

PC-AOA-01

サンプルプログラム解説

第 1.1 版 2023 年 10 月 02 日

目 次

1. 概要	1
1. 1 概要	1
1. 2 必要機器一覧	1
1. 3 開発環境について	2
1. 3. 1 Android 用サンプルアプリケーション	2
1. 3. 2 RX63N 用サンプルプログラム	2
1. 4 PC-AOA-01 使用時の通信設定	3
1. 4. 1 通信設定概要	3
1. 4. 2 通信設定パッケージ	4
1. 5 通信に関する基本設定	5
1. 6 RX63N 用サンプルプログラムの移植について	5
1. 6. 1 弊社 CPU ボードへの移植	5
1. 6. 2 その他の機器への移植	5
2. サンプルプログラム動作説明	6
2. 1 動作概要	6
2. 2 サンプルプログラム動作モード設定	7
2. 3 動作手順	9
2. 4 画面構成要素説明	13
3. Android サンプルアプリケーション概要	15
3. 1 動作概要	15
3. 2 フォルダ構成	15
3. 3 ファイル構成	17
3. 4 Eclipse を用いたインストール方法	19
4. RX63N 用サンプルプログラム概要	20
4. 1 動作概要	20
4. 2 フォルダ構成	21
4. 3 ファイル構成	22
4. 3. 1 CubeSuite+用サンプルプログラム	22
4. 3. 2 HEW 用サンプルプログラム	23

5. サンプルプログラムにおけるシリアル通信構成.....	24
5. 1 通信パッケージ構成.....	24
5. 1. 1 基本パッケージ構成.....	24
5. 1. 2 パラメータ概要.....	24
5. 1. 3 各種コマンド説明.....	24
5. 2 Android・RX63N 間の通信手順.....	27
5. 2. 1 アプリケーション開始時.....	27
5. 2. 2 LED 制御パッケージの送受信.....	27
5. 2. 3 アプリケーション終了時.....	28

1. 概要

1. 1 概要

本アプリケーションノートでは、PC-A0A-01 に付属するサンプルプログラムについて解説します。

PC-A0A-01 には、Android 上で動作するサンプルアプリケーション、及び弊社製 CPU ボード AP-RX63N-0A (以下 RX63N) 上で動作するサンプルプログラムが付属しています。

サンプルプログラム	動作内容
Android 用サンプルアプリケーション	<ul style="list-style-type: none">Android シリアル通信実装例Android USB アクセサリ通信設定実装例
RX63N 用サンプルプログラム	<ul style="list-style-type: none">RX63N シリアル通信実装例RX63N LED 制御

各サンプルの詳細な動作内容に関しては、後述の「2. サンプルプログラム動作説明」を参照してください。

CPU ボード RX63N の詳細に関しては弊社 HP <https://www.apnet.co.jp> からマニュアルがダウンロードできますので、そちらをご覧ください。

1. 2 必要機器一覧

本アプリケーションノートにおいて、必要な機器は以下のとおりです。

機器名	概要
PC-A0A-01	Android Open Accessory シリアル変換ユニット
Android 端末	Android2.3.4 以上の USB アクセサリ対応端末のみサンプルアプリケーション動作対象 ※1
AP-RX63N-0A	サンプルプログラム実行用 (別売)
電源	RX63N 電源供給用。DC+5V
USB ケーブル	データ通信できるもの

※1：本サンプルアプリケーションの通信において必要な Android Open Accessory は、Android2.3.4 より提供された機能となります。

ただし、Android2.3.4 以上のバージョンであっても、Android Open Accessory が提供されていない端末では使用できません。

1. 3 開発環境について

1. 3. 1 Android 用サンプルアプリケーション

Android 用サンプルアプリケーションは統合開発環境 Eclipse を用いて開発しております。
サンプルアプリケーション作成時の開発環境のバージョンは以下となります。

開発環境	バージョン
Eclipse	Juno Service Release 2

1. 3. 2 RX63N 用サンプルプログラム

RX63N 用サンプルプログラムは統合開発環境 CubeSuite+、及び総合開発環境 High-performance Embedded Workshop (本書では以下 HEW と記述します) を用いて開発しております。

サンプルプログラム作成時の開発環境とコンパイラのバージョンはそれぞれ以下となります。

また、サンプルプログラムはルネサスエレクトロニクス社製のオンチップデバッグエミュレータ「E1 エミュレータ」の使用を前提としております。

フォルダ	開発環境	バージョン	コンパイラ名	バージョン
cubesuite	CubeSuite+	2.00.00	CC-RX	2.00.00.16
hew	HEW	4.09.01.007	RX Standard Toolchain	1.2.1.0

1. 4 PC-AOA-01 使用時の通信設定

1. 4. 1 通信設定概要

PC-AOA-01 を使用して Android 端末とボードとの通信を行う際には、通信開始時に Android 端末から PC-AOA-01 へと通信設定パケットを送信する必要があります。

通信開始時に通信設定パケットを PC-AOA-01 に送信することで自動で通信設定を行い、Android 端末とボード間は PC-AOA-01 の存在を意識することなく透過的に通信を行うことが可能となります。

通信設定を行う際には、Android 端末から送信する通信設定パケットの内容と、ボード側で設定する通信設定の内容を必ず同一のものとしてください

本サンプルでは、Android 端末が USB アクセサリ接続を感知した時に自動でサンプルアプリケーションが起動し、その直後に通信設定パケットを送信して RX63N との通信を開始します。

