

Renesas USB MCU

R01AN1965JJ0103

Rev.1.03

Mar 28, 2016

USB Host Android Open Accessory (AOA) using Basic Mini Firmware

要旨

本資料は、Renesas USB MCU の USB Basic Mini Firmware を使用した USB Host Android Open Accessory (AOA) のアプリケーションノートです。

動作確認デバイス

RL78/G1C

動作確認デバイスと同様の USB モジュールを持つ他の MCU でも本プログラムを使用することができます。このアプリケーションノートのご使用に際しては十分な評価を行ってください。

なお、本プログラムは Renesas Starter Kit 上で動作確認を行っています。

目次

1.	はじめに.....	2
2.	デバイスクラスドライバの登録.....	4
3.	Android Open Accessory (AOA)	4
4.	動作確認環境	4
5.	ソフトウェア構成	5
6.	ホスト Vendor AOA サンプルアプリケーションプログラム (APL)	9
7.	USB ホストベンダクラスドライバ (HVEN)	14
8.	制限事項.....	23
9.	e ² studio 用プロジェクトのセットアップ	24
10.	e ² studio 用プロジェクトを CS+で使用する場合.....	26

1. はじめに

本アプリケーションノートは、USB-BASIC-F/W (1.2 章を参照) を使用した USB Host Android Open Accessory (AOA) サンプルアプリケーションに関して記述しています。

1.1 機能と特長

USB Host Android Open Accessory (AOA) は Universal Serial Bus Revision 2.0 仕様 (以降 USB と記述) に準拠し、Android Open Accessory に対応した Android 機器との通信を行うことができます。

本クラスドライバは弊社の提供する USB Basic Mini Firmware と組み合わせて使用することを前提としています。

1.2 関連ドキュメント

1. Universal Serial Bus Revision 2.0 specification
 2. Android™(※) Open Accessory
[<http://developer.android.com/index.html>]
(※) Android is a trademark of Google Inc.
 3. ユーザーズマニュアル ハードウェア編
 4. Renesas USB MCU USB Basic Mini Firmware アプリケーションノート
ルネサス エレクトロニクスホームページ より入手できます。
- ルネサス エレクトロニクスホームページ
【<http://japan.renesas.com/>】
 - USB デバイスページ
【<http://japan.renesas.com/usb/>】

1.3 用語と略語

本書では使用される用語と略語は以下のとおりです。

AOA	: Android Open Accessory
API	: Application Program Interface
APL	: Application program
cstd	: USB-BASIC-F/Wの Peripheral & Host共通関数のprefix
CS+	: ルネサス統合開発環境
Data Transfer	: Generic name of Control transfer, Bulk transfer and Interrupt transfer
HCD	: Host control driver of USB-BASIC-F/W
HDCD	: Host device class driver (device driver and USB class driver)
HEW	: High-performance Embedded Workshop
HID	: Human interface device class
HM	: Hardware Manual
hstd	: Prefix for host function of USB-BASIC-F/W
HVEN	: Host Vendor
KBD	: Keyboard device
MGR	: Peripheral device state manager of HCD
MSE	: Mouse device
PP	: プリプロセス定義
RSK	: Renesas Starter Kit
SW1/SW2/SW3	: User switches on RSK
USB	: Universal Serial Bus
USB-BASIC-FW	: USB-BASIC-F/W (Host & Peripheral USB Basic Mini Firmware(USB low level) for Renesas USB MCU)
タスク	: 処理の単位
スケジューラ	: タスク動作を簡易的にスケジューリングするもの
スケジューラマクロ	: スケジューラ機能呼び出すために使用されるもの
データ転送	: Control転送、Bulk転送、Interrupt転送の総称

1.4 本書の読み方

本書は章の順番通りに読み進める必要はありません。はじめにサンプルプログラムの内容を確認し、ユーザ個別のソリューションに必要な関数およびインタフェースの情報をお読みください。

5.3章にソース一覧を掲載しています。MCU固有ソースは、"`\devicename\src\HwResource`"にあります。アプリケーションに必要なファイルを確認してください。

ユーザ独自のソリューションを作成するためにはアプリケーションの変更が必要です。6章はホストベンダアプリケーションの動作を説明しています。

すべてのコードモジュールはタスクに分割されます。タスク間でメッセージの受け渡しが行われていることを予めご理解ください。関数(タスク)の実行順序はスケジューラが決定します。このため重要なタスクに優先権を持たせることができます。また、タスクに登録されたコールバックメカニズムを使用することで、各タスクは並列処理(ノンブロッキング)で動作します。タスクのメカニズムは1.2章の"USB-BASIC-F/W Application Note"で説明しています。HVENのタスクについては5.4章を参照してください。

2. デバイスクラスドライバの登録

ユーザが作成したクラスドライバは、USB-BASIC-F/W に登録することで USB デバイスクラスドライバとして機能します。 `r_usb_vendor_hapl.c` ファイル内の `usb_hsmpl_driver_registration()` 関数を参考に USB-BASIC-F/W にクラスドライバを登録してください。詳細は、USB-BASIC-F/W のアプリケーションノートを参照してください。

3. Android Open Accessory (AOA)

本ソフトウェアは Android Open Accessory 1.0/2.0 仕様に準拠します。詳細は 1.2 章を参照してください。

Android Open Accessory 仕様は、Android OS を開発した Google 社が定めたインタフェース仕様で、以下の仕様が定められています。

- Hardware 接続仕様
- AOA プロトコル定義
- Android OS ソフトウェア API

Android 機器でこのインタフェース仕様を用いることができるのは、Android Platform API 10 以上を使用可能な物となります。すなわち、Android OS バージョン 2.3.4 以降(タブレットの場合は 3.1 以降)が対象となります。また AOA2.0 仕様は Android OS バージョン 4.1 以降が対象となります。

3.1 基本機能

AOA の主な機能は、以下のとおりです。

1. 接続デバイスの照合
2. デバイスの能力と状態について問い合わせ。
3. 出力状態と特徴項目の設定
4. Android 機器とのデータ送受信

4. 動作確認済環境

4.1 コンパイラ

動作確認を行ったコンパイラは以下の通りです。

- a. CA78K0R コンパイラ V.1.71
- b. CC-RL コンパイラ V.1.01

4.2 評価ボード

動作確認を行った評価ボードは以下の通りです。

- a. Renesas Starter Kit for RL78/G1C (型名: R0K5010JGC001BR)

