

# Bluetooth® Low Energy プロトコルスタック

## スマートフォン接続対応事例集

### 要旨

このドキュメントは、ルネサス Bluetooth Low Energy マイコン RL78/G1D を使用した Bluetooth 応用製品の開発に利用するソフトウェア「Bluetooth Low Energy プロトコルスタック」(以降、BLE ソフトウェア)を使用する際に、スマートフォンとの接続性を向上させるために行った対応事例を説明する。

### 動作確認デバイス

RL78/G1D

Android デバイス

### 関連資料

資料名	資料番号	
	和文	英文
Bluetooth Low Energy プロトコルスタック		
ユーザーズマニュアル	R01UW0095J	R01UW0095E
API リファレンスマニュアル 基本編	R01UW0088J	R01UW0088E
rBLE コマンド仕様書	R01AN1376J	R01AN1376E
クイックスタート	R01AN2767J	R01AN2767E

## 目次

1. 概要	3
1.1 本文書で使用する機器	3
1.2 事例一覧	3
2. 端末の電源 OFF→ON で再度接続が確立できない場合がある	4
2.1 概要	4
2.2 状態説明	5
2.3 改善案	6
2.3.1 端末側の改善案	6
2.3.2 Android デバイス側の改善案	8
3. 端末の電源 ON で接続が確立できない場合がある(ペアリングシーケンスの失敗)	9
3.1 概要	9
3.2 状態説明	10
3.2.1 HCI スヌープログ(正常時)	11
3.2.2 HCI スヌープログ(症状発生時)	11
3.3 改善案	12
3.4 端末プログラム実装例	13
3.4.1 デレイ用メッセージ ID の追加	13
3.4.2 LTK 応答用メッセージ処理の追加	13
4. 端末の電源 ON で接続が確立できない場合がある(Feature 交換シーケンスの失敗)	15
4.1 概要	15
4.2 状態説明	16
4.3 改善案	18
5. Android デバイスと接続が維持できない(端末側でパケット受信に失敗)	19
5.1 概要	19
5.2 状態説明	20
5.3 改善案	20
5.4 端末プログラム実装例	21
5.4.1 ライブラリ関数/変数参照の追加	21
5.4.2 インターオペラビリティ改善処理の追加	21
6. 付録	23
6.1 解析環境	23
6.1.1 Packet Sniffer ログ	24
6.1.2 Bluetooth HCI スヌープログ	24

## 1. 概要

### 1.1 本文書で使用する機器

- Android OS が使用された市販のスマートフォンやタブレット  
(以降、Android デバイスと記載。)
- Android デバイスがマスターデバイスとなる。スレーブデバイスに対して接続要求を行い、スレーブデバイスと接続するとペアリングや通信の暗号化を行う。
- RL78/G1D を使用した機器  
(以降、端末と記載。)
- 端末がスレーブデバイスとなる。端末は電源を入れると自動的にアドバタイジングを開始する。Android デバイスからの接続要求により接続すると、ペアリングや通信の暗号化を行う。

### 1.2 事例一覧

<b>1. 端末の電源 OFF→ON で再度接続が確立できない場合がある</b>
Android デバイスと端末で接続を確立している状態から、端末の電源を切った後すぐに電源を入れ、Android デバイスから接続要求を行うと、接続を確立できない場合がある。
<b>2. 端末の電源 ON で接続が確立できない場合がある(ペアリングシーケンスの失敗)</b>
端末の電源を入れ、Android デバイスから接続要求を行うと、ペアリングシーケンスが失敗し、接続を確立できない場合がある。
<b>3. 端末の電源 ON で接続が確立できない場合がある(Feature 交換シーケンスの失敗)</b>
端末の電源を入れ、Android デバイスから接続要求を行うと、Feature 交換シーケンスが失敗し、接続を確立できない場合がある。
<b>4. Android デバイスと接続が維持できない(端末側でパケット受信に失敗)</b>
Android デバイスと接続を行った際にペアリングや GATT 通信が行えず切断する。

## 2. 端末の電源 OFF→ON で再度接続が確立できない場合がある

### 2.1 概要

- 現象  
Android デバイスと接続を確立している状態から、端末の電源を切った後すぐに電源を入れ、Android デバイスから接続要求を行うと、接続を確立できない場合がある。
- 想定原因  
端末の電源を切る時に端末から切断要求が出ないため、Android デバイスは端末との通信ができなくなりタイムアウト(スーパービジョン・タイムアウト)を待つ状態となる。端末に再度電源を入れるとアドバタイジングが開始され、Android デバイスは新たな接続を確立しようとするが、当該端末に対しては、電源を切る前のスーパービジョン・タイムアウトを待つ接続が存続するため、「二重接続」という想定外の状態になることが原因と推測される。
- 対策  
端末の電源を切る前に、切断コマンドを Android デバイスに送信する。Android デバイスのアプリケーションで切断が通知されるまで同じ端末へ接続要求ができないようにする。
- 症状確認デバイス  
Android 7.1.1 を搭載している一部の Android デバイス