通信設定の流れを以下の Fig1.4-1 に示します。

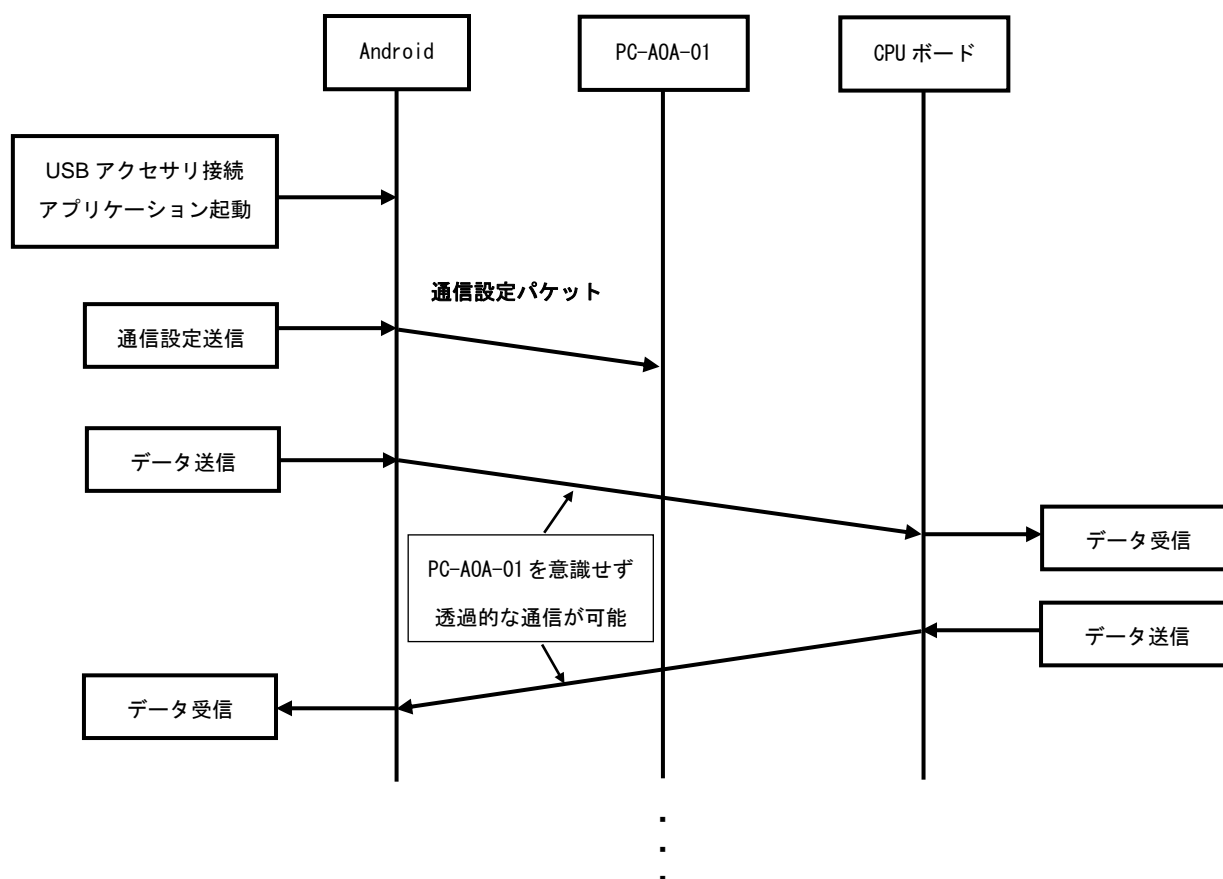


Fig 1.4-1 PC-AOA-01 を利用した通信例とデータの流れ

1. 4. 2 通信設定パケット

通信設定パケットは、ボーレート値、データビット長、ストップビット数、パリティの有無、フロー制御の有無の5つの要素で構成されています。

ボーレート値が32bitで表わされるため4Byte必要となり、その他の要素が1Byteで表わされるため、通信パケットのサイズは合計で8Byteとなっております。

PC-A0A-01を用いて通信を行う際には、以下の設定値を参考にして、Androidとボードとの通信開始時に通信設定パケットの送信を行なってください。

ボーレート 最下位 8bit	ボーレート	ボーレート	ボーレート 最上位 8bit	データ ビット長	ストップ ビット数	パリティ 有無	フロー制御 有無
-------------------	-------	-------	-------------------	-------------	--------------	------------	-------------

ボーレート

ボーレートは下位 8bit から順番に格納します。

例えば 38400bps のボーレート値を設定する場合は、16 進数表現で「0x00009600」となるため、

0x00	0x96	0x00	0x00
------	------	------	------

という順番で 1Byte ずつ送信します。

データビット長

設定値	概要
0x01	ビット長 8bit
0x02	ビット長 7bit

ストップビット数

設定値	概要
0x01	ストップビット数 1bit
0x02	ストップビット数 2bit

パリティ有無

設定値	概要
0x00	パリティなし
0x01	odd (奇数パリティ)
0x02	even (偶数パリティ)
0x03	mark (常に 1)
0x04	space (常に 0)

フロー制御有無

設定値	概要
0x00	フロー制御なし
0x01	フロー制御あり

1. 5 通信に関する基本設定

本サンプルの Android 端末と RX63N 間の通信における設定値は以下となります。

ボーレート	38400bps
ビット長	8
パリティ	なし
ストップビット	1
フロー制御	なし

1. 6 RX63N 用サンプルプログラムの移植について

1. 6. 1 弊社 CPU ボードへの移植

本サンプルプログラムは、弊社の CPU ボードである RX63N での動作を対象に作成しておりますが、一部改変を行うことにより、弊社の他の CPU ボードに本サンプルプログラムを移植することが可能です。

移植のためにソースの変更を行う際には、移植対象の CPU ボードに添付されているサンプルプログラムを参考にしていただくと比較的容易に移植を行うことが可能です。

改変を行う主な箇所としては、タイマ処理 (tmr.c)、シリアル通信処理 (sci.c)、LED 点灯処理 (led.c)、サンプルプログラムメイン処理 (sample.c) におけるレジスタ値の変更となります。

LED 点灯処理につきましては、各 CPU ボードに添付されているサンプルプログラムには実装されていないため、それぞれのボードのハードウェアマニュアルを参考にして変更を行ってください。

ただし、CPU 能力やメモリ容量等によっては正常に動作しない場合がありますのでご注意ください。

1. 6. 2 その他の機器への移植

本サンプルプログラムを動作させるためにはシリアル通信ソフトウェア及びタイマ割り込みソフトウェアが必要となります。

また、シリアル通信ではフロー制御が無いため十分なデータ受信能力が必要となります。

移植する際につきましては、シリアル通信処理に関しましては sci.c、タイマ割り込み処理に関しましては tmr.c を参考にしていただくと、比較的容易に移植を行うことが可能となっております。

2. サンプルプログラム動作説明

2. 1 動作概要

Android 端末と RX63N に本サンプルをそれぞれ導入した後に、Android 端末と RX63N を PC-A0A-01 を介して接続することで Android サンプルアプリケーションが自動的に起動します。

Android 上でボタンを押したりシークバーを操作することで LED 点灯状況が変化します。

USB アクセサリ接続を解除すると、Android サンプルアプリケーションが閉じて通信が終了します。

それぞれの機器の接続は Fig. 2. 1-1 を参考にしてください。

PC-A0A-01 の CN2 と RX63N の CN6 をシリアルインターフェースケーブルで接続し、PC-A0A-01 の CN1 と Android 端末を USB ケーブルで接続して使用してください。