5. ソフトウェア構成

5.1 モジュール構成

AOAはベンダークラスドライバと、デバイスドライバから構成されます。

Figure 5.1にAOAのソフトウェアモジュール構成図を、Table 5-1にモジュール説明を示します。

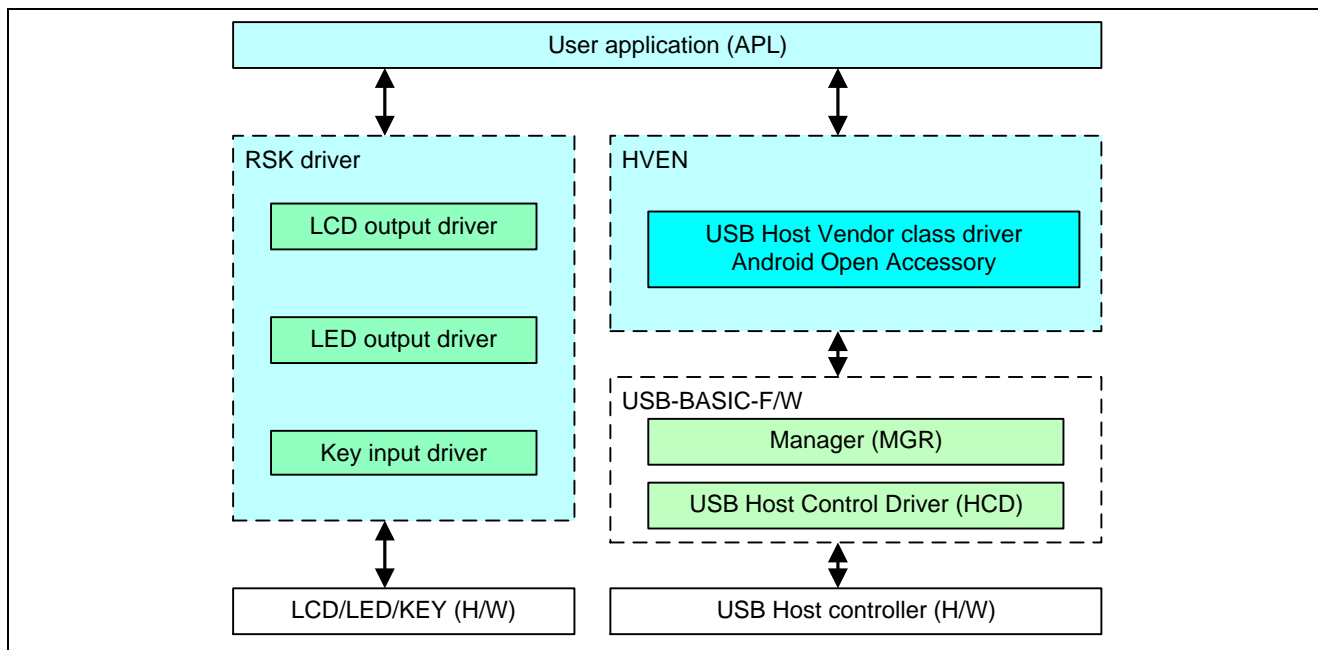


Figure 5.1 モジュール構成図

Table 5-1 モジュール機能概要

モジュール名	機能概要	備考
APL	ユーザアプリケーションプログラムです。 スイッチ操作によりAndroidデバイスとの通信を行います。	ユーザ作成
VENDOR	登録されたデバイスクラスドライバは接続デバイスの操作をチェックします。USB-BASIC-F/WはAPLを経由して接続デバイスとAOAの対応を確認します。APLの要求に従って、USB-BASIC-F/Wに以下のデータ転送要求を行います。 1) AOAプロトコルによる接続デバイスの制御 2) 接続デバイスとのデータ通信 転送結果はコールバック関数によってAPLに通知します。	
USB-BASIC-F/W	USBホストコントロールドライバです。 (ハードウェア制御とデバイスステート管理を行います。)	

5.2 アプリケーションプログラム機能概要

ホストデモ APL の主な機能は以下のとおりです。

1. 接続した Android 機器へ AOA プロトコルでデータを送受信し、Android 機器上で動作する Standard Android ADK Demo Kit アプリケーション(以下 Demo Kit アプリ)と通信を行う。
 - a) Demo Kit アプリが In モードの場合、RSK の SW1-3 及び RV1 を操作すると、Demo Kit アプリが状態表示を行う。
 - b) Demo Kit アプリが Out モードの場合、Demo Kit アプリ上の Servo1-3 のゲージ及び Led1-3 の RGB ゲージを操作すると、ゲージの中心をしきい値に、RSK ボードの LED0-2 の点灯/消灯が行われる。
2. AOA2.0 モードの場合、Android 機器画面以上にマウスカーソルが出現し、RSK ボードの SW1-3 の押下でマウスカーソル操作を行う。

Table 5-2 及び Table 5-3 に入出力の仕様を示します。

Table 5-2 Demo Kit In モード仕様

RSK ボード入力	動作内容
SW1	Buttons 「B1」 マーク 点灯 / 消灯
SW2	Buttons 「B2」 マーク 点灯 / 消灯 Joystick Y 軸方向入力 押下を繰り返すと Y 軸の移動方向が交互に入れ替わる。
SW3	Buttons 「B3」 マーク 点灯 / 消灯 Joystick X 軸方向入力 押下を繰り返すと X 軸の移動方向が交互に入れ替わる。
RV1	Android マーク 点灯 / 消灯(RV1 の AD 入力 が 128 以上で点灯) Temp / Light の値変更(0~255 範囲)

Table 5-3 Demo Kit Out モード仕様

Demo Kit 入力	動作内容
Servo1 Led1(RGB 共通)	RSK ボード LED0 操作 0~127 : 消灯 128~255 : 点灯
Servo2 Led(RGB 共通)	RSK ボード LED1 操作 0~127 : 消灯 128~255 : 点灯
Servo3 Led(RGB 共通)	RSK ボード LED2 操作 0~127 : 消灯 128~255 : 点灯
Relay1 & Relay2	動作なし

5.3 ファイル構成

5.3.1 フォルダ構成

以下に本サンプルプログラムで提供するファイルのフォルダ構成を示します。

各 MCU と評価基板に依存するソースコードはそれぞれのハードウェアリソースフォルダ (`\devicename\src\HwResource`) にあります。

workspace		
+ CS+/CCRL		
+ RL78G1C		プロジェクト・ファイル格納箇所
+ AOA1.0		ホストビルド結果(AOA1.0 仕様)
+ AOA2.0		ホストビルド結果(AOA2.0 仕様)
+ src		
+——— VENDOR [<i>Vendor Class driver</i>]		Table 5-4 参照
+——— inc		Vendor ドライバ共通ヘッダファイル
+——— src		Vendor ドライバ
+——— SmpMain [<i>サンプルアプリケーション</i>] Table 5-4 参照		
+——— APL		AOA 通信用アプリケーション
+——— inc		AOA 通信用共通ヘッダ
+——— USBSTDFW [<i>全ての USB ドライバに共通な基本ファームウェア</i>]		
+——— inc		USB ドライバ共通ヘッダファイル
+——— src		USB ドライバ
+——— HwResource [<i>MCU 初期化等のハードウェアアクセス層</i>]		
+——— inc		H/W リソースヘッダファイル
+——— src		H/W リソース