## 2.2 状態説明

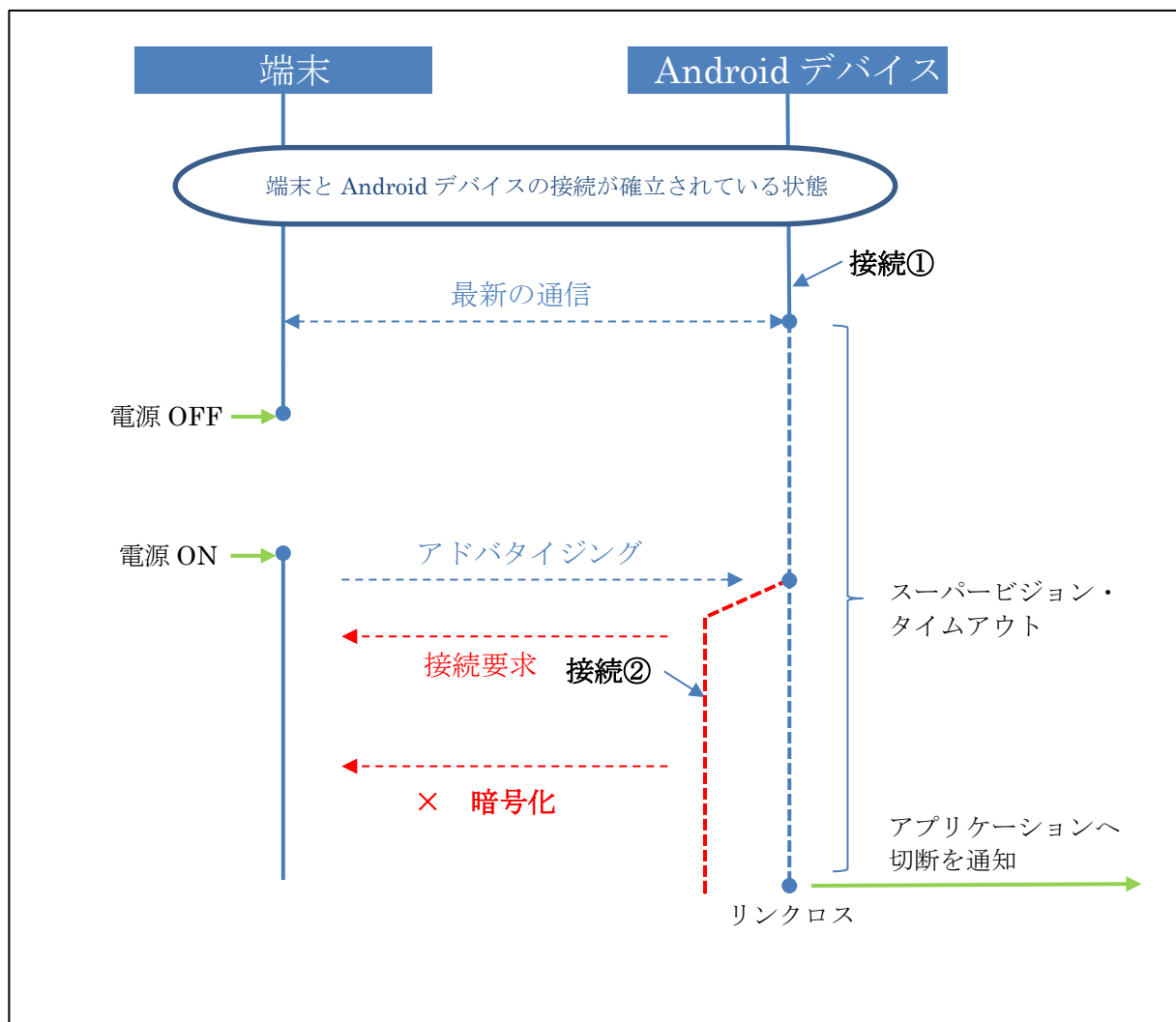


Figure 2-1 端末の電源 OFF→ON で再接続できない場合の状態説明

- 端末と Android デバイスの間でペアリングが完了し正常に接続が確立されている状態から、端末の電源を OFF した場合、Android デバイスの「接続①」はスーパービジョン・タイムアウト(Android 7.1.1 を搭載する一部の Android デバイスの場合、20 秒)待ち状態にある。
- 上記の状態にあるときに端末の電源を ON にすると、Android デバイスは端末からのアドバタイジングに対して「接続②」の接続要求を出し、接続を確立しようとするが、「接続①」の存在により端末と二重に接続している状態になり暗号化が開始されず**異常な接続状態となる**。

## 2.3 改善案

## 2.3.1 端末側の改善案

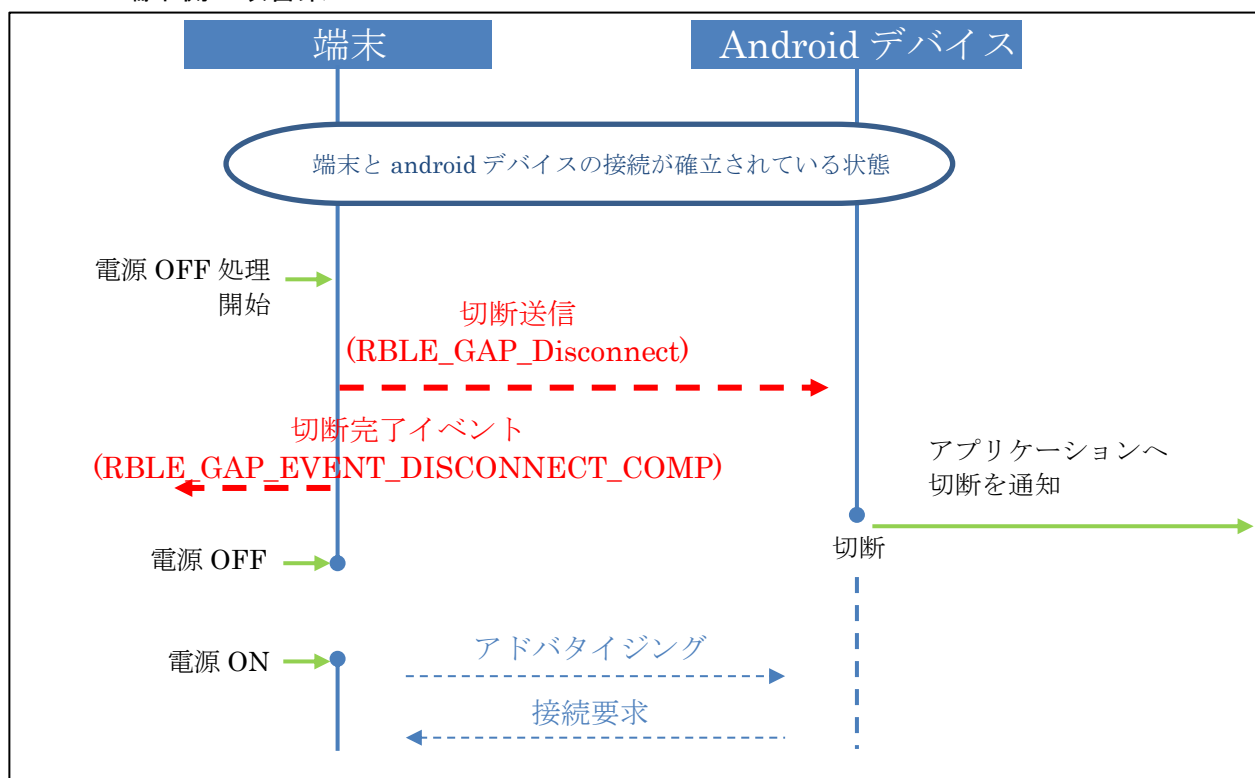


Figure 2-2 端末の電源 OFF→ON で再接続できない場合の改善案(端末側-1)

- 端末の電源を OFF する前に、切断コマンド(RBLE\_GAP\_Disconnect)を実行し、Android デバイスへ切断を送信する。端末は切断完了イベント(RBLE\_GAP\_EVENT\_DISCONNECT\_COMP)を待ち、電源を OFF する。Android デバイスとの接続を明示的に切断することにより、端末の電源 ON によるアドバタイジングに対して、正常に接続することができる。

