



まい USB 取扱説明書

ルネサス エレクトロニクス社 マイコン向けソフトウェア製品

-本書を必ずよく読み、ご理解された上でご利用ください

株式会社 **北斗電子**
REV.1.3.0.0

注意事項	1
安全上のご注意	2
1. はじめに	4
2. 製品内容	5
3. 概要	6
4. 製品仕様	7
4.1. USB ホスト(Windows 側)	7
4.2. USB ファンクション(ターゲット基板側)	7
5. 「まい USB」を組み込む方法(開発編)	8
5.1. USB ホスト(Windows 側)	8
5.2. USB ファンクション(ターゲット基板側)	8
5.2.1. 選択型割り込みに関して(RX マイコン)	9
5.2.2. RA マイコンでの割り込み定義の追加	10
5.2.1. RA マイコンでのライブラリのリンク	13
6. 「まい USB」を組み込む方法(運用編)	15
7. 「まい USB」を組み込む方法(ドライバの導入)	16
7.1. Windows10 の場合	16
8. API 関数(ユーザインタフェース)	23
8.1. USB ホスト(Windows 側)	23
8.1.1. API 関数一覧	23
8.1.2. API 関数詳細	24
8.2. USB ファンクション(ターゲット基板側)	28
8.2.1. API 関数一覧	28
8.2.2. API 関数詳細	29
9. ターゲット基板別制約	37
10. 付録	39
10.1. サンプルアプリケーションについて	39
11. 注意事項	41
11.1. 64bit 版の注意事項	41
11.2. 「まい USB」と「まい USB Lite」の違いについて	41
12. 著作権・免責に関して	42
お問合せ窓口	42

注意事項

本書を必ずよく読み、ご理解された上でご利用ください

【ご利用にあたって】

1. 本製品をご利用になる前には必ず取扱説明書をよく読んで下さい。また、本書は必ず保管し、使用上不明な点がある場合は再読し、よく理解して使用して下さい。
2. 本書は株式会社北斗電子製マイコンボードの使用方法について説明するものであり、ユーザシステムは対象ではありません。
3. 本書及び製品は著作権及び工業所有権によって保護されており、全ての権利は弊社に帰属します。本書の無断複製・複製・転載はできません。
4. 弊社のマイコンボードの仕様は全て使用しているマイコンの仕様に準じております。マイコンの仕様に関しましては製造元にお問い合わせ下さい。弊社製品のデザイン・機能・仕様は性能や安全性の向上を目的に、予告無しに変更することがあります。また価格を変更する場合や本書の図は実物と異なる場合もありますので、御了承下さい。
5. 本製品のご使用にあたっては、十分に評価の上ご使用下さい。
6. 未実装の部品に関してはサポート対象外です。お客様の責任においてご使用下さい。

【限定保証】

1. 弊社は本製品が頒布されているご利用条件に従って製造されたもので、本書に記載された動作を保証致します。
2. 本製品の保証期間は購入戴いた日から1年間です。

【保証規定】

保証期間内でも次のような場合は保証対象外となり有料修理となります

1. 火災・地震・第三者による行為その他の事故により本製品に不具合が生じた場合
2. お客様の故意・過失・誤用・異常な条件でのご利用で本製品に不具合が生じた場合
3. 本製品及び付属品のご利用方法に起因した損害が発生した場合
4. お客様によって本製品及び付属品へ改造・修理がなされた場合

【免責事項】

弊社は特定の目的・用途に関する保証や特許権侵害に対する保証等、本保証条件以外のものは明示・黙示に拘わらず一切の保証は致し兼ねます。また、直接的・間接的損害金もしくは欠陥製品や製品の使用方法に起因する損失金・費用には一切責任を負いません。損害の発生についてあらかじめ知らされていた場合でも保証は致し兼ねます。

ただし、明示的に保証責任または担保責任を負う場合でも、その理由のいかんを問わず、累積的な損害賠償責任は、弊社が受領した対価を上限とします。本製品は「現状」で販売されているものであり、使用に際してはお客様がその結果に一切の責任を負うものとします。弊社は使用または使用不能から生ずる損害に関して一切責任を負いません。

保証は最初の購入者であるお客様ご本人にのみ適用され、お客様が転売された第三者には適用されません。よって転売による第三者またはその為になすお客様からのいかなる請求についても責任を負いません。

本製品を使った二次製品の保証は致し兼ねます。

安全上のご注意

製品を安全にお使いいただくための項目を次のように記載しています。絵表示の意味をよく理解した上でお読み下さい。

表記の意味



取扱を誤った場合、人が死亡または重傷を負う危険が切迫して生じる可能性がある事が想定される



取扱を誤った場合、人が軽傷を負う可能性又は、物的損害のみを引き起こすが可能性がある事が想定される

絵記号の意味

	一般指示 使用者に対して指示に基づく行為を強制するものを示します		一般禁止 一般的な禁止事項を示します
	電源プラグを抜く 使用者に対して電源プラグをコンセントから抜くように指示します		一般注意 一般的な注意を示しています