[Note]

- CS+フォルダ下には、CA78K0R コンパイラ用のプロジェクトが格納されています。
- CS+上で CC-RL コンパイラをご使用になる場合は、「10 e2 studio 用プロジェクトを CS+で使用する場合」を参照してください。

5.3.2 ファイル一覧

Table 5-4 に Vendor AOA が提供するファイル、フォルダ名を示します。

Table 5-4 ファイル構成

フォルダ名	ファイル名	説明
VENDOR/inc	r_usb_vendor_api.h	USB ホスト VENDOR AOA 定義マクロ
VENDOR/src	r_usb_vendor_hapi.c	USB ホスト VENDOR AOA API 関数
VENDOR/src	r_usb_vendor_hdriver.c	USB ホスト VENDOR AOA ドライバ関数
SmplMain	main.c	メインループ処理
SmplMain/inc	r_usb_aoa_common.h	USB ホスト VENDOR AOA 共通ヘッダ
SmplMain/APL	r_usb_vendor_hapl.c	サンプルアプリケーションプログラム

5.4 システムリソース

5.4.1 システムリソース定義

Vendor AOA をスケジューラへ登録して使用するためのタスク ID とタスク優先度定義を Table 5-5 に示します。

これらについては、`r_usb_kernelid.h` ヘッダファイルで定義します。

Table 5-5 スケジューラ登録 ID 一覧

スケジューラ登録タスク	説明	備考
USB_HVEN_TSK	HVEN (R_usb_hvndr_Task) Task ID: USB_HVEN_TSK Task priority: 2	
USB_HCD_TSK	HCD (R_usb_hstd_HcdTask) Task ID: USB_HCD_TSK Task priority: 0	
USB_MGR_TSK	MGR (R_usb_hstd_MgrTask) Task ID: USB_MGR_TSK Task priority: 1	
USB_HSMP_TSK	HSMP (usb_hsmpl_apl_task) Task ID: USB_HSMP_TSK Task priority: 3	
メールボックス ID / デフォルト受信タスク	メッセージ名称	備考
USB_HVEN_MBX / USB_HVEN_TSK	HVEN -> HVEN / APL -> HVEN mailbox ID	
USB_HCD_MBX / USB_HCD_TSK	HCD task mailbox ID	
USB_MGR_MBX / USB_MGR_TSK	MGR task mailbox ID	
USB_HSMP_MBX / USB_HSMP_TSK	Host Sample task mailbox ID	

6. ホスト Vendor AOA サンプルアプリケーションプログラム (APL)

ホストデモアプリケーションは Android 機器が接続されたとき、Android 機器上で動作する Demo Kit アプリケーションと通信を行います。Vendor AOA アプリケーションは Google 社が提唱している AOA Protocol に従って動作します。1.2 章の 2 項を参照してください。

6.1 動作環境について

Figure 6.1 に本ソフトウェアの動作環境を示します。

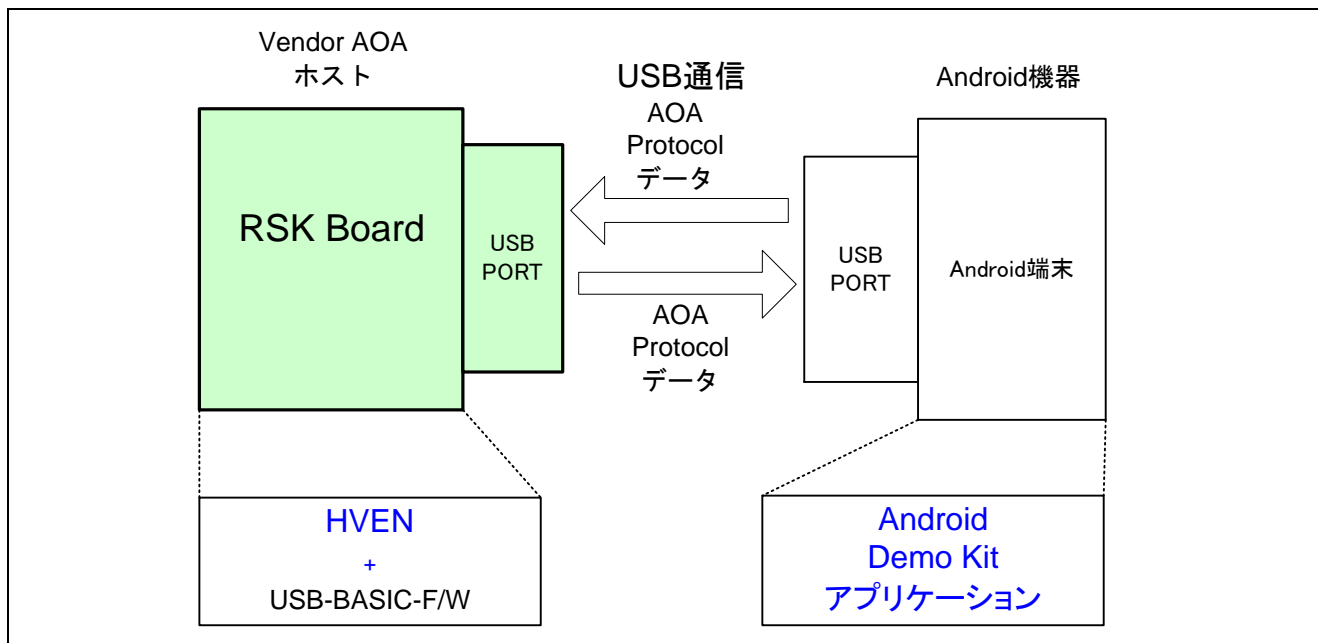


Figure 6.1 動作環境例

6.1.1 AOA2.0 通信

ビルドモードを AOA1.0 から AOA2.0 に切り替えると、ENABLE_AOA2_HID マクロが有効になり、Android 機器上で動作する Android OS に対し、HID リポートを送信することが可能になります。この時、Android 機器画面上にマウスカーソルが出現し、マウス動作をさせることが可能です。

6.2 アプリケーションプログラム説明

アプリケーション操作は以下のとおりです。

- Android 機器の接続

接続された USB デバイスが AOA Protocol を用いた通信が可能な Android 機器であることを自動判別します。可能であった場合、再エnumレーションを行い、バルク IN/OUT を用いたデータ通信を開始します。詳細は Figure 6.2 を参照してください。

- データ通信中の操作 (RSK ボード側)