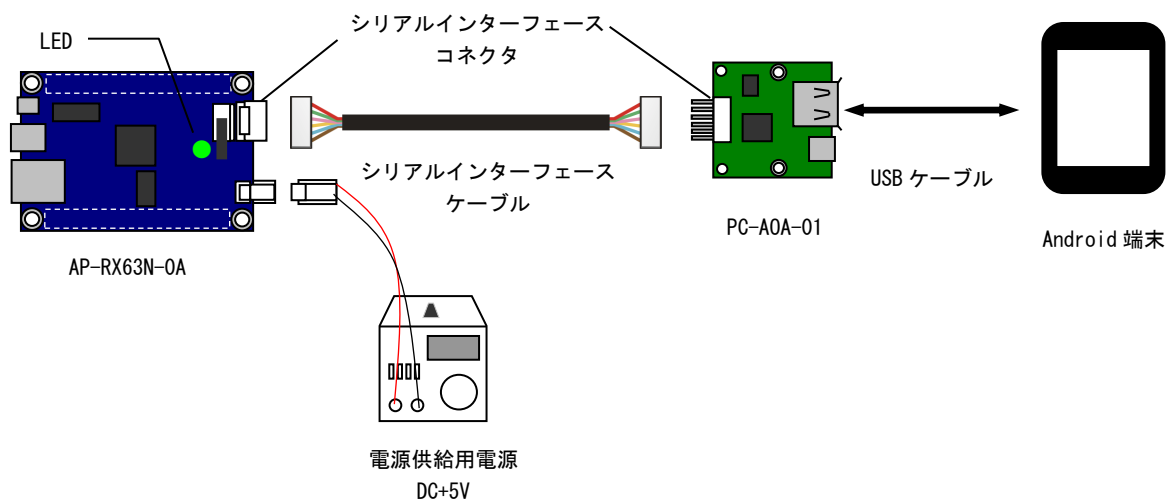


Fig 2. 1-1 PC-A0A-01 と Android 端末、RX63N の接続例

2. 2 サンプルプログラム動作モード設定

本サンプルプログラムを使用する際には以下の各 SW の設定を行ってください。

その他のモードの設定方法等につきましては、「AP-RX63N-0A ハードウェアマニュアル」をご覧ください。

なお、下記以外の条件で動作させる場合には、ソースファイルやコンパイラオプションなどを変更する必要があります。

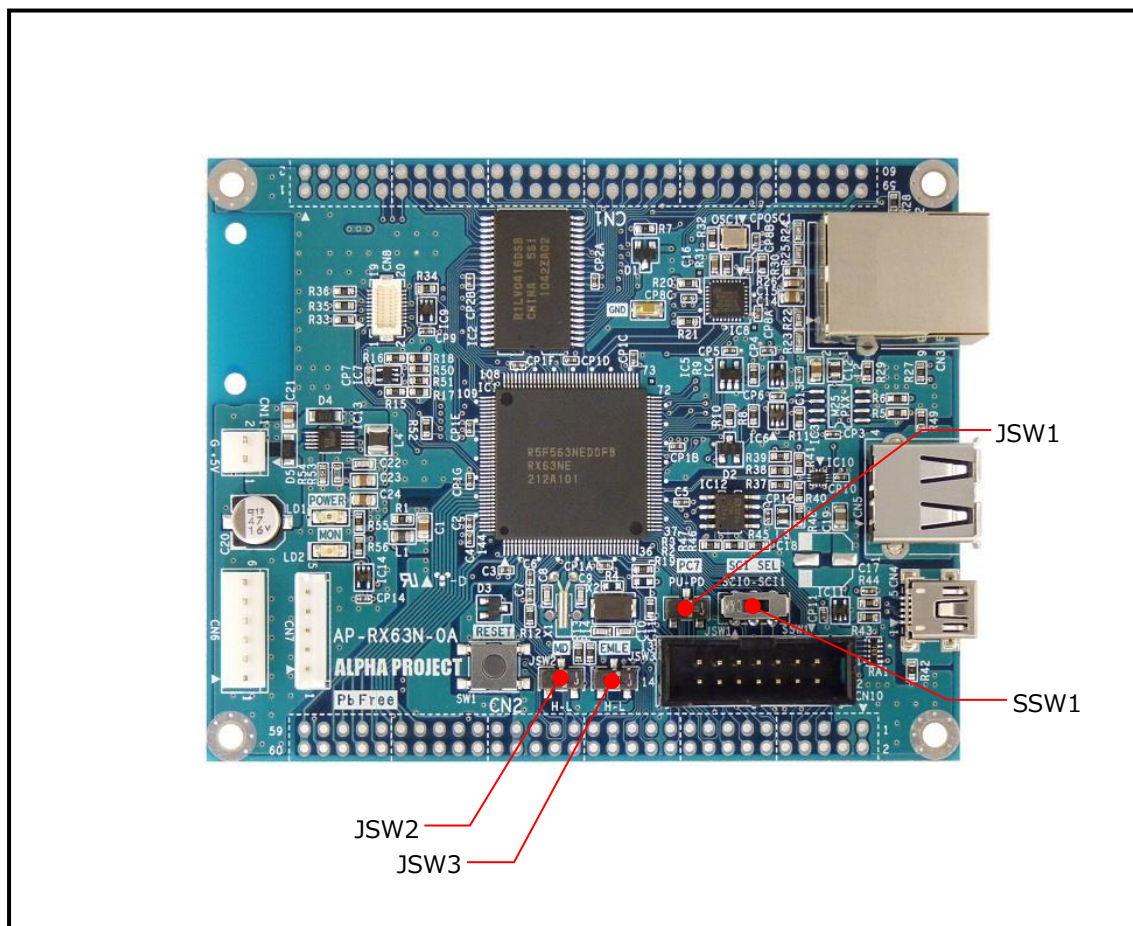


Fig 2. 2-1 AP-RX63N-0A 外形図

JSW1 の設定



Fig 2. 2-2 動作時の JSW1 設定

JSW2、JSW3 の設定

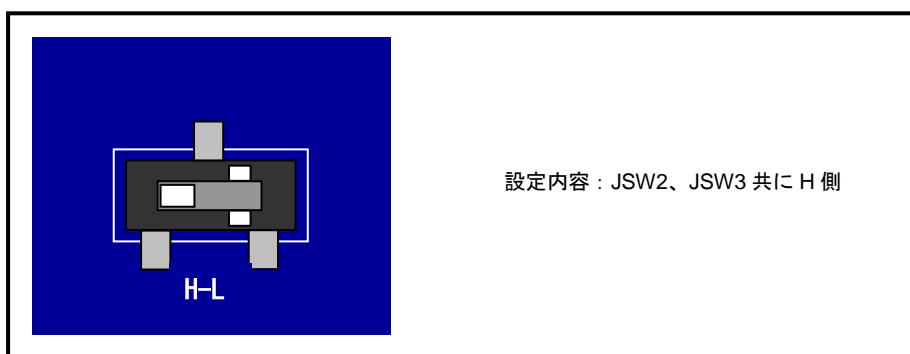


Fig 2. 2-3 動作時の JSW2、JSW3 設定

SSW1 の設定

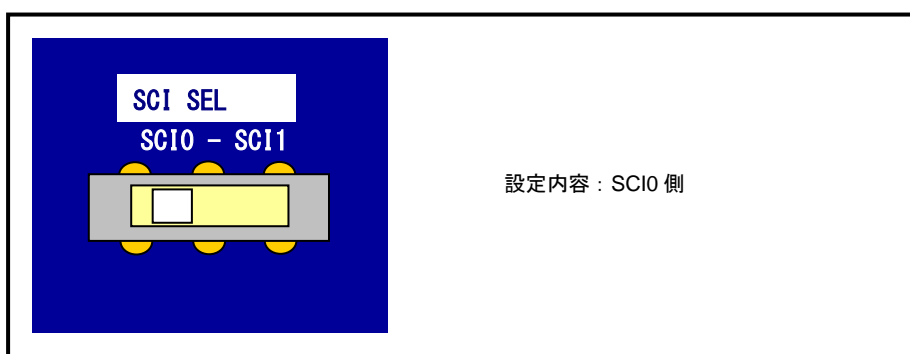


Fig 2. 2-4 動作時の SSW1 設定

2. 3 動作手順

- ①. 使用する Android 端末にサンプルアプリケーションをインストールします。
Google Play を開き、「aoa sample」と検索をすることで、弊社提供の AOA Sample がアプリケーション一覧に表示されますので、そちらを選択して、インストールを行なってください。

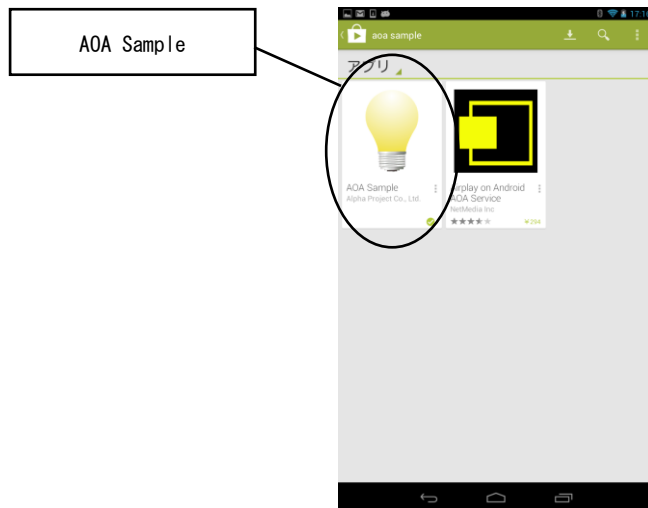


Fig 2. 3-1 AOA Sample 検索画面 (Nexus7 使用時)

また、下記 Fig2. 2-2 の QR コードを読み取ることで、AOA Sample のダウンロードページへ直接アクセスすることが可能です。



Fig 2. 3-2 AOA Sample ダウンロードページの QR コード

- ②. PC と RX63N を E1 エミュレータを介して接続し、RX63N にサンプルプログラムをダウンロードしてください。
サンプルプログラムは CubeSuite+版と HEW 版がそれぞれ用意されています。
それぞれのサンプルプログラムの実行方法を以下に示します。