警告



以下の警告に反する操作をされた場合、本製品及びユーザシステムの破壊・発煙・発火の危険があります。マイコン内蔵プログラムを破壊する場合があります。

1. 本製品及びユーザシステムに電源が入ったままケーブルの抜き差しを行わないでください。
2. 本製品及びユーザシステムに電源が入ったままで、ユーザシステム上に実装されたマイコンまたはIC等の抜き差しを行わないでください。
3. 本製品及びユーザシステムは規定の電圧範囲でご利用ください。
4. 本製品及びユーザシステムは、コネクタのピン番号及びユーザシステム上のマイコンとの接続を確認の上正しく扱ってください。



発煙・異音・異臭にお気づきの際はすぐに使用を中止してください。

電源がある場合は電源を切って、コンセントから電源プラグを抜いてください。そのままご使用すると火災や感電の原因になります。

注意



以下のことをされると故障の原因となる場合があります。

1. 静電気が流れ、部品が破壊される恐れがありますので、ボード製品のコネクタ部分や部品面には直接手を触れないでください。
2. 次の様な場所での使用、保管をしないでください。
ホコリが多い場所、長時間直射日光が当たる場所、不安定な場所、衝撃や振動が加わる場所、落下の可能性がある場所、水分や湿気の多い場所、磁気を発するものの近く
3. 落としたり、衝撃を与えたり、重いものを乗せないでください。
4. 製品の上に水などの液体や、クリップなどの金属を置かないでください。
5. 製品の傍で飲食や喫煙をしないでください。



ボード製品では、裏面にハンダ付けの跡があり、尖っている場合があります。

取り付け、取り外しの際は製品の両端を持ってください。裏面のハンダ付け跡で、誤って手など怪我をする場合があります。



CD メディア、フロッピーディスク付属の製品では、故障に備えてバックアップ（複製）をお取りください。

製品をご使用中にデータなどが消失した場合、データなどの保証は一切致しかねます。



アクセスランプがある製品では、アクセスランプ点灯中に電源の切断を行わないでください。

製品の故障の原因や、データの消失の恐れがあります。



本製品は、医療、航空宇宙、原子力、輸送などの人命に関わる機器やシステム及び高度な信頼性を必要とする設備や機器などに用いられる事を目的として、設計及び製造されておりません。

医療、航空宇宙、原子力、輸送などの設備や機器、システムなどに本製品を使用され、本製品の故障により、人身や火災事故、社会的な損害などが生じても、弊社では責任を負いかねます。お客様ご自身にて対策を期されるようご注意ください。

1. はじめに

ちょっとした周辺機器をつなごうとすると、シリアル・ポートなどのインターフェースを使用することが一般的でした。

最近のパソコンでは、シリアル・ポートが存在しないことも少なくなく、USB ポートを利用することが主流になってきました。ターゲット機器ではシリアル・ポートがお手軽であるため、USB シリアル変換を使っています。

しかし、USB シリアル変換を使用するとユーザに仮想 COM ポートを選択させることが必要となる上、通信速度の低いものになってしまいます。

USB にはシリアルよりはるかに速い伝送速度を実現できるといったメリットがあるばかりか、USB 電源を活用することによりターゲットの電源周りをシンプルにできるメリットがあります。

そういったメリットから、ターゲット機器を USB 化したいと思われる方々が、多くなってきています。しかし、USB 機器を開発しようとする、具体的な作成事例などがあまり存在しないため、USB 規格、Windows ドライバー及びターゲット基板について十分に理解するのに多くの時間を要します。

「まい USB」はそういった難しい知識をアプリケーションに意識させることなく、まるでシリアル通信かのように USB 通信を実現するお手軽なライブラリソフトウェアです。

この取扱説明書はソフトウェアの使用方法について説明しています。

正しくご使用していただくために、この取扱説明書をよくお読みください

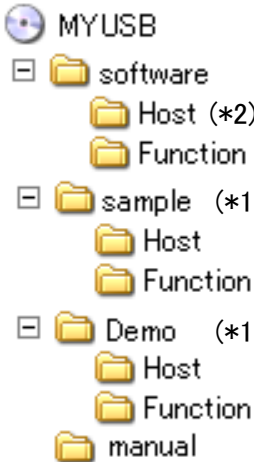
2. 製品内容

次のものが全てそろっているかどうかをご確認ください。

- CD.....1 枚
- ソフトウェア使用許諾書.....1 枚

※「myusb_XXXX.拡張子」の XXXX は製品 ID(PID)を意味しています。

CD 内容



The diagram shows a tree structure for the CD content. At the top is a CD icon labeled 'MYUSB'. Below it are three main folders: 'software', 'sample (*1)', and 'Demo (*1)'. The 'software' folder contains 'Host (*2)' and 'Function'. The 'sample (*1)' folder contains 'Host' and 'Function'. The 'Demo (*1)' folder contains 'Host' and 'Function'. A bracket on the right groups the 'software', 'sample (*1)', and 'Demo (*1)' folders under the label 'ソフト一式'. Below the 'Demo (*1)' folder is a 'manual' folder, which is grouped under the label '取扱説明書(本誌)'. At the bottom of the diagram is a bullet point: '・ ソフトウェア使用許諾書(PDF形式)'.