RSK ボード上の SW1~3 及び RV1 を操作すると、Android Demo Kit 側へ HID マウス移動量及び温度/照度のセンサデータを生成して送信します。

AOA2.0 動作の場合、Android OS も HID マウス移動量を受信するため、Android 端末にマウスカーソルが出現し、HID マウス動作をさせることが可能です。

- データ通信中の操作 (Android Demo Kit 側)

Android Demo Kit 側を操作すると、RSK ボード上の LED の点灯/消灯設定を生成し、Vendor AOA 側で RSK ボードの LED を操作することが可能です。

Table 6-1 データ通信中の動作

対象	説明
RSK ボード	Android Demo Kit へ HID マウス移動量通知 AD データの送付により温度/照度データの更新を実施
Android Demo Kit 側	RSK ボード上の LED の点灯/消灯

6.3 エンドポイント仕様

Vendor AOA が使用するエンドポイント仕様を Table 6-2. に示します。

Table 6-2 エンドポイント仕様

EP 番号	Pipe 番号	転送方法	説明
0	0	Control In/Out	標準リクエスト、クラスリクエスト
受信した ディスクリプタに従う	4	Bulk In	デバイスからホストへのデータ転送
受信した ディスクリプタに従う	5	Bulk Out	ホストからデバイスへのデータ転送

エンドポイント番号はデバイスのエンドポイントディスクリプタに従います。

6.4 接続する Android 機器について

AOA プロトコル通信が可能な Android 機器が接続可能です。Android OS 2.3.4 以降(タブレットの場合は 3.1 以降)が対象です。AOA2.0 プロトコル通信の場合、Android OS4.1 以降が対象となります。

HUB 内蔵デバイスや複合デバイスは接続できません。

6.5 APL 関数一覧

サンプルアプリケーションの関数一覧を Table 6-3 に示します。

Table 6-3 サンプルアプリケーション関数一覧

関数名	説明
main	メインループ処理
usb_hsmpl_main_init	システム初期化 ホスト USB 用の各種タスクスタートアップ処理
usb_hsmpl_apl_task	Vendor AOA サンプルアプリケーションタスク
usb_hsmpl_driver_registration	Vendor AOA ドライバ登録
usb_hsmpl_class_check	接続デバイスチェック
usb_hsmpl_device_state	デバイスステート変化検出コールバック
usb_hsmpl_AOA_SendData	データ送信処理(Android 機器への送信)
usb_hsmpl_AOA_SendData_result	データ送信完了処理
usb_hsmpl_AOA_RecvData	データ受信処理(Android 機器からの受信)
usb_hsmpl_AOA_RecvData_result	データ受信完了処理
usb_hsmpl_AOA_mouse_switch_operation	RSK ボード SW 処理

6.6 シーケンスチャート

サンプルアプリケーションプログラムのシーケンスを以下に示します。

RSK ボードが SET_CONFIGURATION の送信を完了した後、Vendor Request を AOA Protocol に従って送信することにより、Android 端末のアクセサリとして動作します。

AOA Protocol を Figure 6.2 に示します。

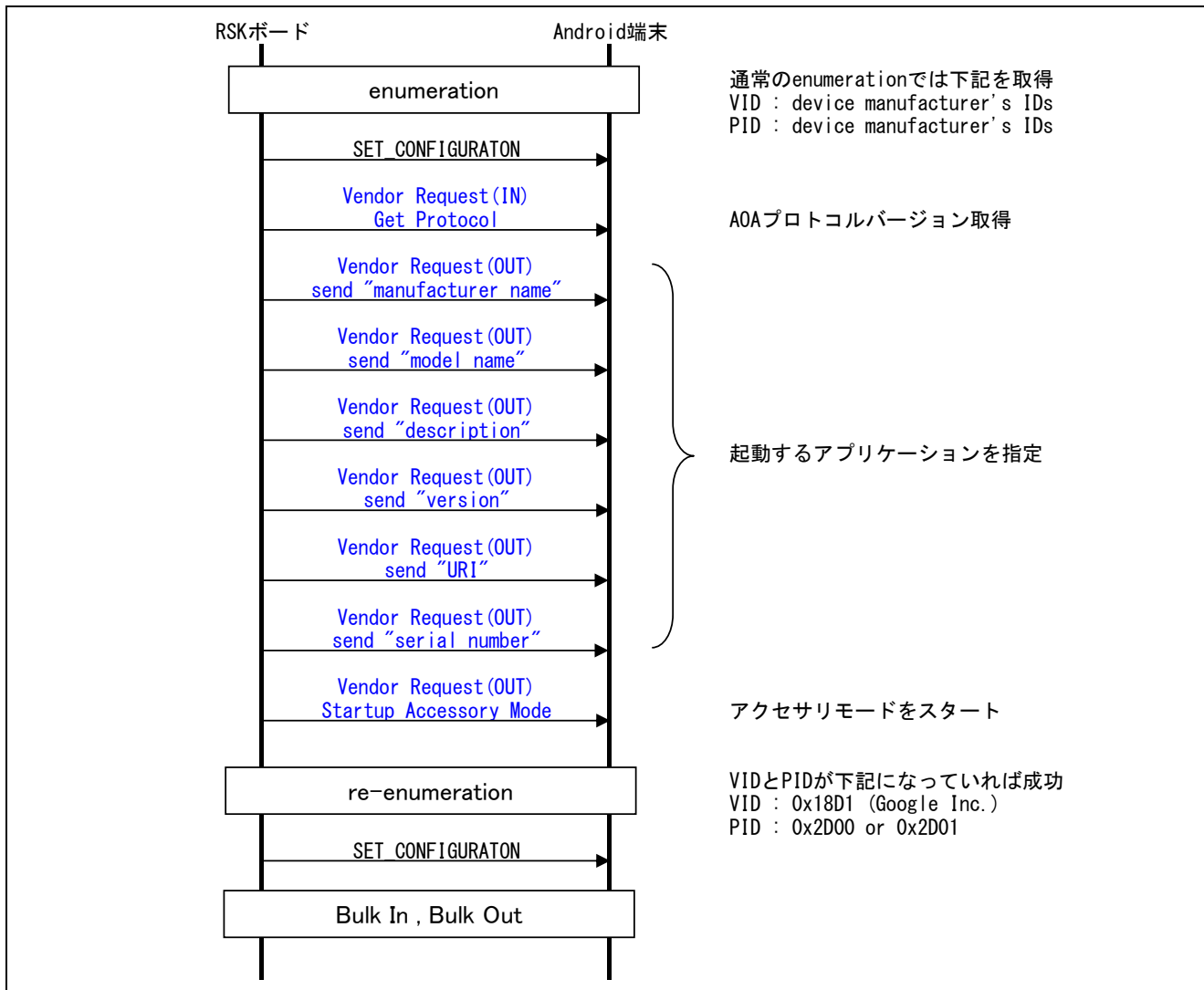


Figure 6.2 AOA Protocol のシーケンス

Android 端末が接続されると、通常のエニユメレーション処理を実行した後、AOA Protocol に至る所定の手順を経て、再エニユメレーションを実行します。その後にバルクを用いて Android 端末のアクセサリとして動作が可能になります。