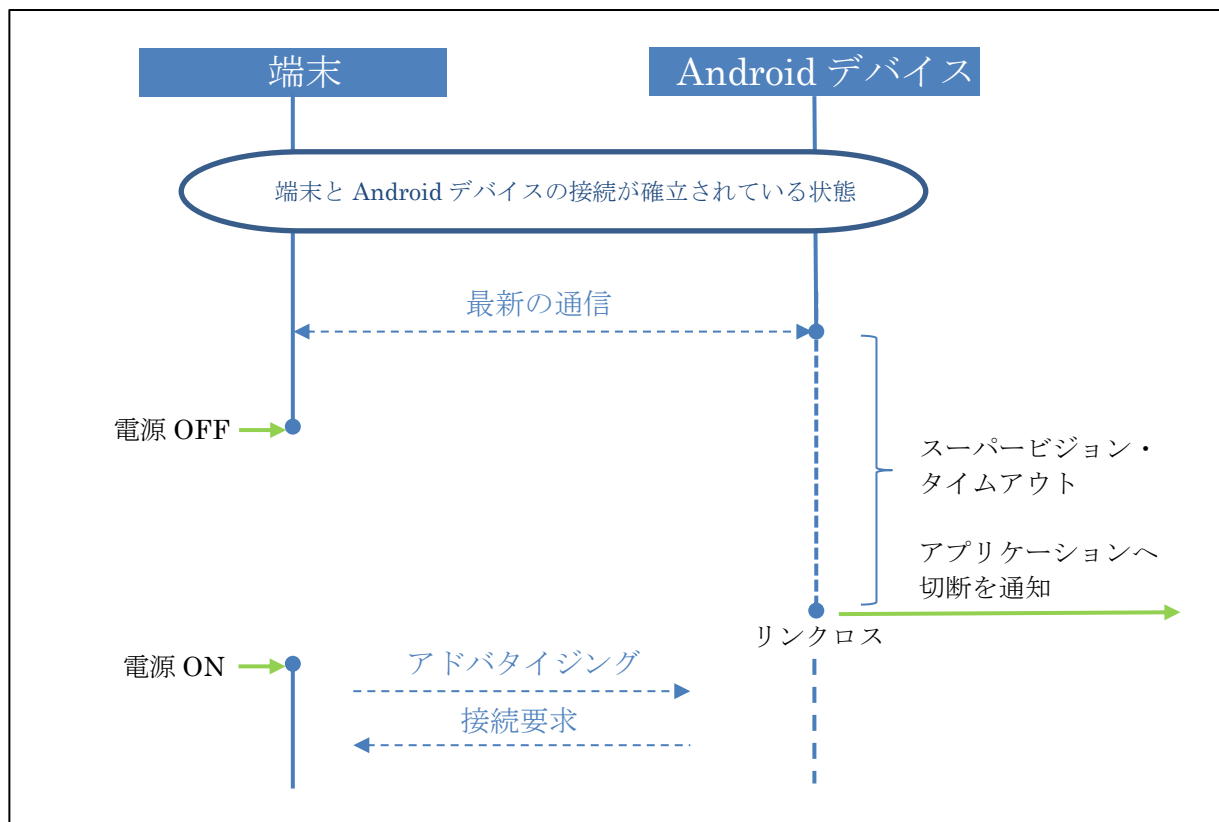


Figure 2-3 端末の電源 OFF→ON で再接続できない場合の改善案(端末側-2)

- スーパービジョン・タイムアウトの時間経過以降に、端末の電源を ON する。Android デバイスと端末との接続が切断されているため、端末からのアダプタイジングに Android デバイスが接続要求を出すことにより正常に接続が確立される。

## 2.3.2 Android デバイス側の改善案

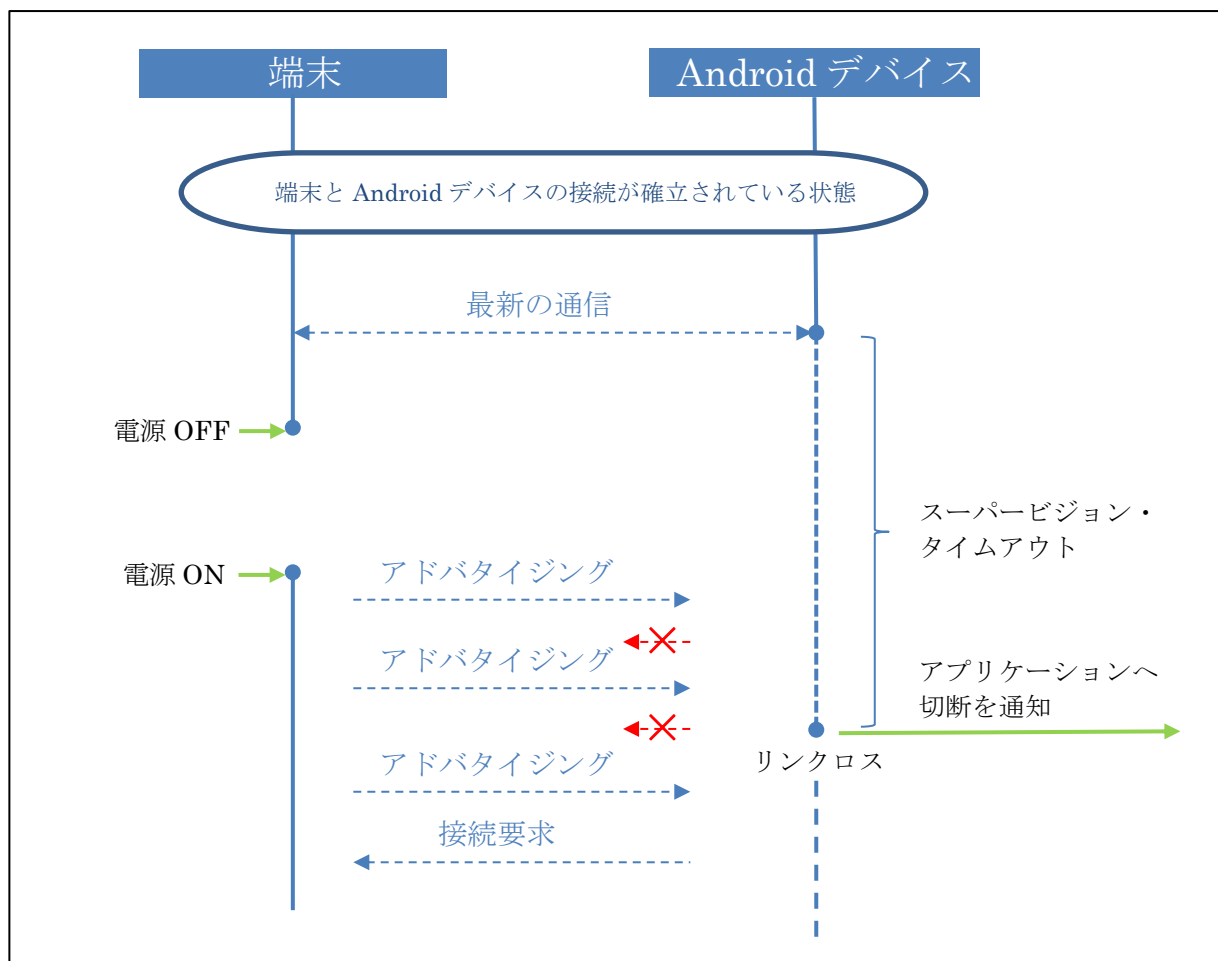


Figure 2-4 端末の電源 OFF→ON で再接続できない場合の改善案(Android デバイス側)

- 端末と接続を継続中と判断する間(スーパービジョン・タイムアウトの時間満了まで)に受信した端末からのアドバタイジングに対して Android デバイスが接続要求を出さないことにより異常な接続状態になることを回避する。  
(※ 可能であれば、接続中の端末からのアドバタイジングを受け取った時点で、リンクロスと判断し、アプリケーションからコントローラへ切断要求を出して切断状態に移行した後、端末へ接続要求を出す対策が最も望ましい。)



### 3. 端末の電源 ON で接続が確立できない場合がある(ペアリングシーケンスの失敗)

#### 3.1 概要

- 現象  
端末の電源を入れ、Android デバイスから接続要求を行うと、ペアリングシーケンスが失敗し、接続を確立できない場合がある。
- 想定原因  
Android デバイスのペアリングシーケンスにおいて、Android デバイスのコントローラ層から Encryption Change イベントよりも先に Encryption Information と Master Identification をホスト層に通知してしまうことにより、ペアリングシーケンスの順番に齟齬が生じて処理が停止してしまう。
- 対策  
Android デバイスで Encryption Change イベントの後に Encryption Information と Master Identification が通知されるように、RL78/G1D のプログラムで LTK の要求に応答する関数 `RBLE_SM_Ltk_Req_Resp` の実行を遅延させる。  
  