・ CubeSuite+版

- ① PC と RX63N を E1 エミュレータを介して接続します。
- ② CubeSuite+を起動し、サンプルプログラムの cubesuite フォルダ内にある「pc_aoa_01_sample.mtpj」を読み込みます。
- ③ メニューの[デバッグ] - [デバッグツールの接続]を選択してください。RX63N へプログラムを送信する準備が整います。
- ④ メニューの[デバッグ] - [ビルド&デバッグ・ツールへダウンロード]を選択してください。ビルドが完了すると RX63N にプログラムがダウンロードされます。
- ⑤ メニューの [デバッグ] - [実行] を選択するとサンプルプログラムが実行されます。

・ HEW 版

- ① PC と RX63N を E1 エミュレータを介して接続します。
- ② HEW を起動し、サンプルプログラムの hew フォルダ内にある「pc_aoa_01_sample.hwp」を読み込みます。
- ③ 最初の読み込みを行なったときに、「ワークスペース (Workspace) が移動しました」という内容の確認メッセージが表示されますので「はい」を選択してください。
- ④ 最初の読み込みを行なったときに、コンパイラバージョンによって、バージョンの選択を行なうダイアログが表示されることがあります。表示された場合には、使用するコンパイラバージョンを選択してください。
- ⑤ メニューの [ビルド] - [すべてをビルド] を選択してください。ビルドが完了すると RX63N へとプログラムをダウンロードする旨の確認要求が出ますので、「はい」を選択してください。
- ⑥ メニューの [デバッグ] - [実行] を選択するとサンプルプログラムが実行されます。

- ③. PC-A0A-01 を介して、Android 端末と RX63N を接続してください。USB アクセサリの接続を感知すると、Android サンプルアプリケーションが自動的に起動します。
- USB アクセサリに接続せずに Android サンプルアプリケーションを起動した場合は、「USB Accessory is not connected!」という表示だけを行い通信処理は発生しません。

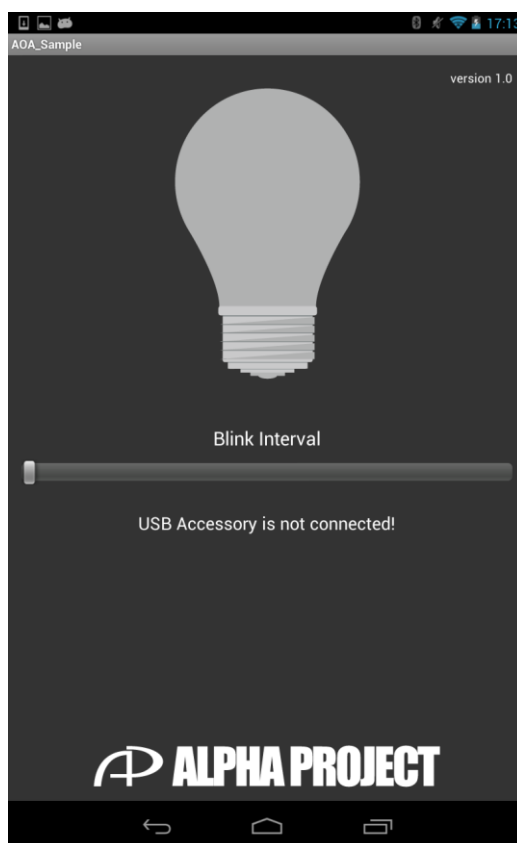


Fig 2.3-3 USB アクセサリ非接続時 (Nexus7 使用時)

- ④. サンプルアプリケーションのボタンやシークバーを操作するたびに Android から LED 制御のパケットが送信され、RX63N 上の LED 制御が行われます。
LED 制御パケット送信後は Android は RX63N からの応答パケットを待ちます。2 秒間応答パケットを受信できない場合はタイムアウトエラーを表示します。

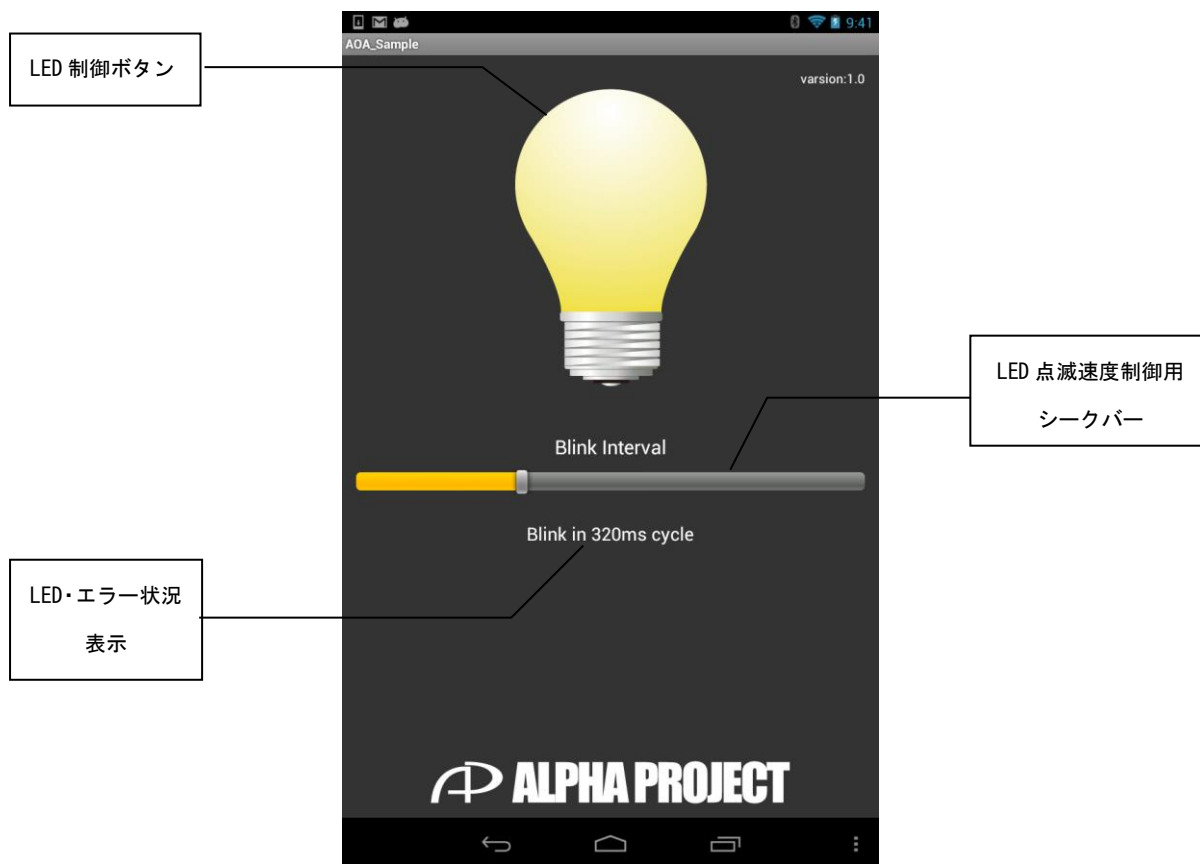


Fig 2.3-4 サンプルアプリケーション画面構成 (Nexus7 使用時)

- ⑤. Android 端末上の戻るボタンを押す、あるいは Android 端末の USB アクセサリ接続を解除するとサンプルアプリケーションが終了します。




2. 4 画面構成要素説明

・LED 制御ボタン

電球のアイコンにタッチすると、電球の画像が切り替わるとともに、LED 制御のパケットが送信されて RX63N 上の LED の ON と OFF も切り替わります。

ただし、USB アクセサリが未接続の状態では電球のアイコンにタッチしても動作はしません。

使用している画像は以下のとおりです。

	<p>LED が点灯中、あるいは点滅中であることを示します。 この画像にタッチをすると LED が消灯し、LED 消灯を示す画像に切り替わります。</p>
	<p>LED が消灯中であることを示します。 この画像にタッチをすると LED が点灯し、LED 点灯、点滅を示す画像に切り替わります。</p>
	<p>USB アクセサリが未接続であることを示します。 この画像にタッチしても何も起こりません。</p>