6.7 サンプルアプリケーションの動作フロー

USB-BASIC-F/W の概略フローを Figure 6.3 に示します。

USB-BASIC-F/W は USB データ送受信の制御機能を実行するタスクを含んでいます。H/W の割り込みが発生すると USB-BASIC-F/W へメッセージとして通知されます。USB-BASIC-F/W は USB 割り込みハンドラからメッセージを受け取ると、割り込み要因を判別して適切な処理を実行します。

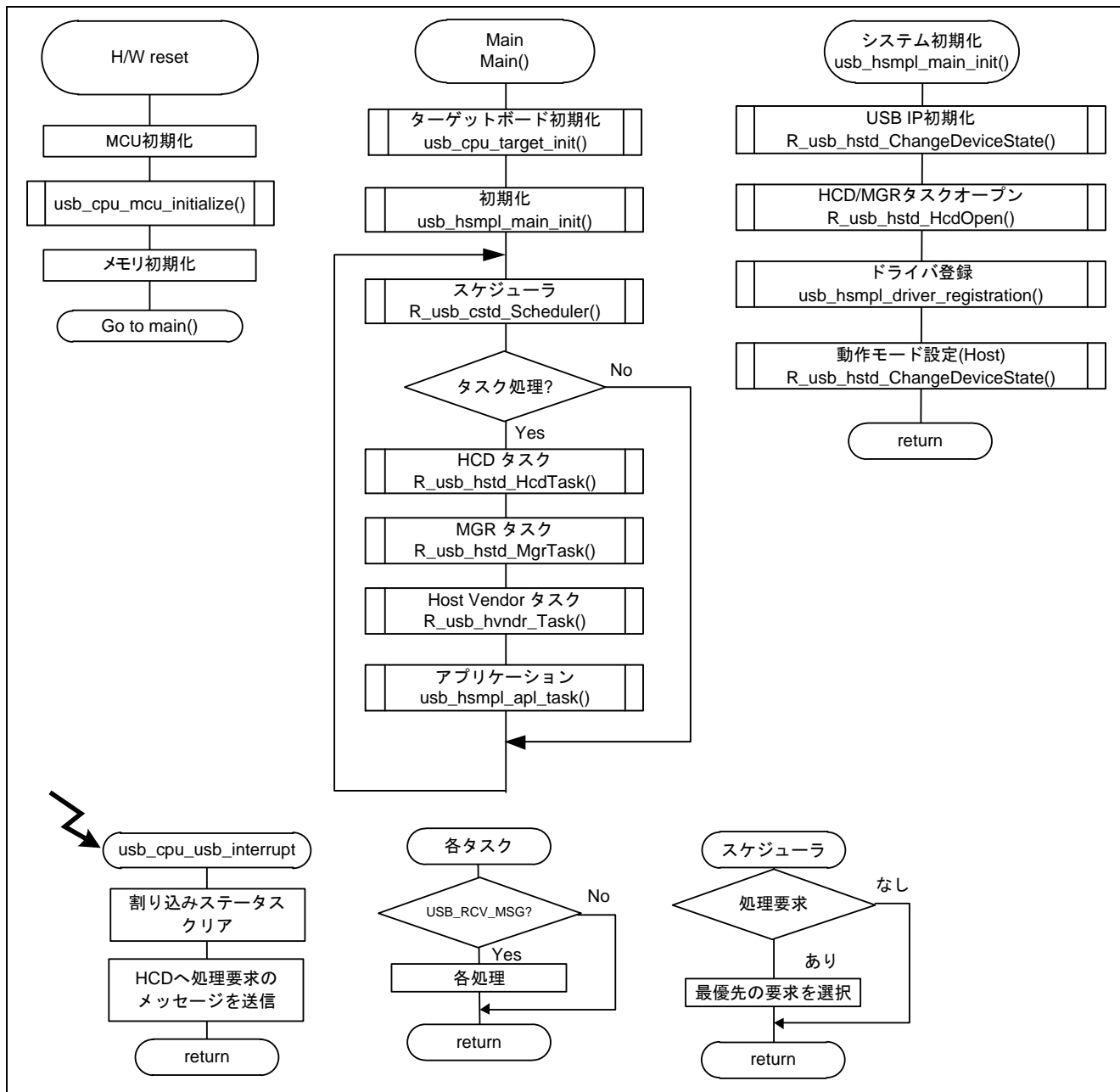


Figure 6.3 概略フロー

7. USB ホストベンダクラスドライバ (HVEN)

7.1 基本機能

本ソフトウェアは、Android Open Accessory の仕様に準拠しています。1.2 章を参照してください。

HVEN の基本機能を以下に示します。

1. Android 機器と AOA1.0/2.0 仕様の Protocol 通信を行う。
2. Android 機器とデータ送受信を行う。

7.2 HVEN タスク説明

本タスクは、USB_HVEN_MBX にメッセージを受け取り、メッセージ種別に従って処理を実行します。メッセージ種別による処理概要を Table 7-1 に示します。

Table 7-1 HVEN メッセージ種別

メッセージタイプ	処理概要	メッセージ送信元
USB_SMPL_CHECKREQUEST	エnumレーションシーケンスに従い、ストリングディスクリプタの取得やパイプ設定を行います。	R_usb_hvndr_ClassCheck() usb_hvndr_control_result() USB-BASIC-F/W がエnumレーション処理で接続デバイスの動作可否をコールバック関数で確認します。
USB_SMPL_AOA_PROTOCOL_RESULT_CHECK	バルク IN/OUT 転送開始データ転送完了時に APL にコールバック関数で通知する	usb_hvndr_AOA_Protocol_result() バルク IN/OUT 転送を使用した AOA Protocol 通信終了時に API 関数が実行される
USB_SMPL_AOA_HID_RESULT_CHECK	AOA Protocol による HID リポートデータ送信データ転送完了時に APL にコールバック関数で通知する	usb_hvndr_AOA_HID_result() バルク IN/OUT 転送を使用した AOA Protocol 通信終了時に API 関数が実行される(AOA2.0 モードの場合)

7.3 構造体

7.3.1 HVEN 構造体

Vendor AOA クラスのパラメータ構造体を Table 7-2~Table 7-4 に示します。

Table 7-2 aoa_param_t 構造体

型	メンバ名	説明
input_t	Input	RSK ボードからの入力・センサデータの格納構造体メンバ
id_t	dev_id	Android 機器の VID/PID の格納用構造体メンバ