- ・ ソフトウェア使用許諾書(PDF形式)

(*1)製品によっては、どちらか一方のご用意となります。

(*2)

¥software¥Host¥Windows10_64bit ドライバー Windows10(Ver1607~21H2)向け

3. 概要

「まいUSB」は、ホスト(Windows¹)とファンクション(ターゲット基板)で行うUSB通信を自動的に行うソフトウェアライブラリです。

「まいUSB」のAPI関数をコールするだけで、USB通信を簡単に行うことができます。

API関数の詳細については、8章のAPI関数をご参照ください。

「まいUSB」を使用したUSB通信の動作概念図を図 1に示します。

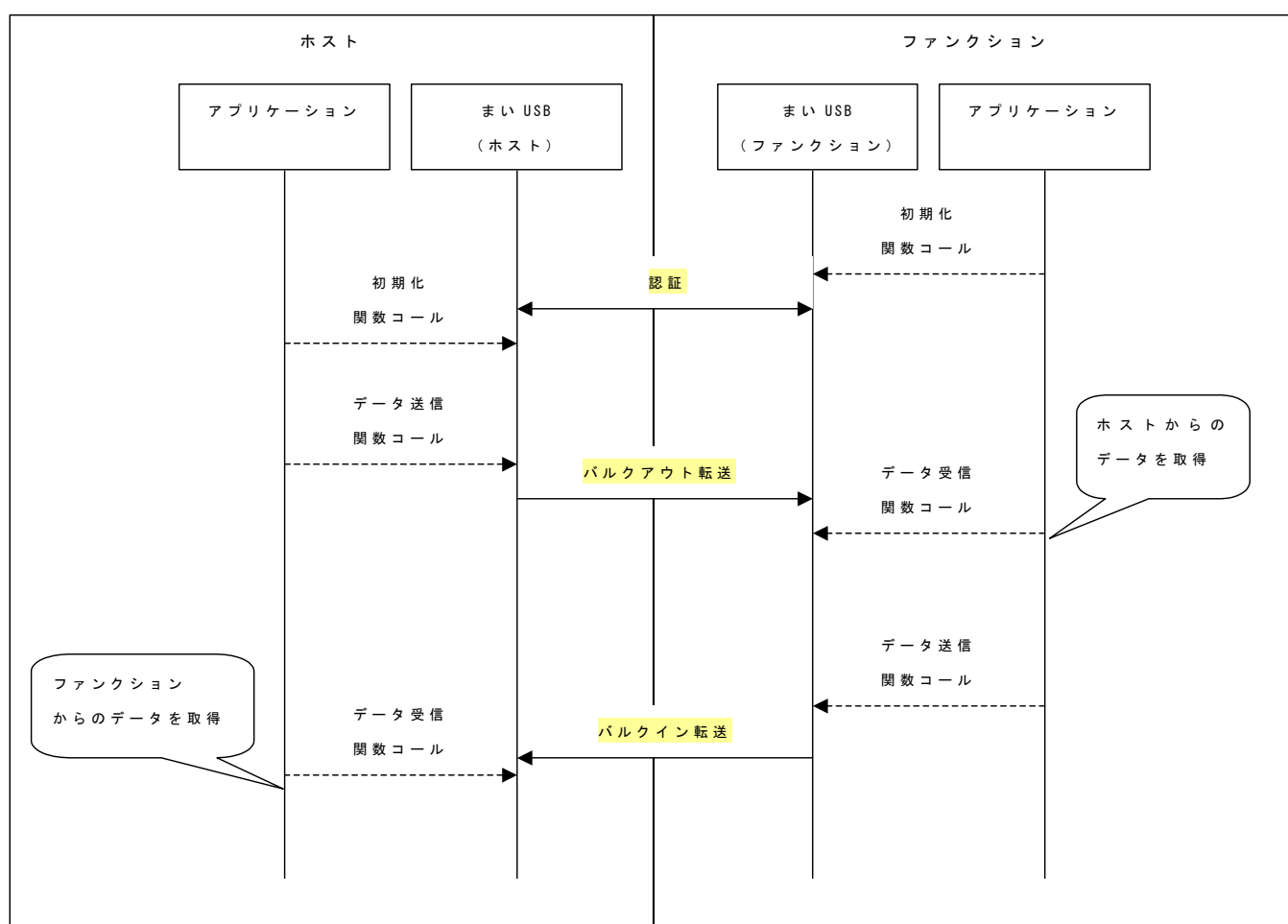


図 1「まいUSB」動作概念図

¹ WindowsOS は、Windows10 の 64bitOS にて動作いたします

4. 製品仕様

4.1. USB ホスト(Windows 側)

対応 OS : Windows10
64 ビットオペレーティングシステム

2022 年 2 月以降の購入分に関しましては、Windows10 1607(RS1)~21H2 に対応しております。
Windows7/Windows8.1 向けのドライバー(sys)ファイルは対応を終了致しました。

まい USB64 のドライバーファイル、MYUSB_nnnn.sys (nnnn は、PID 番号が入ります)は、64bitOS のみインストール可能です。

アプリケーションソフトは、x86(32bit アプリケーション)対応となります。
(アプリケーション向けの DLL は、32bit DLL です)