**※遅延時間は、「接続インターバル×2」を推奨。**
- 症状確認デバイス  
Android 7.0 を搭載している一部の Android デバイス

## 3.2 状態説明

- 症状発生状態

RL78/G1D とスマートフォンの接続で、接続できない現象が発生し、「Bluetooth への接続ができません」という内容のエラーメッセージがスマートフォンのアプリケーションで表示される。

- 解析方法

症状の発生する Android デバイスと端末を使用し、Android デバイス内の通信ログである HCI スヌープログを解析。

- 解析結果

Android デバイスで症状発生状態のエラーメッセージが表示されるとき、端末と Android デバイスは接続後のペアリングシーケンスが途中で停止している状態になっている。端末が Master Identification を送信した後、Android デバイスが Identify Information を送信しないためペアリングシーケンスが完了せず、端末と Android デバイスが不完全な接続を維持した状態のままとなっている。Android デバイスのコントローラ層から Encryption Change イベントよりも先に Encryption Information と Master Identification をホスト層に通知してしまうことにより、ペアリングシーケンスの順番に齟齬が生じて処理が停止してしまうためと考えられる。

## 3.2.1 HCI スヌープログ(正常時)

No.	Time	Source	Destination	Protocol	Length	Info
145	28.886628	host	controller	HCI_CMD	32	Sent LE Start Encryption
146	28.892502	controller	host	HCI_EVT	7	Rcvd Command Status (LE Start Encryption)
147	29.177797	controller	host	HCI_EVT	7	Rcvd Encryption Change
148	29.178117	RenesasE_00:7f...	localhost ()	SMP	26	Rcvd Encryption Information
149	29.178819	RenesasE_00:7f...	localhost ()	SMP	20	Rcvd Master Identification
150	29.180198	localhost ()	RenesasE_00:7f...	SMP	26	Sent Identity Information
151	29.180302	host	controller	HCI_CMD	43	Sent LE Add Device to Resolving List
152	29.180370	localhost ()	RenesasE_00:7f...	SMP	17	Sent Identity Address Information
153	29.180429	localhost ()	RenesasE_00:7f...	ATT	16	Sent Read By Group Type Request, GATT Pri...

[Source と Destination]  
 host : スマートフォンの上位ドライバ  
 localhost : スマートフォンの上位ドライバ  
 controller : スマートフォンの下位ドライバ  
 RenesasE\_00 : RL78/G1D

Figure 3-1 HCI スヌープログ(正常時)

- ① Android デバイス内でコントローラ層から LE Start Encryption イベントの直後に Encryption Change イベントが発生すると、ホスト層から Identify Information が送信されている。
- ② 端末から送信された Encryption Information と Master Identification が Android デバイス内のコントローラ層から通知され、ホスト層から Identify Information を送信する。その後、ペアリングシーケンスが完了する。

## 3.2.2 HCI スヌープログ(症状発生時)

No.	Time	Source	Destination	Protocol	Length	Info
168	29.038978	host	controller	HCI_CMD	32	Sent LE Start Encryption
169	29.049877	controller	host	HCI_EVT	7	Rcvd Command Status (LE Start Encryption)
170	29.331540	RenesasE_00:7f...	localhost ()	SMP	26	Rcvd Encryption Information
171	29.331783	RenesasE_00:7f...	localhost ()	SMP	20	Rcvd Master Identification
172	29.331918	controller	host	HCI_EVT	7	Rcvd Encryption Change
173	55.516396	remote ()	localhost ()	L2CAP	488	Rcvd Connection oriented channel

[Source と Destination]  
 host : スマートフォンの上位ドライバ  
 localhost : スマートフォンの上位ドライバ  
 controller : スマートフォンの下位ドライバ  
 RenesasE\_00 : RL78/G1D

Figure 3-2 HCI スヌープログ(症状発生時)

- ① 端末から送信された Encryption Information と Master Identification が Android デバイスのコントローラ層から Encryption Change イベントの前に通知されている。この状態が発生すると、Android デバイスのホスト層が Identify Information を送信しないためペアリングシーケンスは完了していない、不完全な接続が維持されている状態となる。

※ Android デバイス内でコントローラ層からの LE Start Encryption イベントの後、Encryption Change イベントの発生が遅れると、ホスト層から Identify Information が送信されない。