Table 7-3 input_t 構造体

型	メンバ名	説明
int8_t	mouse_x	X 軸方向マウス移動量
int8_t	mouse_y	Y 軸方向マウス移動量
int8_t	temp_c	温度データ
uint16_t	lux	照度データ

Table 7-4 id_t 構造体

型	メンバ名	説明
uint8_t	v_hi	Android 機器ベンダーID 上位 8bit
uint8_t	v_lo	Android 機器ベンダーID 下位 8bit
uint8_t	p_hi	Android 機器プロダクト ID 上位 8bit
uint8_t	p_lo	Android 機器プロダクト ID 下位 8bit

7.3.2 ベンダクラスリクエスト

Table 7-5 に、対応しているベンダクラスリクエスト(bRequest)を示します。AOA プロトコルに従ったリクエストコードを発行します。詳細は Figure 6.2 を参照してください。

Table 7-5 ベンダクラスリクエスト

リクエスト	bRequest	説明
ACCESSORY_GET_PROTOCOL	51	AOA プロトコルバージョン取得
ACCESSORY_SEND_STRING	52	Accessory Identification コード送信
ACCESSORY_START	53	アクセサリモードスタート

また ACCESSORY_SEND_STRING を送付した際のインデックス(wIndex)を Table 7-6 に示します。

Table 7-6 Accessory Identification コードインデックス

リクエスト	wIndex	説明
ACCESSORY_STRING_MANUFACTURER	0	Manufacture name
ACCESSORY_STRING_MODEL	1	Model name
ACCESSORY_STRING_DESCRIPTION	2	Description
ACCESSORY_STRING_VERSION	3	Version
ACCESSORY_STRING_URI	4	URI
ACCESSORY_STRING_SERIAL	5	Serial number

7.4 HVEN API 一覧

HVEN API 一覧を Table 7-7 に示します。

Table 7-7 List of HVEN API Functions

関数名	説明	備考
R_usb_hvndr_Task	HVEN タスク処理	
R_usb_hvndr_ClassCheck	接続デバイス動作確認用メッセージ通知	
R_usb_hvndr_PipeRegistration	パイプコントロールレジスタ設定処理	

R_usb_hvndr_Task

HVEN タスク処理

形式

```
void R_usb_hvndr_Task(void)
```

引数

— —

戻り値

— —

解説

`g_usb_hvndr_task()`関数を呼び出します。

HVEN タスクはアプリから要求された処理を行い、アプリに処理結果を通知します。

補足

当該ループについては USB-BASIC-F/W アプリケーションノートを参照してください。

使用例

```
void usb_apl_task_switch(void)
{
    /* Main Loop */
    while( 1 )
    {
        if( R_usb_cstd_Scheduler() == USB_FLGSET )
        {
            R_usb_hstd_HcdTask();      /* HCD Task */
            R_usb_hstd_MgrTask();      /* MGR Task */
            R_usb_hvndr_Task();         /* HVEN Task */
            usb_hsmpl_apl_task();      /* HSMPL Task */
        }
    }
}
```

R_usb_hvndr_ClassCheck

ディスクリプタチェック処理

Format

void R_usb_hvndr_ClassCheck(uint8_t **table)

Argument

**table Address array of the device information table
[0] : Address of Device Descriptor
[1] : Address of Configuration Descriptor
[2] : Address of global variable that mean the Device Address

Return Value

— —

Description

本関数は接続デバイスの動作可否を判断する処理の実行を HVEN タスクに要求します。USB-BASIC-F/W が *classcheck* コールバックを実行した場合に本関数を呼び出してください。

HVEN タスクはペリフェラルデバイスのコンフィグレーションディスクリプタからエンドポイントディスクリプタを参照し、パイプ情報テーブル編集及び、使用するパイプ情報のチェックを行います。

Note

Example

```
USB_STATIC void usb_hhid_class_check(uint8_t **table)
{
    R_usb_hvndr_ClassCheck(table);
    usb_shven_smpl_devaddr = (usb_addr_t)(*table[2]);
}
```

R_usb_hvndr_PipeRegistration

パイプ設定処理

形式

void R_usb_hvndr_PipeRegistration(void)

引数

— —

戻り値

— —

解説

本関数はパイプ情報テーブルのアドレスフィールドを更新します。AOA Protocol 通信で使用されるパイプをハードウェアに設定します。

補足

1. USB-BASIC-F/W のアプリケーションノート（パイプ情報）を参照してください。
2. エンドポイントディスクリプタを参照できないパイプ情報テーブルのフィールドはあらかじめ設定しておいてください。

使用例

```
void usb_smp_task( void )
{
    :
    R_usb_hvndr_PipeRegistration();
    :
}
```

7.5 HVEN 関数 一覧

HVEN 内部で AOA に関わる主な関数一覧を Table 7-7 に示します。

Table 7-8 List of HVEN Internal Functions

関数名	説明	備考
usb_hvndr_AOA_Protocol	AOA Protocol 処理関数	
usb_hvndr_AOA_HID	HID リポートデータ送出処理関数	

usb_hvndr_AOA_Protocol

AOA Protocol 処理関数

形式

```
void          usb_hvndr_AOA_Protocol(usb_addr_t addr, usb_cb_t complete )
```

引数

```
addr          Device Address
complete      Callback function address
```

戻り値

```
—            —
```

解説

AOA Protocol フォーマットのデータを生成して、データ送信処理を行います。

HVEN タスクより繰り返し呼び出され、AOA Protocol の最初のシーケンス用データ生成/送信処理を行います。詳細は Figure 6.2 を参照してください。

補足

```
—
```

使用例

```
void          g_usb_hvndr_task(void)
{
    :
    usb_hvndr_AOA_Protocol(g_usb_hvndr_DevAddr, (usb_cb_t)&usb_hvndr_AOA_Protocol_
result);
    :
}
```

usb_hvndr_AOA_HID

HID 処理関数

形式

```
void          usb_hvndr_AOA_HID( usb_addr_t addr, usb_cb_t complete )
```

引数

addr	Device Address
complete	Callback function address

戻り値

—

解説

AOA Protocol に従い、HID Accessory 処理を行います。

HVEN タスクより繰り返し呼び出され、HID リポートディスクリプタ及び HID リポートデータの送信を行います。

補足

本関数は AOA2.0 モードで使用します。

使用例

```
void          g_usb_hvndr_task(void)
{
    :
    usb_hvndr_AOA_HID( g_usb_hvndr_DevAddr, (usb_cb_t)&usb_hvndr_AOA_HID_result );
    :
}
```

