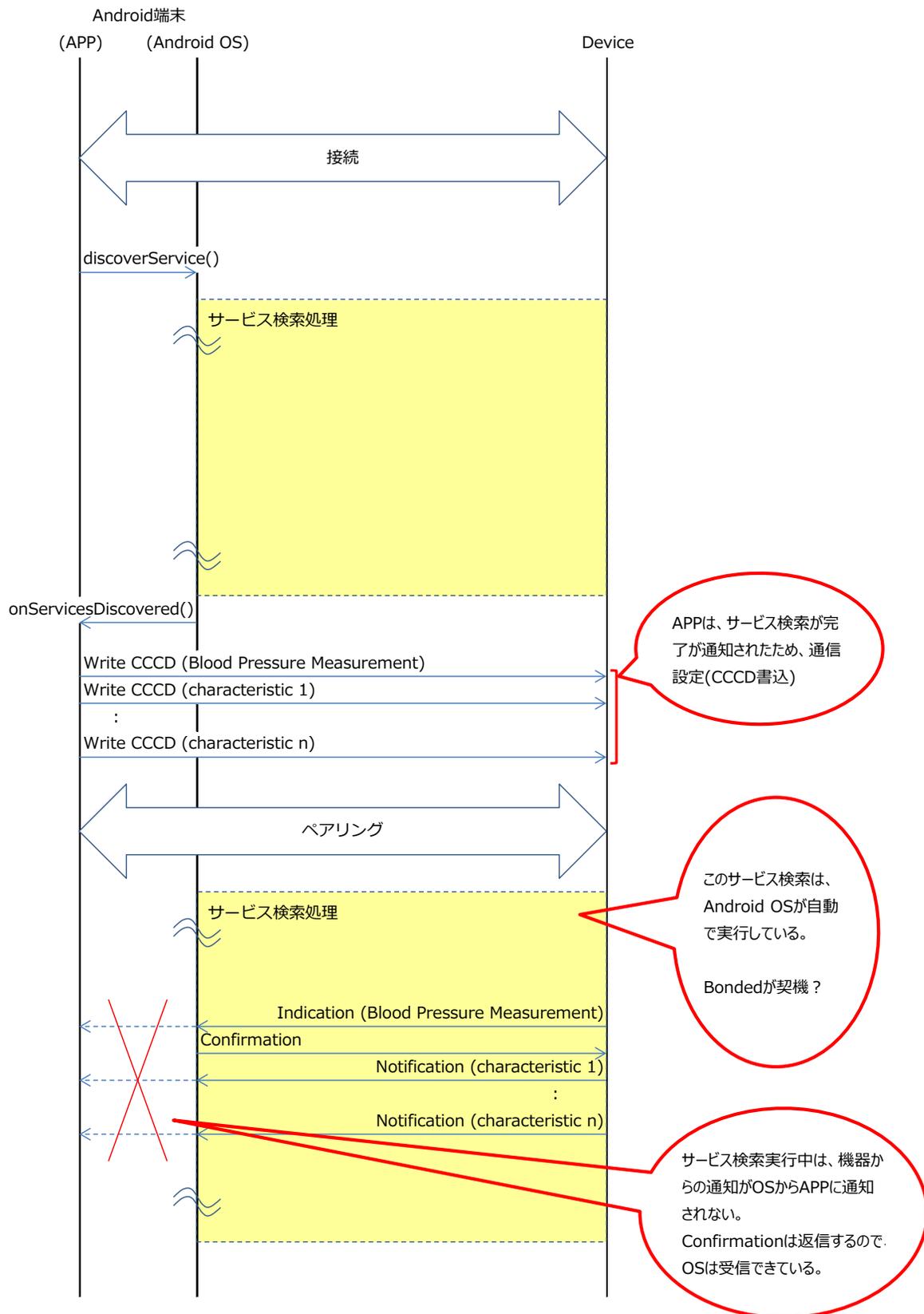
Android6.0 端末 (\*1) に対し Indication/Notification を通知しても、スマホアプリまで通知が来ないことがある。  
(\*1) Nexus6 (Android6.0.1) で確認。16年4月の互換性 WS で検証した他6.0端末、Android5 系の一部端末でも同じ症状を確認している。

###### <原因調査結果>

1. Air ログを確認した結果、Android がサービス検索処理中に Indication/Notification を通知すると、アプリに通知されない。
2. 下記シーケンスのとおり、アプリからの要求によりサービス検索を実行後、Android OS 内部でサービス検索を再実行している。この「OS が内部実行するサービス検索」と Indication/Notification の受信が重なると、アプリに通知されない。
- 3.

###### <回避策>

- ・接続とペアリングを同時に行うことでサービス探索を1回にし、受信の重なりを防ぐ
- ・サービス探索処理中の Indication/Notification でも受け取れるように、Android 端末側に手を入れる



## 5. Terms and Abbreviations

用語	説明
Bluetooth LE	Bluetooth Low Energy
BR/EDR	Basic Rate / Enhanced Data Rate
CCC	Client Characteristic Configuration
GATT	Generic Attribute Profile
HIDS	HID Service
HOGP	HID over GATT Profile
HPS	HTTP Proxy Service
IOP	Interoperability
OBEX	Object Exchange Protocol
OTP	Object Transfer Profile
OTS	Objet Transfer Service
PAMP	Physical Activity Monitor Profile
PXP	Proximity Profile
SMP	Security Manager Protocol
UDS	User Data Service
UUID	Universal Unique Identifier

表4. 用語

## 6. References