4.2. USB ファンクション(ターゲット基板側)

「まい USB」では以下の領域名で設定しており、以下のサイズを必要とします。
ご使用の際には、下記領域名を設定してください。

セクション名	領域名	サイズ	備考
プログラム領域	PMYUSB*	約 6KB	
定数領域	CMYUSB*	約 256Byte	
未初期化データ領域	BMYUSB*	約 6KB	通信バッファにて固定領域で約 5KB 使用しておりますが、R8C 用では HEAP 領域に任意設定可能にしております。(固定領域は 128Byte)
スタック	-	約 64Byte	

リンクに与えるセクション名として、上記領域名を追加してください。

※RA 向けではセクション名の設定は不要です

5. 「まい USB」を組み込む方法(開発編)

5.1. USB ホスト(Windows 側)

- ・ API 関数をご使用になられるアプリケーションファイルに、「usb_host_dll.h」をインクルードするように実装してください。
- ・ 「myusb_XXXX.dll」をロードし、8.1.2 章に記載しています API 関数のポインタを取得するように実装してください。
- ・ アプリケーションの仕様に合わせて上記で取得した関数ポインタをご使用ください。
- ・ ビルドを行う際には、CD に入っています「usb_host_dll.h」を開発環境(インクルードパス上)にコピーしてください。

5.2. USB ファンクション(ターゲット基板側)

- ・ API 関数をご使用になられるアプリケーションのファイルに、「usb_function_api.h」をインクルードするように実装してください。
- ・ アプリケーションの仕様に合わせて 8.2.2 章に記載しています API 関数をご使用ください。
- ・ コンパイルを行う際には、CD に入っています「usb_function_api.h」を開発環境(インクルードパス上)にコピーしてください。
- ・ リンクを行う際には、CD に入っています「myusb_XXXX.拡張子」を開発環境(入カライブラリファイル指定パス上)にコピーしてください。

お使いのコンパイル/リンク環境によって以下の形式のものをご利用下さい。

- ・SYSROF 形式: myusb_XXXX_sysrof.lib
- ・バンク切換なしの ELF 形式: myusb_XXXX_elf.lib
- ・バンク切換ありの ELF 形式: myusb_XXXX_resbank.lib(バンク切換に対応したマイコンのみ)
- ・アーカイバ: myusb_XXXX.a
- ・R8C 用ライブラリ: myusb_XXXX.lib
- ・RX 向け LittleEndian: myusb_XXXX_USBY_little.lib
- ・RX 向け BigEndian: myusb_XXXX_USBY_big.lib
- ・RA 向け: libRAzzz_MYUSB_LIB_XXXX.a

※XXXX: PID 番号が入ります

※Y: USB ポートが複数あるマイコンの場合、0, A 等が入ります

※zzz: ターゲットマイコン名(RA6M2 等が入ります)

5.2.1. 選択型割り込みに関して(RX マイコン)

まい USB では、USBIO の割り込みを使用します。

RX64M, RX71M, RX72M, RX72N, RX66N のマイコンでは、USBIO の割り込みは選択型割り込み B に割り当てられており、割り込み番号 USBIO(割り込み要因番号 62)を、割り込み番号 128~207 のいずれかに割付を行う必要があります。

まい USB では、USBIO を割り込み番号 185 に割り当てを行っています。(スマートコンフィグレータのデフォルト値です。)

ユーザプログラムで、選択型割り込みを使用しない場合は、特に割り込み番号に関して設定等は不要です。

ユーザプログラムで、選択型割り込みを使用する場合、

(1)

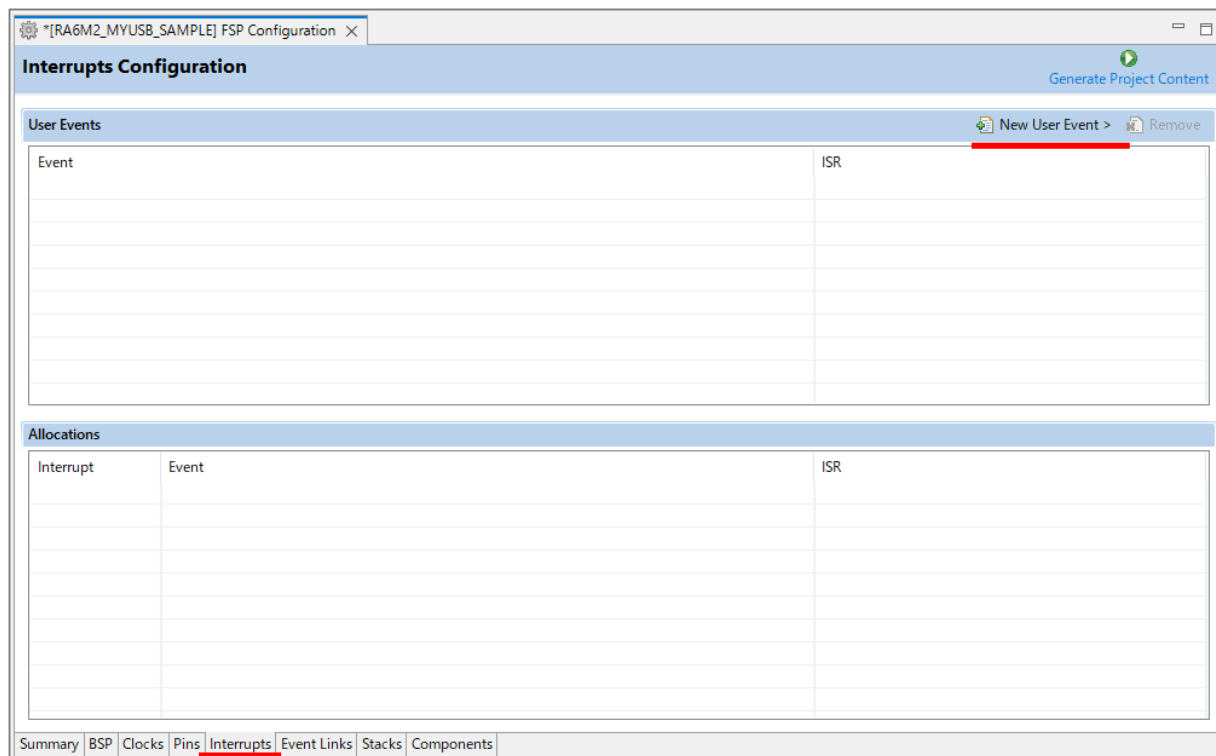
- ・185 番は未使用とする
- ・ユーザプログラムの選択型割り込みの設定を行う
- ・選択型割り込み設定レジスタの書き込み保護を行わない状態で MyUsbFunctionInit() を実行する
→MyUsbFunctionInit() 内で、USBIO の割り当てと選択型割り込み設定レジスタの書き込み保護を実行します
→選択型割り込みの設定(ICU.SLIBXRn, ICU.SLIBRn, ICU.SLIARn レジスタ値の設定)を MyUsbFunctionInit()
実行前に行ってください