8. 制限事項

Vendor AOA には、以下の制限事項があります。

1. 本サンプルでは audio モードには対応していません。
2. Vendor AOA ドライバに接続可能なデバイスは1つだけです。2つ以上のデバイスを同時に接続しないでください。

9. e² studio 用プロジェクトのセットアップ

(1). e² studio を起動してください。

※ はじめてe² studio を起動する場合、Workspace Launcher ダイアログが表示されますので、プロジェクトを格納するためのフォルダを指定してください。

(2). [ファイル] → [インポート]を選択してください。インポートの選択ダイアログが表示されます。

(3). インポートの選択画面で、[既存プロジェクトをワークスペースへ]を選択してください。

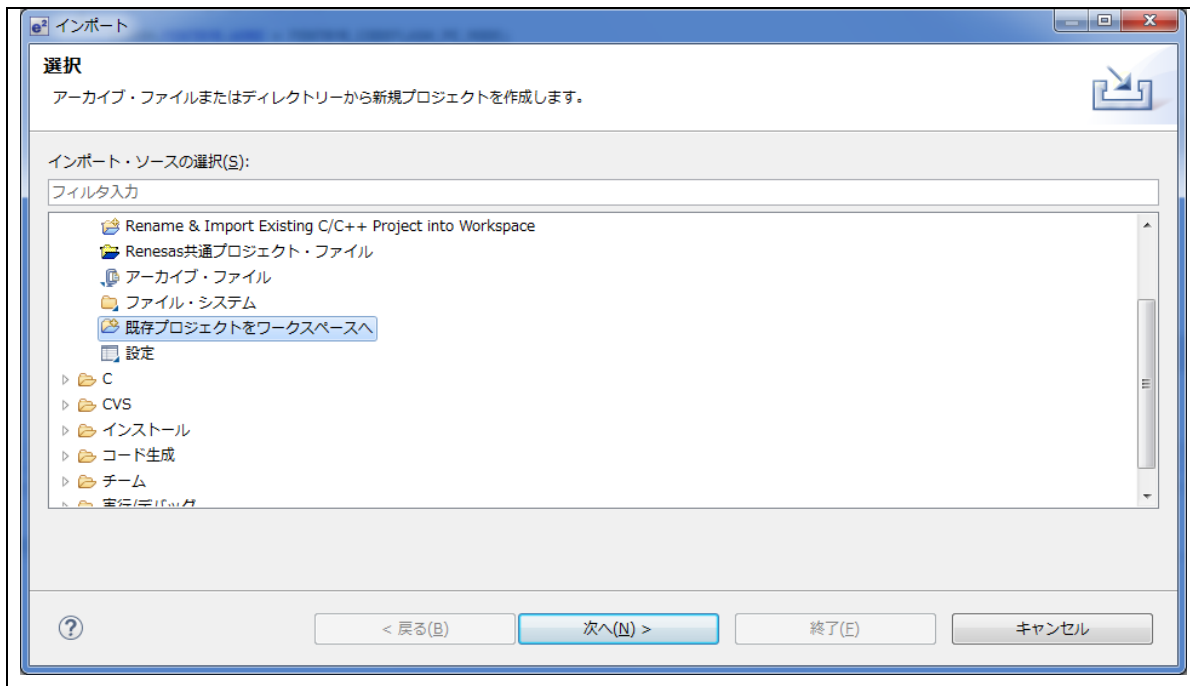


Figure 9-1 インポートの選択

(4). [ルートディレクトリの選択]の[参照]ボタンを押下して、「.cproject」(プロジェクトファイル)が格納されたフォルダを選択して下さい。

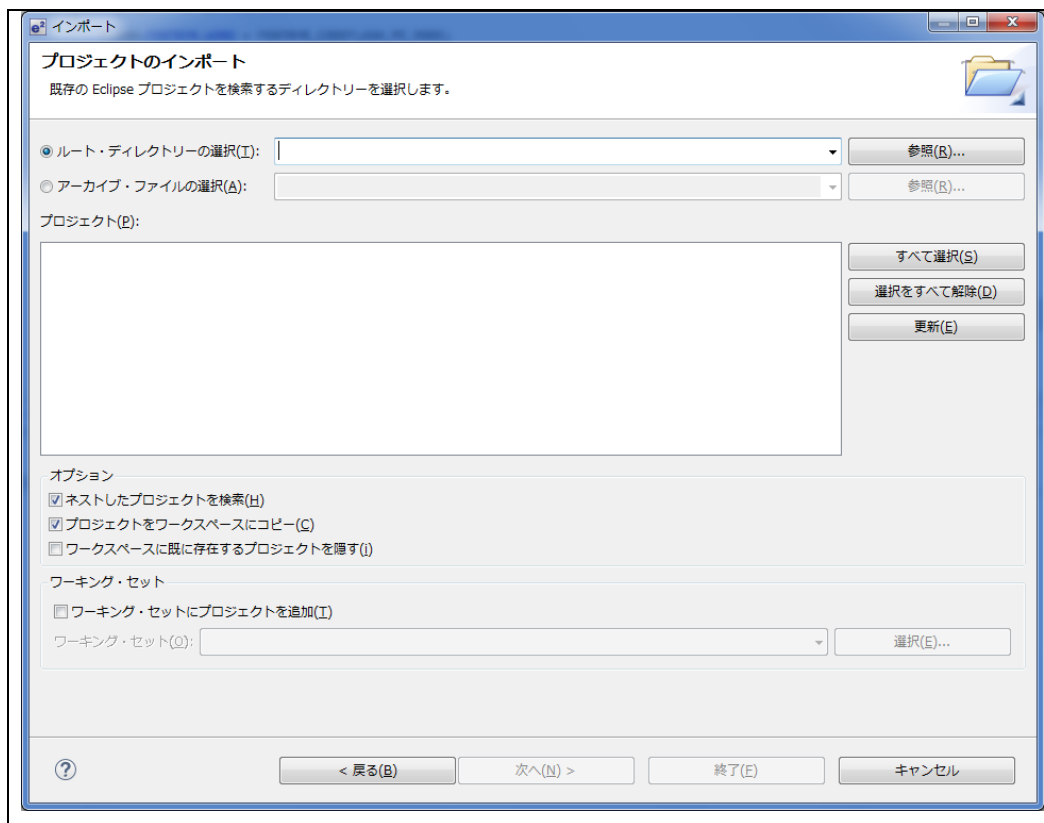


Figure 9-2 プロジェクトのインポート画面

(5). [終了]をクリックして下さい。

プロジェクトのワークスペースへのインポートが完了します。

10. e² studio 用プロジェクトを CS+で使用する場合

本プロジェクトは、統合環境 e² studio で作成されています。本プロジェクトを CS+で動作させる場合は、下記の手順を行ってください。

[Note]