・LED 点滅速度制御用シークバー

LED が点灯、点滅している状態でシークバーにタッチ、スライドすることで LED 点滅速度が変化します。

左側に動かすと点滅速度が早くなり、右側に動かすと点滅速度が遅くなります。

LED が消灯時、USB アクセサリ未接続時にタッチしても何も起こりません。

・LED、エラー状況表示

RX63N 上の LED の点灯状況表示と、エラーを検出した際にエラーの内容を表示します。
表示内容と状態の一覧は以下のとおりです。

表示内容	概要
LED ON	LED が点灯中です。
LED OFF	LED が消灯中です。
Blink in XX ms cycle	LED が点滅中です。XX は点滅周期を示します。
Timeout error!	タイムアウトエラーを検出しました。
Checksum error! (RX)	RX63N 用サンプルプログラムがチェックサムエラーを検出しました。
Checksum error! (android)	Android 用サンプルアプリケーションがチェックサムエラーを検出しました。
Command error! (RX)	RX63N 用サンプルプログラムがコマンドエラーを検出しました。
Command error! (android)	Android 用サンプルアプリケーションがコマンドエラーを検出しました。
Parameter error! (RX)	RX63N 用サンプルプログラムがパラメータエラーを検出しました。
Parameter error! (android)	Android 用サンプルアプリケーションがパラメータエラーを検出しました。
USB Accessory is not connected!	USB アクセサリが接続されていません。

3. Android サンプルアプリケーション概要

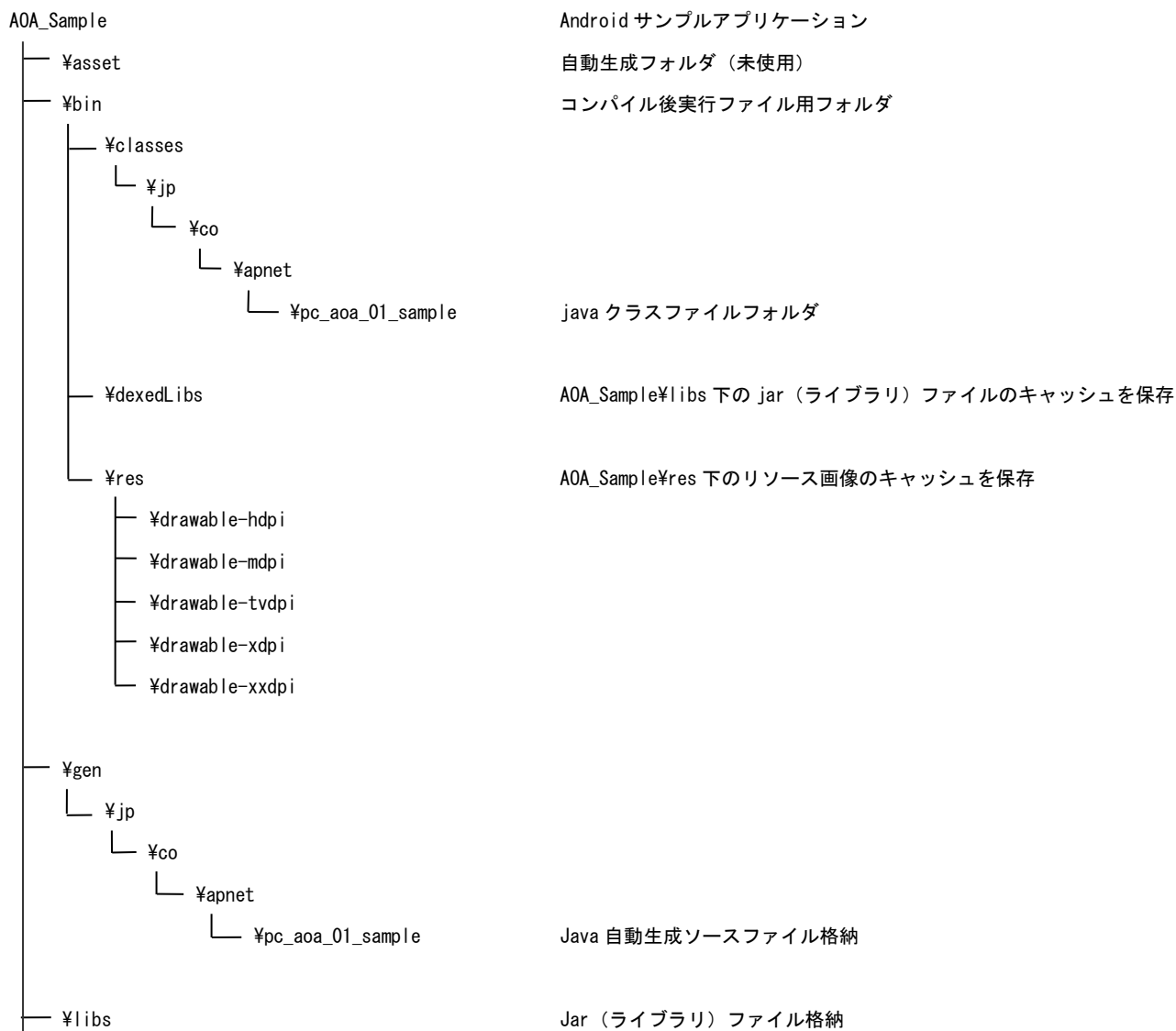
3. 1 動作概要

Android 用サンプルプログラムは、下記の動作を行います。

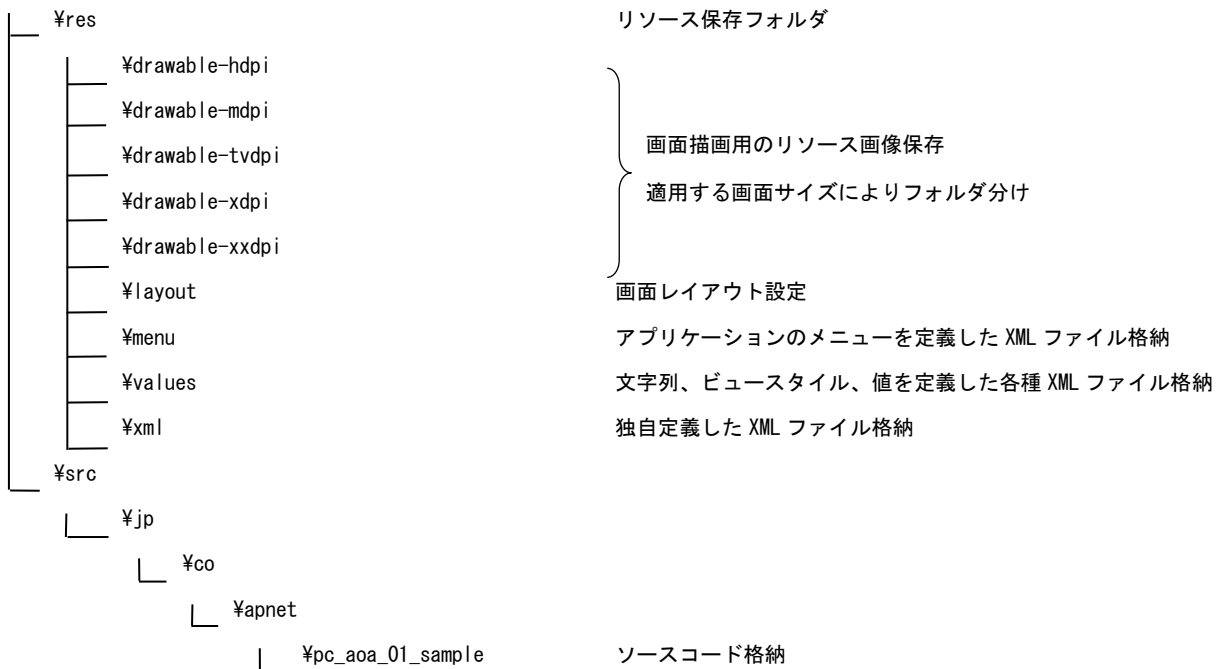
- Android 端末が USB アクセサリの接続を検出するとアプリケーションが自動起動します。起動後、通信設定を自動で行い、ボタンやシークバーを操作することで RX63N 上の LED を点灯制御するデータを送信します。USB アクセサリの接続が解除されると自動的にアプリケーションが終了します。