3.3 改善案

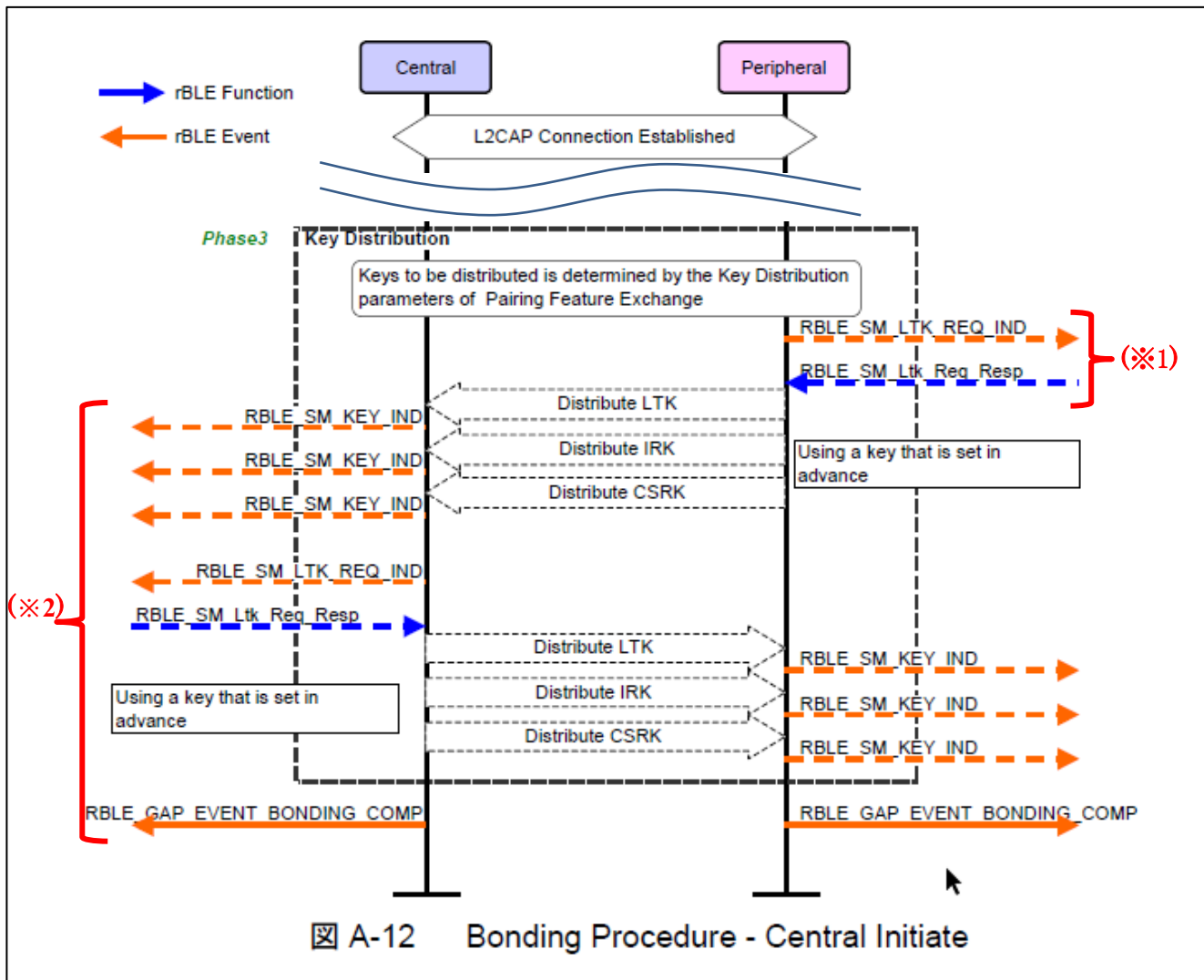


図 A-12 Bonding Procedure - Central Initiate

Figure 3-3 端末の電源 ON で接続できない場合(ペアリングシーケンスの失敗)の改善案

- 端末プログラムのペアリングシーケンス処理で、RBLE\_SM\_LTK\_REQ\_IND イベントが発生してから RBLE\_SM\_Ltk\_Req\_Resp 関数を呼び出す間にウェイトを挿入(Figure 3-3 の **(\*1)**箇所)することで Encryption Information と Master Identification の送信を意図的に遅延させることができる。これにより、Android デバイス内で、HCI イベントの Encryption Change が発生した後に、端末からの Encryption Information と Master Identification を受け取ることができ、正常なペアリングシーケンスが実行できる。
- Central 側であるスマートフォン側は、Android の場合は OS が自動的に Figure 3-3 の **(\*2)**箇所の処理を行うため、アプリケーション側で行う処理はない。

### 3.4 端末プログラム実装例

カーネルタイマを用いて RBLE\_SM\_Ltk\_Req\_Resp 関数の呼び出しを遅延させるための実装例を以下に示す。

#### 3.4.1 ディレイ用メッセージ ID の追加

端末プログラムで使用しているカーネルのメッセージ ID の enum 列挙に、ディレイ用メッセージ ID を追加する。

注： 弊社サンプルプログラムの場合、r\_ble\_sample\_app\_peripheral.h に定義がある。

```
typedef enum {
    APP_MSG_BOOTUP = KE_FIRST_MSG(APP_TASK_ID) + 1,
    APP_MSG_RESET_COMP,
    APP_MSG_SECLIB_SET_PARAM_COMP,
    APP_MSG_CONNECTED,
    APP_MSG_SECLIB_CHK_ADDR_COMP,
    APP_MSG_SECLIB_PASSKEY_IND,
    APP_MSG_SECLIB_ENC_COMP,
    APP_MSG_DISCONNECTED,
    APP_MSG_PROFILE_ENABLED,
    APP_MSG_PROFILE_DISABLED,
    APP_MSG_TIMER_EXPIRED,
    APP_MSG_LTK_REQ_DELAY, // ①ディレイ用メッセージ ID
} APP_MSG_ID;
```

Figure 3-4 ディレイ用メッセージ ID の追加

#### 3.4.2 LTK 応答用メッセージ処理の追加

RBLE\_SM\_LTK\_REQ\_IND イベントが発生した際にカーネルタイマをセットしてウェイト時間を設けて LTK 応答を行うためのメッセージ関数処理を追加する。

注： 弊社サンプルプログラムの場合、r\_ble\_sample\_app\_peripheral.c にメッセージ関数処理を追加する。

```
// ②LTK 応答用のメッセージ関数のプロトタイプ宣言
static int_t app_ltk_req_delay(ke_msg_id_t const msgid, void const *param,
    ke_task_id_t const dest_id, ke_task_id_t const src_id);
```

Figure 3-5 LTK 応答用メッセージ関数のプロトタイプ宣言の追加

```
const struct ke_msg_handler app_connect_handler[] = {
    { APP_MSG_CONNECTED, (ke_msg_func_t)app_profile_enable },
    /* { APP_MSG_PROFILE_ENABLED, (ke_msg_func_t)NULL }, */
    { APP_MSG_TIMER_EXPIRED, (ke_msg_func_t)app_timer_expired },
    // ③タイマ時間経過後のメッセージ関数を登録
    { APP_MSG_LTK_REQ_DELAY, (ke_msg_func_t)app_ltk_req_delay },
};
```

Figure 3-6 接続中のメッセージハンドラへタイマ時間経過後のメッセージ関数を追加

```
/* ##### SM Event Handler ##### */
void app_sm_callback(RBLE_SM_EVENT *event)
{
    switch (event->type) {
        case RBLE_SM_LTK_REQ_IND:
            req_result = event->param.ltk_req;

            // ④ディレイ時間ウェイトのためのカーネルタイマを設定 (単位時間は10msec)
            // -> LTK(Long Term Key)の応答を遅延させる
            ke_timer_set(APP_MSG_LTK_REQ_DELAY, APP_TASK_ID, 50); // 500msec ウェイト
            break;

        default:
            break;
    }
}
```

Figure 3-7 RBLE\_SM\_LTK\_REQ\_IND イベント発生時にカーネルタイマ設定処理を追加

```
// ⑤LTK 応答用のメッセージ関数を追加実装 -> ディレイ時間の経過後に実行される
static int_t app_ltk_req_delay(ke_msg_id_t const msgid, void const *param,
                              ke_task_id_t const dest_id, ke_task_id_t const
src_id)
{
    /* Generate LTK/EDIV/NB. */
    seclib_generate_key(&ld_ltk.val);
    seclib_generate_nb(&ld_ltk.nb);
    ld_ltk.ediv = SecLib_Rand();
    ld_ltk.valid = SECDB_VALID_KEY;

    /* LE Long Term Key Request Reply */
    RBLE_SM_Ltk_Req_Resp(req_result.idx, RBLE_OK,
                        RBLE_SMP_KSEC_NONE,
                        ld_ltk.ediv,
                        &ld_ltk.nb,
                        &ld_ltk.val);

    return KE_MSG_CONSUMED;
}
```

Figure 3-8 LTK 応答用メッセージ関数の追加

## 4. 端末の電源 ON で接続が確立できない場合がある(Feature 交換シーケンスの失敗)

### 4.1 概要

- 現象  
端末の電源を入れ、Android デバイスから接続要求を行うと、Feature 交換シーケンスが失敗し、接続を確立できない場合がある。
- 想定原因  
Android デバイスのホスト層から実行される接続要求コマンド HCI\_LE\_Create\_Connection と、接続完了イベント LE Connection Complete の間に、他のコマンドやイベントが実行されると、リモートデバイスがサポートしている機能をリードするコマンド HCI\_LE\_Read\_Remote\_Used\_Features 以降の接続を確立するためのシーケンスが実行されない。
- 対策  
Android デバイスアプリケーションで接続要求を実行した後は、接続完了イベント LE Connection Complete を受け取るまで他の処理を入れない。
- 症状確認デバイス  
Android 5.0.1 を搭載している一部の Android デバイス