(2)

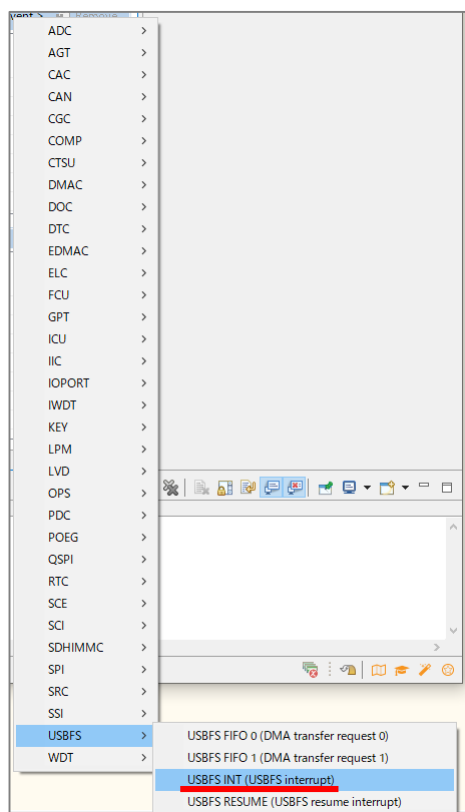
- ・ICU.SLIBR185.BIT.SLI = 62; で割り込み要因 62 番を、割り込み番号 185 に設定する
- ・他、ユーザプログラムの選択型割り込みの設定を行う
- ・選択型割り込み設定レジスタの書き込み保護を行う(ICU.SLIPRCR.BIT.WPRC = 1;)
- ・MyUsbFunctionInit()を実行