3. 2 フォルダ構成

Android 用サンプルアプリケーションは以下のような構成になっています。



次ページに続きます。



3. 3 ファイル構成

Android 用サンプルアプリケーションは以下のファイルで構成されています。

<AOA_Sample フォルダ内>

.classpath	コンパイル時のクラスパス設定ファイル
.project	アプリケーション（プロジェクト）設定ファイル
AndroidManifest.xml	アプリケーション定義 XML ファイル
Ic_launcher_web.png	高解像度アプリケーションアイコン（512×512px）
Proguard_project.txt	プロジェクトのターゲットバージョン記述
Project.properties	Java プログラムの圧縮、最適化、難読化

<AOA_Sample\gen\jp\co\apnet\pc_aoa_01_sample フォルダ内>

BuildConfig.java	デバッグビルド/リリースビルド自動切り替え
R.java	リソース管理ファイル

<AOA_Sample\res\drawable-hdpi フォルダ内>

aplogo.png	弊社ロゴデータ（550×64px）
ic_launcher.png	起動用アイコン画像（72×72px）
lampdisable.png	USB アクセサリ未接続時に表示する画像（200×300px）
lampoff.png	LED 点灯、点滅時に LED 制御ボタンとして表示する画像（200×300px）
lampon.png	LED 消灯時に LED 制御ボタンとして表示する画像（200×300px）

<AOA_Sample\res\drawable-mdpi フォルダ内>

aplogo.png	弊社ロゴデータ（550×64px）
ic_launcher.png	起動用アイコン画像（48×48px）
lampdisable.png	USB アクセサリ未接続時に表示する画像（200×300px）
lampoff.png	LED 点灯、点滅時に LED 制御ボタンとして表示する画像（200×300px）
lampon.png	LED 消灯時に LED 制御ボタンとして表示する画像（200×300px）

<AOA_Sample\res\drawable-tvdpi フォルダ内>

aplogo.png	弊社ロゴデータ（550×64px）
ic_launcher.png	起動用アイコン画像（96×96px）
lampdisable.png	USB アクセサリ未接続時に表示する画像（300×450px）
lampoff.png	LED 点灯、点滅時に LED 制御ボタンとして表示する画像（300×450px）
lampon.png	LED 消灯時に LED 制御ボタンとして表示する画像（300×450px）

<AOA_Sample\res\drawable-xhdpi フォルダ内>

aplogo.png	弊社ロゴデータ（550×64px）
ic_launcher.png	起動用アイコン画像（96×96px）
lampdisable.png	USB アクセサリ未接続時に表示する画像（300×450px）
lampoff.png	LED 点灯、点滅時に LED 制御ボタンとして表示する画像（300×450px）
lampon.png	LED 消灯時に LED 制御ボタンとして表示する画像（300×450px）

<AOA_Sample\res\drawable-xxhdpi フォルダ内>

aplogo.png	弊社ロゴデータ (1100×128px)
ic_launcher.png	起動用アイコン画像 (144×144px)
lampdisable.png	USB アクセサリ未接続時に表示する画像 (400×600px)
lampoff.png	LED 点灯、点滅時に LED 制御ボタンとして表示する画像 (400×600px)
lampon.png	LED 消灯時に LED 制御ボタンとして表示する画像 (400×600px)

<AOA_Sample\res\layout フォルダ内>

activity_aoa_sample.xml	画面レイアウト設定
-------------------------	-----------

<AOA_Sample\res\values フォルダ内>

dimens.xml	画面レイアウト外枠のマージン設定
strings.xml	アプリケーションタイトルの設定
styles.xml	背景色、文字色など、画面レイアウト全体のテーマ設定

<AOA_Sample\res\xml フォルダ内>

accessory_filter.xml	USB アクセサリとして使用可能なデバイスの設定
custom_ledbutton	LED 制御ボタンの画像設定

<AOA_Sample\src\jp\co\apnet\pc_aoa_01_sample フォルダ内>

AOA_Sample.java	メイン処理
AOAInterface.java	USB アクセサリ接続用プログラム FTDI 社提供の USB アクセサリ接続用プログラム「FTD311UARTInterface.java」 をベースに一部改変して作成

3. 4 Eclipse を用いたインストール方法

Android サンプルアプリケーションは章番号 2.3 で紹介した、Google Play 経由でインストールする方法の他にも、Eclipse を利用して Android 端末にインストールする方法があります。
インストール方法を以下に示します。

- ① PC から Eclipse を起動し、メニューの [ファイル] - [インポート] を選択してください。
- ② インポート・ソースの選択画面から [一般] - [Existing Android Code Into Workspace] を選択し [次へ (N)] を選択してください。その後、ルート・ディレクトリの [参照...] から AOA_Sample フォルダを選択すると、インポート可能なプロジェクトが選択出来ますので、AOA_Sample を示す項目にチェックを入れて [完了 (F)] を選択してください。
- ③ 実行させたい Android 端末を PC に接続します。
- ④ パッケージ・エクスプローラ内の AOA_Sample フォルダを右クリックし、[実行] - [Android アプリケーション] を選択してください。
- ⑤ [実行中の Android デバイスを選択する] にチェックを入れ、接続されている Android 端末を選択し、[OK] を選択することで、Android 端末でサンプルアプリケーションがインストールされます。
- ⑥ インストール完了後、USB アクセサリを接続することでアプリケーションが自動的に実行されます。

4. RX63N 用サンプルプログラム概要

4. 1 動作概要

RX63N 用サンプルプログラムは、下記の動作を行います。

- Android 用サンプルアプリケーションから受信した各種データに対応して、LED の点灯制御、Android 端末への LED 点灯状況送信、エラー検出した場合のエラー送信を行います。