## 4.2 状態説明

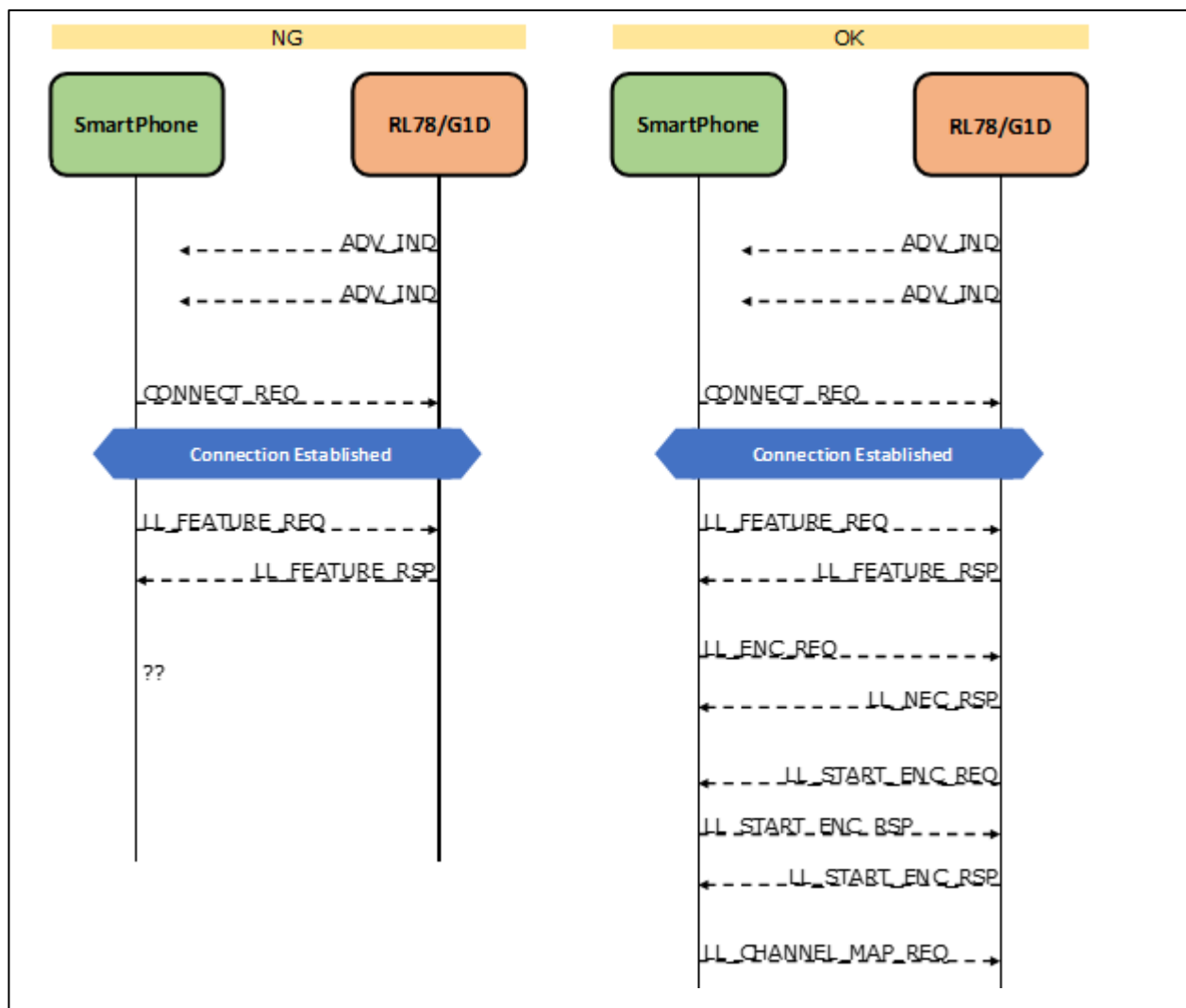


Figure 4-1 端末の電源 ON で接続できない場合 (Feature 交換シーケンスの失敗) の状態説明

- Bluetooth Packet Sniffer ログの解析結果
  - NG の場合、Android デバイスが LL\_FEATURE\_REQ を送信し、端末が LL\_FEATURE\_RSP を送信した後、Android デバイスが応答しなくなる。NG の場合でも、Android デバイスと端末は接続されている。接続を維持したまま、LL\_FEATURE のシーケンスから進まない状態となっている。
  - OK の場合、Android デバイスが LL\_FEATURE\_REQ を送信し、端末が LL\_FEATURE\_RSP を送信した後、シーケンスが正常に処理されている。



Type	Opcode Command	Event
Command	HCI_LE_Create_Connection	
Event	HCI_LE_Create_Connection	Command Status
Command	HCI_LE_Set_Advertising_Parameters	
Event	HCI_LE_Set_Advertising_Parameters	Command Complete
Command	Write_Scan_Enable	
Event	Write_Scan_Enable	Command Complete
Command	Write_Scan_Enable	
Event	Write_Scan_Enable	Command Complete
Command	Write_Extended_Inquiry_Response	
Event	Write_Extended_Inquiry_Response	Command Complete
Command	Write_Extended_Inquiry_Response	
Event	Write_Extended_Inquiry_Response	Command Complete
Command	Write_Extended_Inquiry_Response	
Event	Write_Extended_Inquiry_Response	Command Complete
Command	Write_Extended_Inquiry_Response	
Event	Write_Extended_Inquiry_Response	Command Complete
Command	Write_Extended_Inquiry_Response	
Event	Write_Extended_Inquiry_Response	Command Complete
Command	Write_Class_of_Device	
Event	Write_Class_of_Device	Command Complete
Command	HCI_LE_Set_Advertising_Data	
Event	HCI_LE_Set_Advertising_Data	Command Complete
Command	HCI_LE_Set_Advertising_Parameters	
Event	HCI_LE_Set_Advertising_Parameters	Command Complete
Event		LE Connection Complete
Command	HCI_LE_Read_Remote_Used_Features	
Event	HCI_LE_Read_Remote_Used_Features	Command Status
Event		LE Read Remote Used Features Complete
Command	Change_Local_Name	
Event	Change_Local_Name	Command Complete
Command	Write_Extended_Inquiry_Response	

252ms

Figure 4-2 HCI スヌープログ(症状発生時)

Type	Opcode Command	Event
Command	HCI_LE_Create_Connection	
Event	HCI_LE_Create_Connection	Command Status
Event		LE Connection Complete
Command	HCI_LE_Read_Remote_Used_Features	
Event	HCI_LE_Read_Remote_Used_Features	Command Status
Event		LE Read Remote Used Features Complete
Command	HCI_LE_Start_Encryption	
Event	HCI_LE_Start_Encryption	Command Status
Event		Encryption Change

35ms

Figure 4-3 HCI スヌープログ(正常時)

- HCI スヌープログの解析結果

- 正常時の場合、HCI\_LE\_Create\_Connection~LE Connection Complete まで連続で実行される。その後、HCI\_LE\_Read\_Remote\_Used\_Features と HCI\_LE\_Start\_Encryption のコマンドが実行される。
- 症状発生時の場合、HCI\_LE\_Create\_Connection~LE Connection Complete の間に別のコマンドやイベントが実行される。LE Connection Complete の後、HCI\_LE\_Read\_Remote\_Used\_Features コマンドは実行されるが、HCI\_LE\_Start\_Encryption コマンドが実行されない。

### 4.3 改善案

Android デバイスアプリケーションで接続要求を実行した後は、接続完了イベント LE Connection Complete を受け取るまで他の処理を入れない。

## 5. Android デバイスと接続が維持できない(端末側でパケット受信に失敗)

### 5.1 概要

- 現象  
Android デバイスと接続を行った際にペアリングや GATT 通信が行えず切断する。
- 想定原因  
一部の Android デバイスの送信パケットに対して、端末がパケットの受信に失敗してしまい接続イベントで応答できずに切断してしまうことが原因と推測される。
- 対策  
該当パケットの受信が行えるように端末側のプログラム修正を行う。
- 症状確認デバイス  
Android11 以降を搭載している一部の Android デバイス