上記(1)(2)のいずれかとしてください。

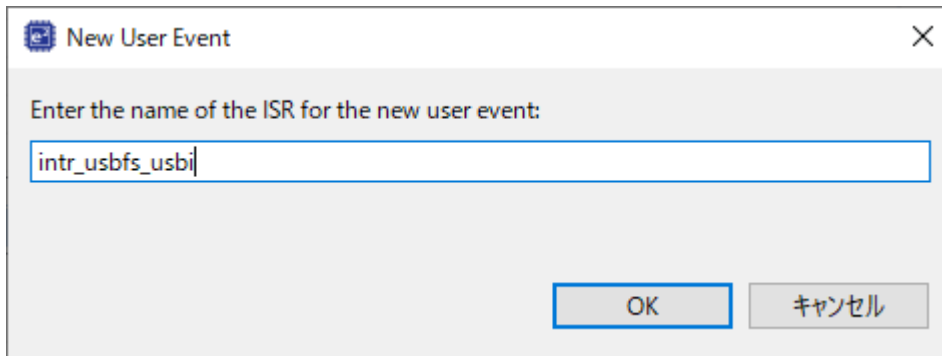
5.2.2. RA マイコンでの割り込み定義の追加



FSP の FSP Configuration の Interrupts タブ New User Event



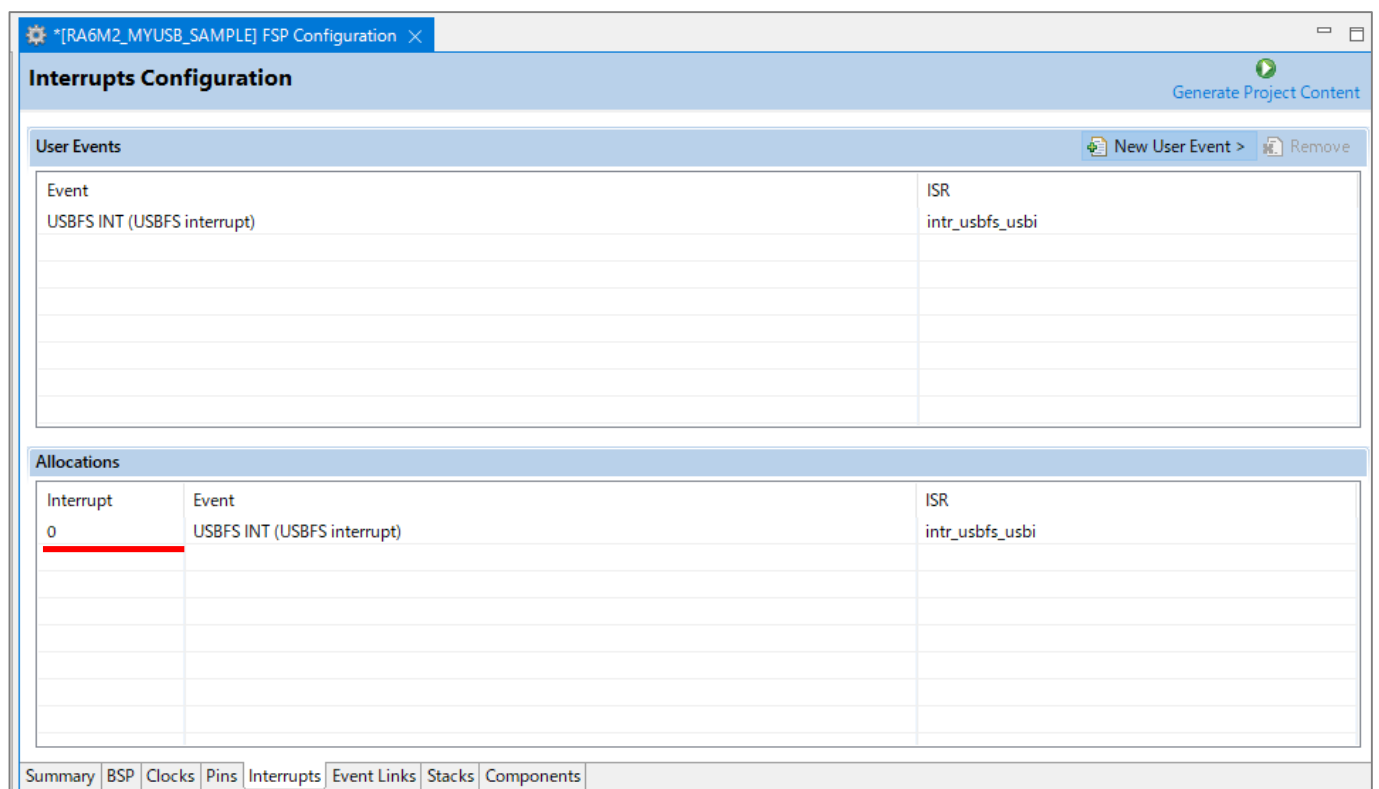
USBFS - USBFS INT を選択



new user event の名称(割り込み関数名)を入力するダイアログがでますので、

intr_usbfs_usbi

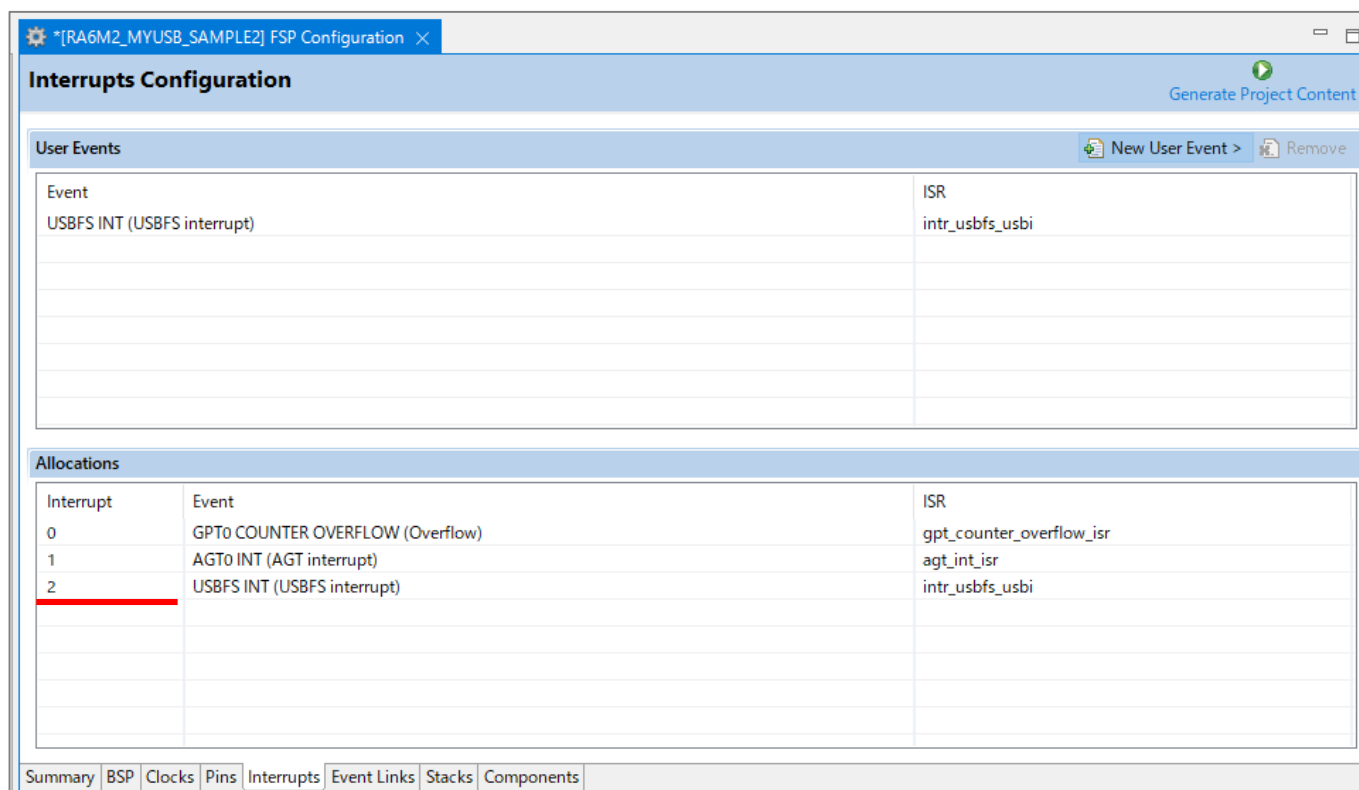
という割り込み関数名を定義してください。



追加後は上記の様になります。

Interrupt の欄の番号が割り込み番号となります。初期化関数呼び出し時(MyUsbFunctionInit)の引数として、この番号を与える必要があります。

※stacks で AGT0, GPT0 を追加した場合



FSP の stack 追加や他の割り込み番号追加で intr_usbfs_usbi として定義した割り込み番号は変わります。所定の stack 追加や割り込み定義を行い最終的に、intr_usbfs_usbi の割り込み番号がいくつになったかを確認して、

```
MyUsbFunctionInit( 2 ); // まい USB 初期化
```