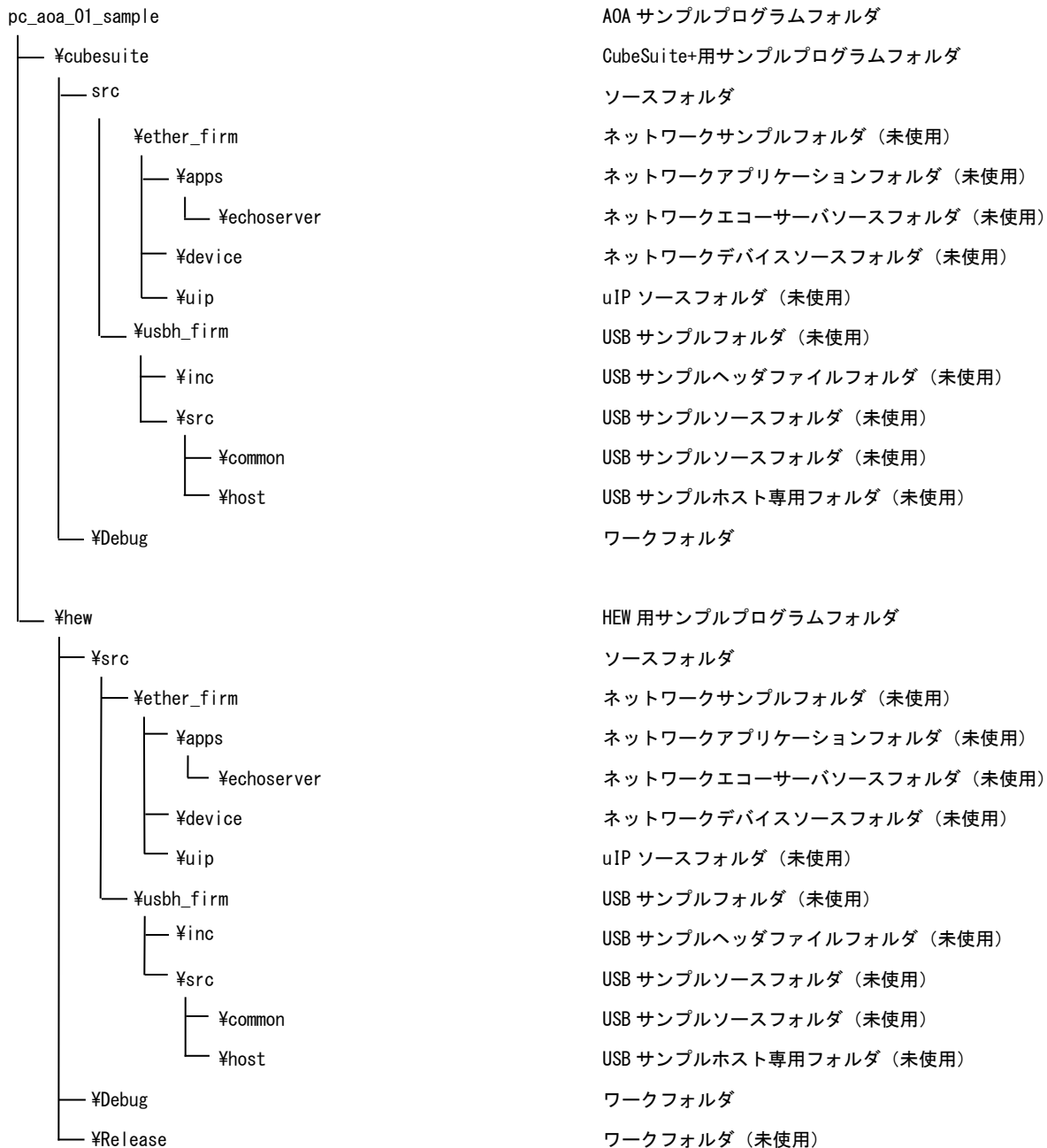
4. 2 フォルダ構成

RX63N 用サンプルプログラムは以下のような構成になっています。

本サンプルプログラムは、弊社の Web サイトにて公開している RX63N 用サンプルプログラムである

「ap_rx63n_0a_usbhost」をベースとして作成しております。

そのため、本サンプルプログラムで未使用であるフォルダには（未使用）と記載しております。



4. 3 ファイル構成

RX63N 用サンプルプログラムは以下のファイルで構成されています。

本アプリケーションノートでは、サンプルプログラムの処理に必要な箇所のみを抜粋して記載しております。

また、RX63N 用サンプルプログラム ap_rx63n_0a_usbhost と比較して、新しく追加した箇所には（新規追加）、変更を加えた部分については（変更）と記載しております。

4. 3. 1 CubeSuite+用サンプルプログラム

<pc_aoa_01_sample%cubesuite フォルダ内>

pc_aoa_01_sample.mtpj	CubeSuite+用プロジェクトファイル
-----------------------	-----------------------

<pc_aoa_01_sample%cubesuite%src フォルダ内>

aoa_sample.c	Android サンプルアプリケーションとの通信（新規追加）
aoa_sample.h	aoa_sample.c、led.c におけるヘッダ定義ファイル（新規追加）
BoardDepend.h	ボード依存定義ヘッダファイル
common.h	共通ヘッダファイル
dbsect.c	データセクション初期化処理
intprg.c	割り込み処理
iodefine.h	内部レジスタ定義ヘッダファイル
led.c	LED 点灯制御処理（新規追加）
pc_aoa_01_sample.c	メイン処理（新規追加）
resetprg.c	リセット・電源投入後起動処理
sample.c	サンプルプログラムメイン処理（変更）
sbrk.c	メモリ確保処理
sbrk.h	メモリ確保ヘッダファイル
sci.c	シリアル処理
tmr.c	タイマ処理（変更）
typedefine.h	型定義ヘッダファイル
vect.h	割り込みベクタテーブルヘッダファイル
vecttbl.c	割り込みベクタテーブル

4. 3. 2 HEW 用サンプルプログラム

<pc_aoa_01_sample¥hew フォルダ内>

pc_aoa_01_sample.hws	HEW 用ワークスペースファイル
----------------------	------------------

<pc_aoa_01_sample¥hew¥pc_aoa_01_sample フォルダ内>

pc_aoa_01_sample.hwp	HEW 用プロジェクトファイル
----------------------	-----------------

<pc_aoa_01_sample¥hew¥pc_aoa_01_sample¥src フォルダ内>

aoa_sample.c	Android サンプルアプリケーションとの通信（新規追加）
aoa_sample.h	aoa_sample.c、led.c におけるヘッダ定義ファイル（新規追加）
BoardDepend.h	ボード依存定義ヘッダファイル
common.h	共通ヘッダファイル
dbsect.c	データセクション初期化処理
intprg.c	割り込み処理
iodef.h	内部レジスタ定義ヘッダファイル
led.c	LED 点灯制御処理（新規追加）
pc_aoa_01_sample.c	メイン処理（新規追加）
resetprg.c	リセット・電源投入後起動処理
sample.c	サンプルプログラムメイン処理（変更）
sbrk.c	メモリ確保処理
sbrk.h	メモリ確保ヘッダファイル
sci.c	シリアル処理
tmr.c	タイマ処理（変更）
typedef.h	型定義ヘッダファイル
vect.h	割り込みベクタテーブルヘッダファイル
vecttbl.c	割り込みベクタテーブル

<pc_aoa_01_sample¥hew¥pc_aoa_01_sample¥Debug¥>

ap_rx63n_0a_usbhost.abs	実行用オブジェクトファイル（elf 形式）
ap_rx63n_0a_usbhost.map	マップファイル
ap_rx63n_0a_usbhost.mot	実行用モトローラ S フォーマット形式ファイル
	コンパイル後は「.obj」、「.lib」等のファイルが生成されます

<pc_aoa_01_sample¥hew¥pc_aoa_01_sample¥Release¥>

（未使用）	未使用 （HEW がデフォルトで作成するフォルダですが、 本サンプルプログラムでは未使用となっています）
-------	--

5. サンプルプログラムにおけるシリアル通信構成

5. 1 通信パッケージ構成

5. 1. 1 基本パッケージ構成

Android 端末と RX63N 間におけるシリアル通信の基本パッケージ構成は以下のとおりになっております。

STX<0x02>	COMMAND	DATA	ETX<0x03>	CHECK
1Byte	1Byte	1Byte	1Byte	1Byte

5. 1. 2 パラメータ概要

パッケージは STX から送信され、CHECK が最終バイトとなります。
データ長は、各パラメータで 1Byte ずつ、計 5Byte となります。

・パラメータ一覧

パラメータ	概要
STX	0x02 固定
COMMAND	実行するコマンド番号を格納する。各種コマンド番号は後述します
DATA	各コマンドにおける必要なデータを格納します
ETX	0x03 固定
CHECK	STX から ETX までのデータの排他的論理和を格納し、チェックサムとします

5. 1. 3 各種コマンド説明

コマンドの種類は、Android 端末から RX63N へと送信されるコマンドパッケージと、RX63N がコマンドパッケージを受信した際に RX63N から Android 端末へと送信される応答パッケージの 2 種類があります。