## 5.2 状態説明

- 症状発生状態

端末と Android デバイスの接続で、「このデバイスを使用するにはアプリが必要です」という内容のエラーメッセージがスマートフォンのアプリケーションで表示される。

注：使用する Android デバイスによってはエラーメッセージが異なる可能性がある。

- 解析方法

症状の発生する Android デバイスと端末を使用し、端末側はオンチップデバッグエミュレータ (E2 エミュレータ Lite 等) を使用してデバッグ接続した状態で、RBLE\_GAP\_Reset API で登録した gap\_call\_back 関数に通知される LE リンク切断完了イベント (RBLE\_GAP\_EVENT\_DISCONNECT\_COMP) 箇所にブレークを設定し、LE リンク切断完了イベント発生時にイベントパラメータの切断理由(reason)を確認する。

注：コンパイラの最適化レベルによってはイベントパラメータを変数名でウォッチ出来ない場合がある。その場合、デバッグ用に uint8\_t 型の大域変数を追加し、その大域変数に切断理由(reason)値を代入し、大域変数をデバッグ上のウォッチウインドウで確認する。

- 解析結果

症状発生の操作を繰り返し行い、端末側 LE リンク切断完了イベントの切断理由が RBLE\_CONN\_FAILED\_TO\_BE\_ES(0x3E) または RBLE\_CON\_TIMEOUT(0x08) である場合、端末が Android デバイスのパケット受信に失敗し切断に至っていると考えられる。

## 5.3 改善案

症状の発生する Android デバイスのパケットを受信できるように、端末側のメイン関数内にインターオペラビリティ改善のための処理を追加する。

## 5.4 端末プログラム実装例

インターオペラビリティ改善のための実装例を以下に示す。

### 5.4.1 ライブラリ関数/変数参照の追加

端末プログラムで使用するライブラリ関数/変数を使用するためのコード[1]を追加する。

注：弊社サンプルプログラムの場合、rf.h にコードを追加する。

```
<Project_Source¥renesas¥src¥driver¥rf¥rf.h>
:
#ifdef RF_H_
#define RF_H_
:

void rf_init(const uint16_t rf_flg);
// [1]ライブラリ関数/変数の外部参照宣言を追加
void rf_renesas_reg_wr(uint16_t addr, uint16_t value);
extern bool sleep_data_save;

/// @}

#endif // RF_H_
```

Figure 5-1 ライブラリ関数/変数参照の追加

### 5.4.2 インターオペラビリティ改善処理の追加

端末プログラムの main 関数および arch\_main\_ent 関数にインターオペラビリティ改善のための処理コード [2]~[5]を追加する。

注：弊社サンプルプログラムの場合、main.c と arch\_main.c にコードを追加する。

```
<Project_Source¥renesas¥src¥arch¥r178¥main.c>
:
_MAINCODE void main( void )
{
:
#ifdef USE_FW_UPDATE_PROFILE
/* during FW update? */
if( true == check_fw_update() ) {
:
// Disable the BLE core
rwble_disable();

// Initialize RF
rf_init(CFG_RF_INIT);

// [2]インターオペラビリティ改善処理を追加
rf_renesas_reg_wr(0x11A4,0x0B3A);
rf_renesas_reg_wr(0x11A6,0x3A3A);

// Initialize BLE stack
rwble_init(&public_addr, CFG_SCA);
:
}
else
#endif
{
/* call arch main */
arch_main();
}
}
```

Figure 5-2 main.c のコード追加

```

<Project_Source\renesas\src\arch\rl178\arch_main.c>
:
void arch_main_ent(void)
{
    struct bd_addr public_addr;    /* Public Device Address */
    bool app_reg_set = false; // [3] インターオペラビリティ改善設定フラグを追加

    /* Disable parity error resets */
    RPECTL = 0x80;
:
:
    // And loop forever
    for (;;)
    {
        // [4] インターオペラビリティ改善設定フラグの判定および改善処理の追加
        if( (sleep_data_save == false) && (app_reg_set == false) )
        {
            rf_renesas_reg_wr(0x11A4,0x0B3A);
            rf_renesas_reg_wr(0x11A6,0x3A3A);
            app_reg_set = true;
        }

        //LED activity
        led_blink();

        // schedule the BLE stack
        rwble_schedule();

        // Checks for sleep have to be done with interrupt disabled
        GLOBAL_INT_DISABLE();
        // Check if the processor clock can be gated
        if ((uint16_t)rwble_sleep() != false)
        {
            // check CPU can sleep
            if ((uint16_t)sleep_check_enable() != false)
            {
                #ifndef CONFIG_EMBEDDED
                /* Before CPU enters stop mode, this function must be called */
                if ((uint16_t)wakeup_ready() != false)
                #endif // #ifndef CONFIG_EMBEDDED
                {
                    // Wait for interrupt
                    WFI();

                    #ifndef CONFIG_EMBEDDED
                    /* After CPU is released stop mode, this function must be called immediately */
                    wakeup_finish();
                    #endif // #ifndef CONFIG_EMBEDDED
                }
            }
        }
        // [5] インターオペラビリティ改善設定フラグ更新処理を追加
        if(sleep_data_save != false)
        {
            app_reg_set = false;
        }
        // Checks for sleep have to be done with interrupt disabled
        GLOBAL_INT_RESTORE();

        sleep_load_data();
    }
}

```

Figure 5-3 arch\_main.c のコード追加

## 6. 付録

解析に利用できる環境を以下に示す。

### 6.1 解析環境

解析環境として、以下の二種類がある。

- 端末機器(RL78/G1D)と Android デバイスの通信を Bluetooth Packet Sniffer でキャプチャし、パケットログで解析
- Android デバイスの BLE 動作状況を Bluetooth HCI スヌープログで解析