rpcs ファイルは、workspace\CCRL(MCU 名)フォルダ内に用意されています。

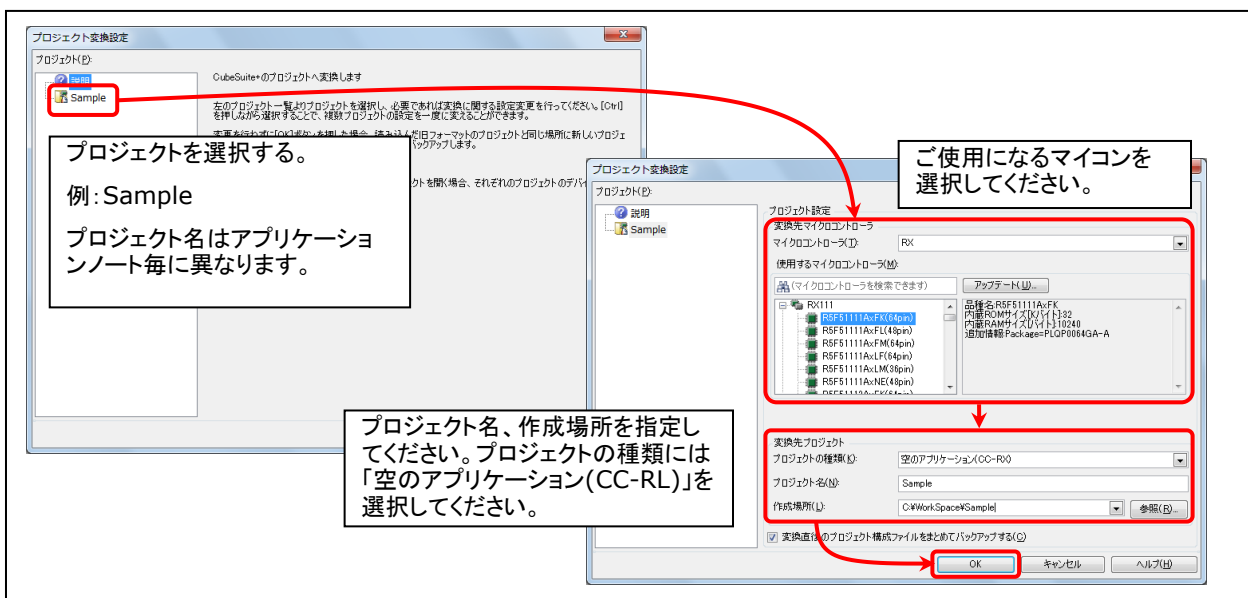
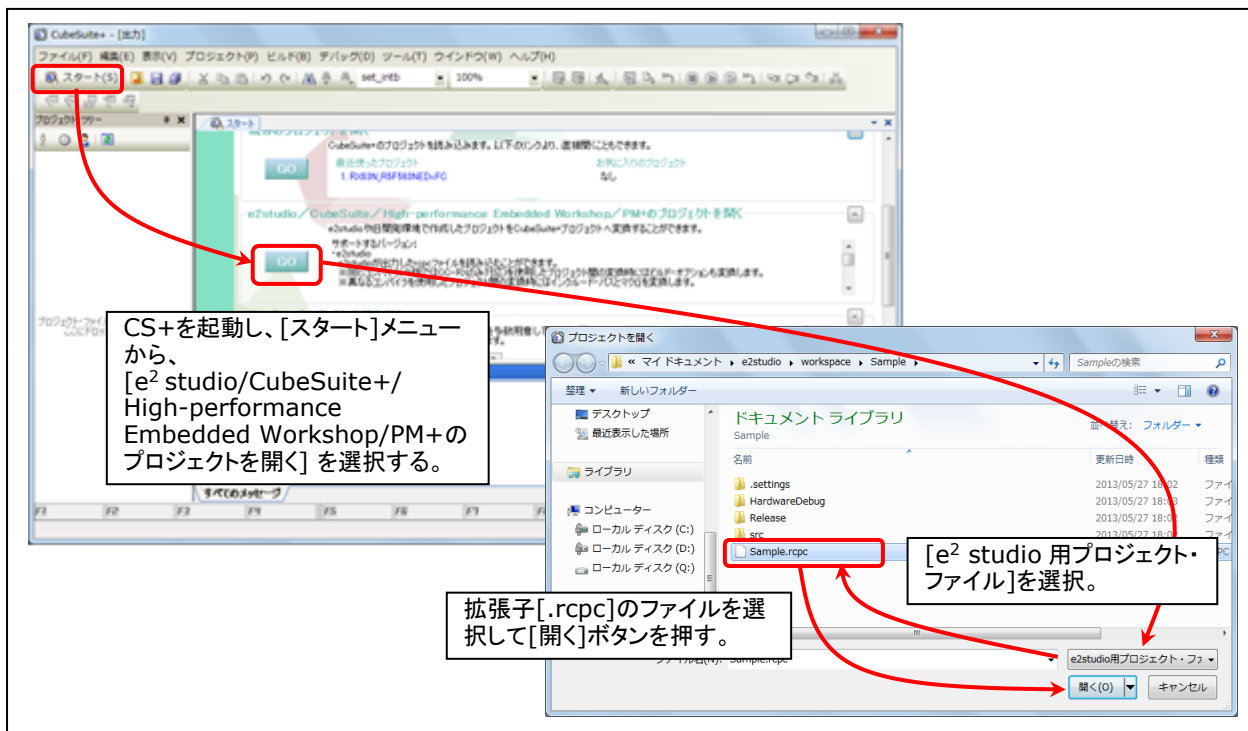


Figure 10-1 e² studio 用プロジェクトの CS+読み込み方法

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問い合わせ先

<http://japan.renesas.com/contact/>

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.02.17	—	初版発行
1.01	2015.03.16	—	“reference” フォルダのドキュメントを更新。
1.02	2016.01.18	—	Technical Update(発行番号: TN-RL*-A055A/J)に対応しました。
1.03	2016.03.28	—	CC-RL コンパイラをサポートしました。

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. 未使用端子の処理

【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】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. リザーブアドレス（予約領域）のアクセス禁止

【注意】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、将来の機能拡張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。

4. クロックについて

【注意】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。

5. 製品間の相違について

【注意】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くと、内部 ROM、レイアウトパターンの相違などにより、電气的特性の範囲で、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。型名が違う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。お客様の機器・システムの設計において、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には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 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に関し、当社は、何らの責任を負うものではありません。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. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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。
5. 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）、交通用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じて、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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。
7. 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、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、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。
8. 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。お客様が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。
9. 本資料に記載されている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。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 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。

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とその総株主の議決権の過半数を直接または間接に保有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。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/contact/>