コマンドパッケージは 2 種類あり、以下に一覧を示します。

・コマンドパッケージ用コマンド番号一覧

コマンド番号	概要
0x01	LED 制御コマンド
0x02	LED 状況取得コマンド

LED 制御コマンド (0x01)

■動作説明

Android からボード上の LED の点灯、消灯、点滅処理を行います。

コマンドパケット

■パケット構成

STX<0x02>	<0x01>	DATA	ETX<0x03>	CHECK
1Byte	1Byte	1Byte	1Byte	1Byte

■パラメータ説明

パラメータ	値の範囲	概要
DATA	0x00	ボード上の LED を点灯させます。
	0x01~0x64	ボード上の LED を点滅させます。 0x01 で 10ms の点滅周期となり、そこから値が 1 上昇するごとに 10ms ずつ点滅周期が開いていき、0x64 では 1000ms (1s) 周期での点滅となります。
	0x7f	ボード上の LED を消灯させます。

応答パケット

■パケット構成

STX<0x02>	<0x81>	DATA	ETX<0x03>	CHECK
1Byte	1Byte	1Byte	1Byte	1Byte

■パラメータ説明

パラメータ	値の範囲	概要
DATA	0x00	ボード上の LED を点灯していることを示します。
	0x01~0x64	ボード上の LED を点滅していることを示します。
	0x7f	ボード上の LED を消灯していることを示します。
	0x80	チェックサムエラーを検出しました。
	0x81	コマンドエラーを検出しました。
	0x82	パラメータエラーを検出しました。

LED 状況取得コマンド (0x02)

■動作説明

ボード上の LED 状況を取得します。

コマンドパケット

■パケット構成

STX<0x02>	<0x02>	<0x00>	ETX<0x03>	CHECK
1Byte	1Byte	1Byte	1Byte	1Byte

応答パケット

■パケット構成

STX<0x02>	<0x82>	DATA	ETX<0x03>	CHECK
1Byte	1Byte	1Byte	1Byte	1Byte

■パラメータ説明

パラメータ	値の範囲	概要
DATA	0x00	ボード上の LED を点灯していることを示します。
	0x01~0x64	ボード上の LED を点滅していることを示します。
	0x7f	ボード上の LED を消灯していることを示します。
	0x80	チェックサムエラーを検出しました。
	0x81	コマンドエラーを検出しました。
	0x82	パラメータエラーを検出しました。

5. 2 Android・RX63N 間の通信手順

5. 2. 1 アプリケーション開始時

Android 端末が USB アクセサリの接続を感知するとサンプルアプリケーションが自動的に起動し、PC-A0A-01 に通信設定パケットを送信し、Android と RX63N 間の通信を可能にします。その後の Android と RX63N 間の通信はすべて PC-A0A-01 を介して行われます。

5. 2. 2 LED 制御パケットの送受信

アプリケーションが起動し、通信設定パケットの送信を終えた後は LED 状況取得、LED 制御の通信を行います。Android サンプルアプリケーションでボタン、シークバーの操作を行うことで LED 制御のコマンドパケットが RX63N 用サンプルプログラムに送られます。RX63N 用サンプルプログラムはそのパケットを解析し、応答パケットを Android へ送信します。LED 制御の通信はアプリケーション終了まで、ボタンやシークバーを操作するたび繰り返し発生します。

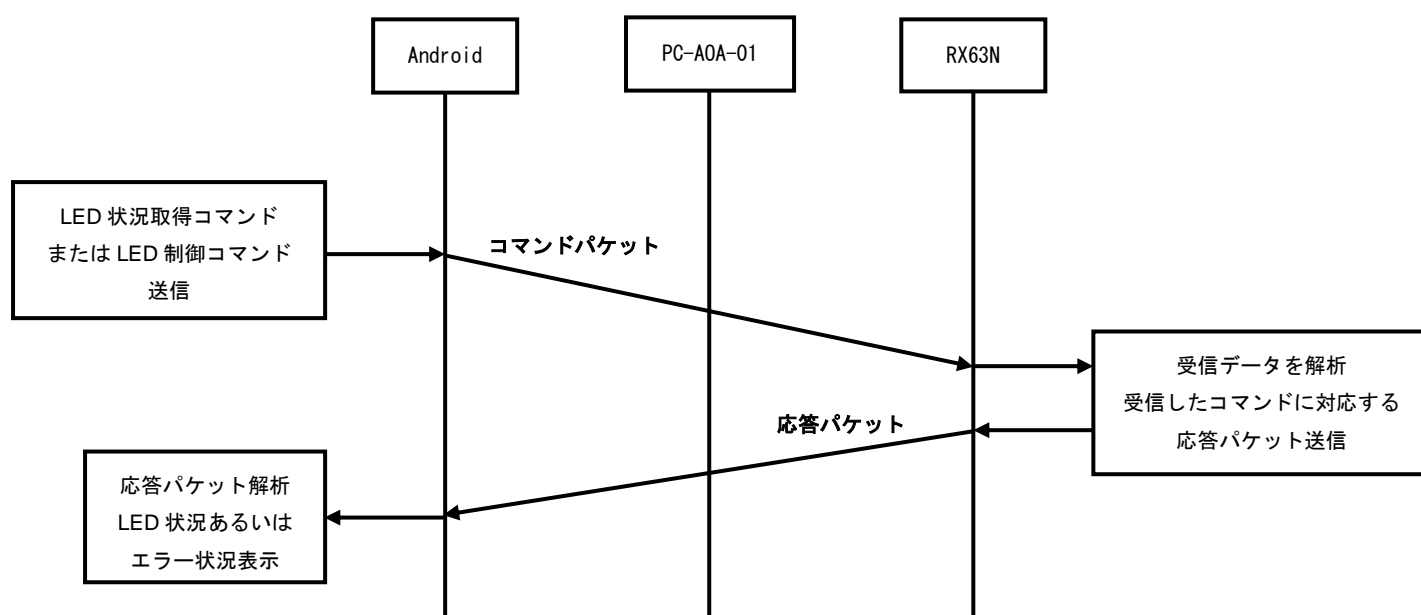


Fig 5.2-1 パケット送受信

5. 2. 3 アプリケーション終了時

USB アクセサリ接続を解除する、あるいは Android 端末上の戻るボタンを押した際にアプリケーションが終了します。アプリケーションを終了した後再度アプリケーションを起動したい場合は、USB アクセサリを再接続するか、もしくは USB アクセサリが接続された状態で、アプリケーション一覧からサンプルアプリケーションを再起動することで再び通信を行うことが可能です。

なお、戻るボタンを押してアプリケーションを終了させる場合、終了直前に Android サンプルアプリケーションから 1Byte のダミーデータを送信し、そのダミーデータを RX63N がループバックする形となっています。

これは、USB アクセサリが接続された状態でアプリケーションを再起動する際に、USB アクセサリとの通信を再び開始するために必要な処理となっております。

上記についての詳細は、FTDI 社のアプリケーションノート「AN_208 FT31xD Demo APK User Guide」を参照してください。

ご注意

- ・本文書の著作権は株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書に記載されているサンプルプログラムの著作権は株式会社アルファプロジェクトが保有しますが、お客様のアプリケーションで使用される場合には、ご自由にご利用いただけます。
- ・本文書の内容は、将来予告なしに変更されることがあります。
- ・本文書に記載されている内容およびサンプルプログラムについての質問等のサポートは一切受け付けておりませんのでご了承ください。
- ・本文書の内容については、万全を期して作成いたしました。万が一不審な点、誤りなどお気づきの点がありましたら弊社までご連絡下さい。
- ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。

- ・Android および Google Play は Google Inc.の商標または登録商標です。
- ・RX および RX63N は、ルネサスエレクトロニクス株式会社の登録商標、商標または商品名称です。
- ・その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト
〒431-3114
静岡県浜松市中央区積志町 834
<https://www.apnet.co.jp>
E-Mail: query@apnet.co.jp