Figure 6-1 解析環境

## 6.1.1 Packet Sniffer ログ

機器間の Air 上の通信をキャプチャし、ログを解析する。

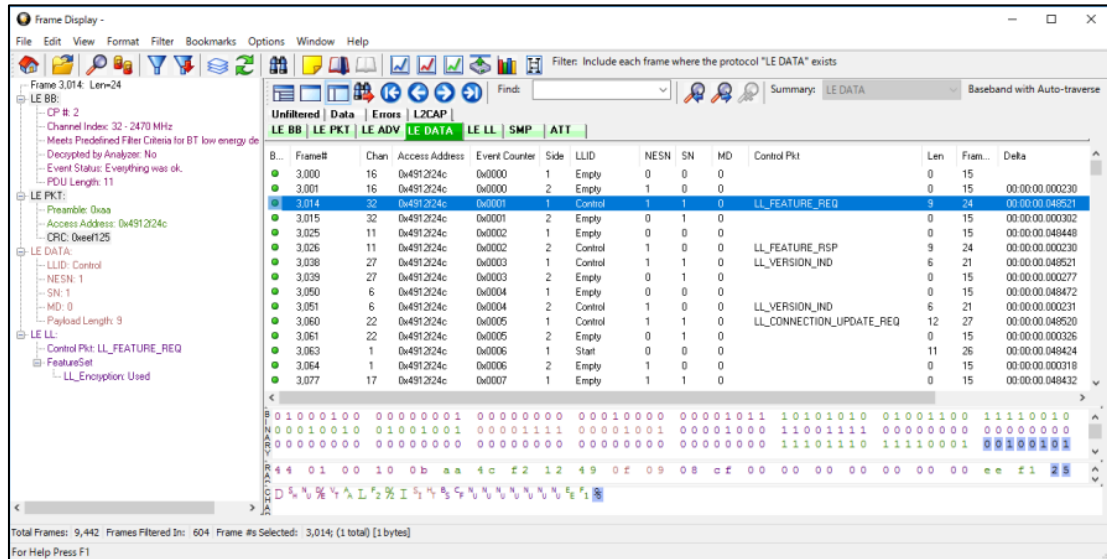


Figure 6-2 Packet Sniffer ログ

## 6.1.2 Bluetooth HCI スヌープログ

Android スマートフォン内の BLE HCI 通信を記録し、ログを解析する。

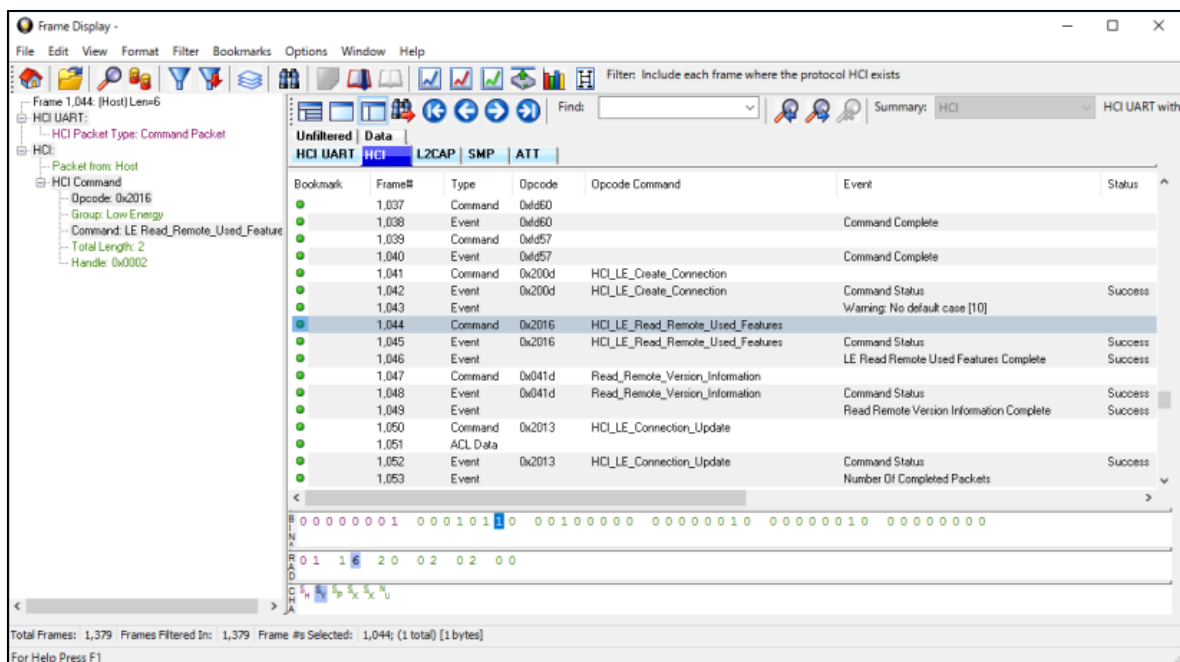


Figure 6-3 Bluetooth HCI スヌープログ



Android の Bluetooth HCI スヌープログの記録方法は以下の通り。

1. Android スマートフォンの「設定」から「開発者向けオプション」を有効化し、「Bluetooth HCI スヌープログ」の設定を ON にする。
2. スマートフォンの Bluetooth を有効にすると、ログの記録が開始される。

ログファイル名は、"btsnoop\_hci.log"

※ファイルの保存先は機種に依存する。詳細はスマートフォンのマニュアルを参照。

3. ログを記録し続けたためファイルサイズが大きくなり見づらい場合は、記録を開始する前に、一旦スマートフォンの Bluetooth を無効にし、「開発者向けオプション」と「Bluetooth HCI スヌープログ」も無効にしたあと、再度有効にしてログの記録をリセットすることが可能。

Bluetooth HCI スヌープログを取得したら、ビューアで開くことができる。

ビューア参考)

- BPA software

<http://fte.com/support/CPAS-download.aspx?demo=BPA%20500&iid=1U>

インストール後、Capture File Viewer を使用する。

- Wireshark

<https://www.wireshark.org/>

ビューアで開いたら、次に示す手順で解析をすることができる。

1. "disconnection"や"response timeout"等のキーワードで検索する。見つかった場合は、エラーの発生箇所からさかのぼりエラー発生原因を解析する。
2. エラー発生箇所がキーワードで見つからない場合は、"create\_connection"のキーワードで接続が確立した箇所を検索する。Command と Event を一つずつチェックしていき、エラー発生箇所と原因を解析する。

53	Command	HCI_LE_Create_Connection				00:00:00.024035	2017/09/26 9:26:58.951985
54	Event	HCI_LE_Create_Connection	Command Status	Success		00:00:00.005141	2017/09/26 9:26:58.957126
55	Event		Warning: No default case [10]			00:00:00.550315	2017/09/26 9:26:59.507441
56	Command	HCI_LE_Read_Remote_Used_Features				00:00:00.000640	2017/09/26 9:26:59.508081
57	Event	HCI_LE_Read_Remote_Used_Features	Command Status	Success		00:00:00.002543	2017/09/26 9:26:59.510624
58	Event		LE Read Remote Used Features Complete	Success		00:00:00.109168	2017/09/26 9:26:59.619792
59	Command	Read_Remote_Version_Information				00:00:00.000234	2017/09/26 9:26:59.620026
60	Event	Read_Remote_Version_Information	Command Status	Success		00:00:00.004088	2017/09/26 9:26:59.624114
61	Event		Read Remote Version Information Complete	Success		00:00:00.093136	2017/09/26 9:26:59.717250
:	:	:	:	:	:	:	:
173	ACL Data					00:00:00.009306	2017/09/26 9:27:07.672764
174	Command	HCI_LE_Connection_Update				00:00:00.006043	2017/09/26 9:27:07.678807
175	Event	HCI_LE_Connection_Update	Command Status	Success		00:00:00.006914	2017/09/26 9:27:07.685721
176	Event		LE Connection Update Complete	LMP Response Timeout		00:00:00.550251	2017/09/26 9:27:08.235972
177	Event		Disconnection Complete	Success		00:00:00.000642	2017/09/26 9:27:08.236614
178	Command					00:02:00.298772	2017/09/26 9:29:08.535386

Figure 6-4 ビューア表示例

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2018/04/27	-	初版発行
1.10	2022/12/23	19	「5. Android デバイスと接続が維持できない(端末側でパケット受信に失敗)」を追加
1.20	2023/04/07	21	iOS デバイスに対するインターオペラビリティ担保のために「5.4.2 インターオペラビリティ改善処理の追加」の処理コード[2]および[4]の設定値を更新

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

### 2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

### 4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

### 5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

### 7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違えば、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違えば製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

[www.renesas.com](http://www.renesas.com)

## お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

[www.renesas.com/contact/](http://www.renesas.com/contact/)

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。