初期化関数(MyUsbFunctionInit)を呼び出す際の引数として指定してください。

5.2.1. RA マイコンでのライブラリのリンク

まい USB のライブラリは、スタティックリンクライブラリ

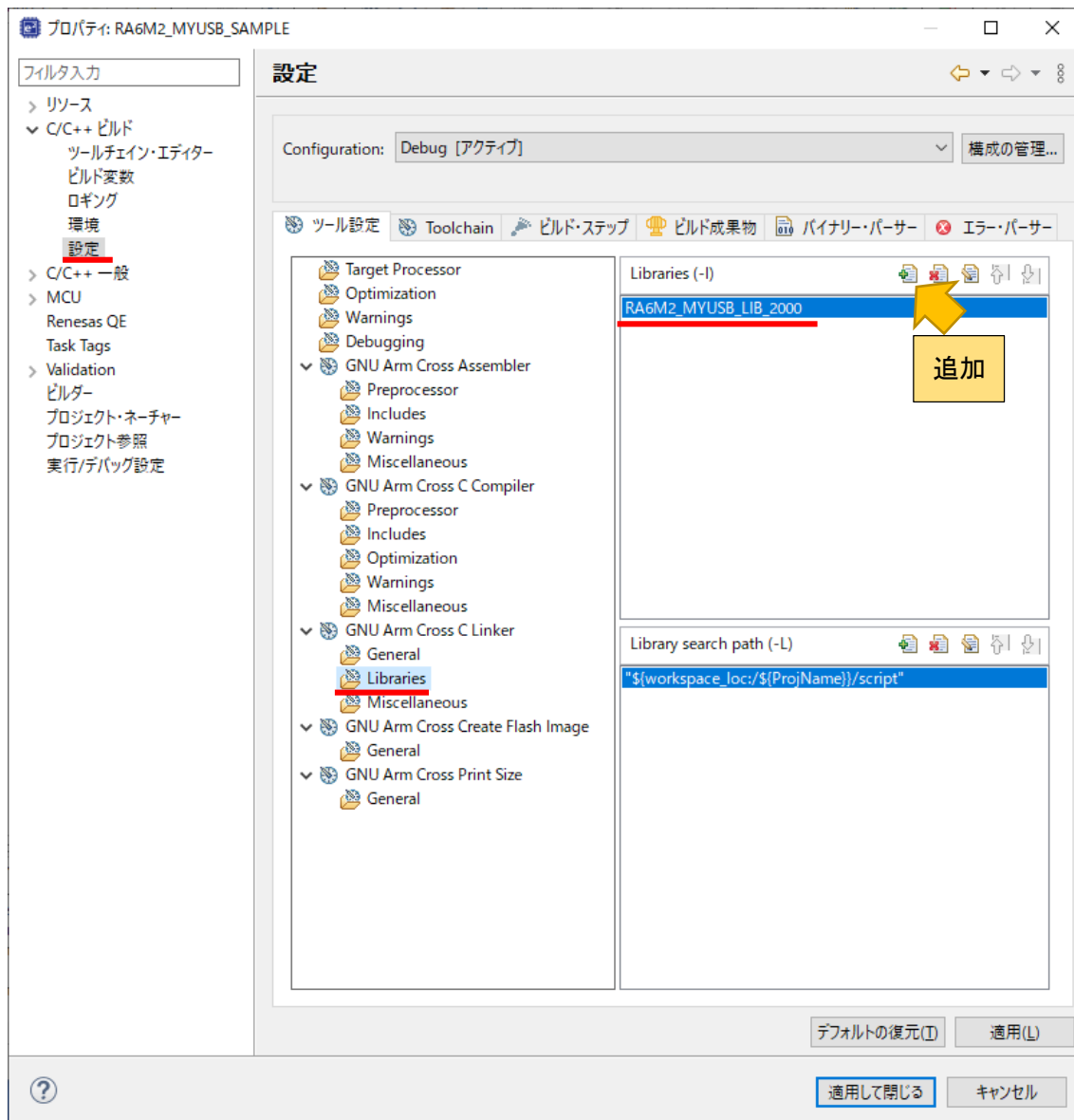
libRAzzz_MYUSB_LIB_XXXX.a

(拡張子.a)として提供されます

zzz:マイコン名

XXXX:PID 番号

(RA6M2 向け PID=2000 の場合、libRA6M2_MYUSB_LIB_2000.a)



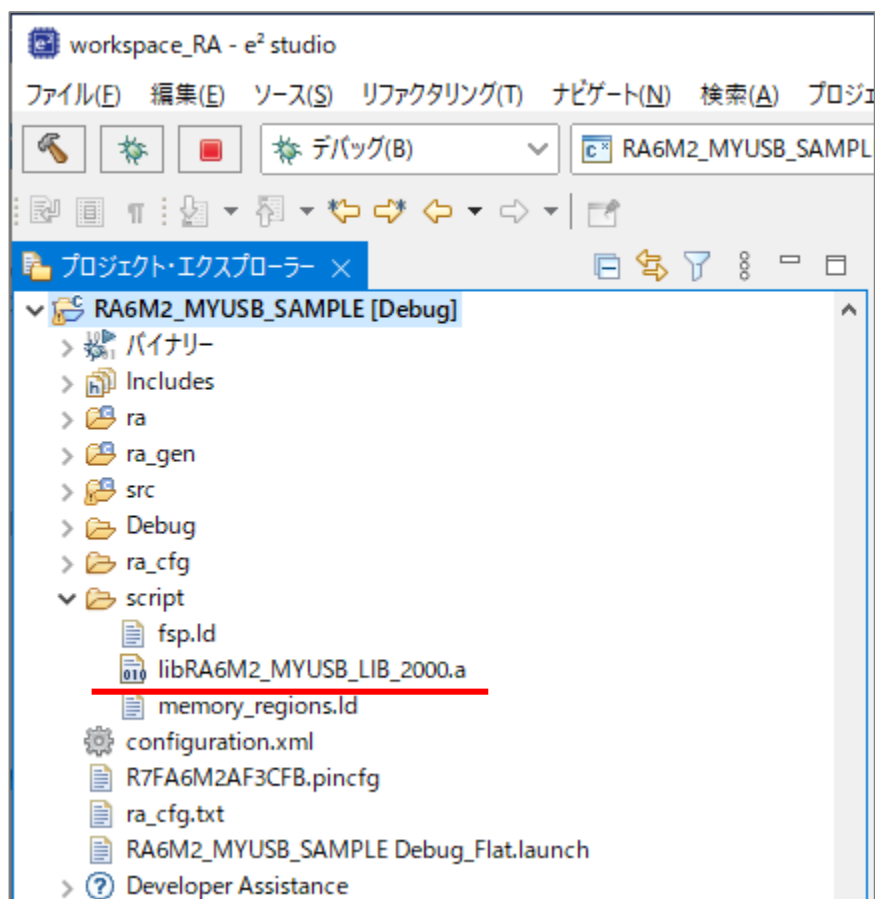
C/C++ビルド - 設定 - GNU Arm Cross C Linker - Libraries

Libraries 欄に

RAzzz_MYUSB_LIB_XXXX

を追加(ファイル名の先頭の lib と末尾の.a を除いたもの)

libRAzzz_MYUSB_LIB_XXXX.a ファイルを、プロジェクトフォルダの scripts 内にコピー



※他のフォルダ内にライブラリファイルを配置した場合、Library search path(-L)オプションを指定してください

6. 「まい USB」を組み込む方法(運用編)

- (1) 5.2 章にて開発しました USB ファンクションソフトウェアをお手持ちの Writer でターゲット基板に書き込んでください。
- (2) ホストマシンの電源を投入し、システムを起動してください。
- (3) ホストマシンとターゲット基板を USB ケーブルで接続し、ターゲット基板の電源を投入してください。
※初回接続時には、ホストマシンにてドライバーのインストールが促されますので、5 章を参照の上、ドライバーをインストールしてください。
- (4) 5.1 章にて開発しました USB ホストアプリケーションと CD に入っています「myusb_XXXX.dll」を同じフォルダに格納してください。
- (5) USB ホストアプリケーションを起動してください。
- (6) 完了

7. 「まい USB」を組み込む方法(ドライバの導入)

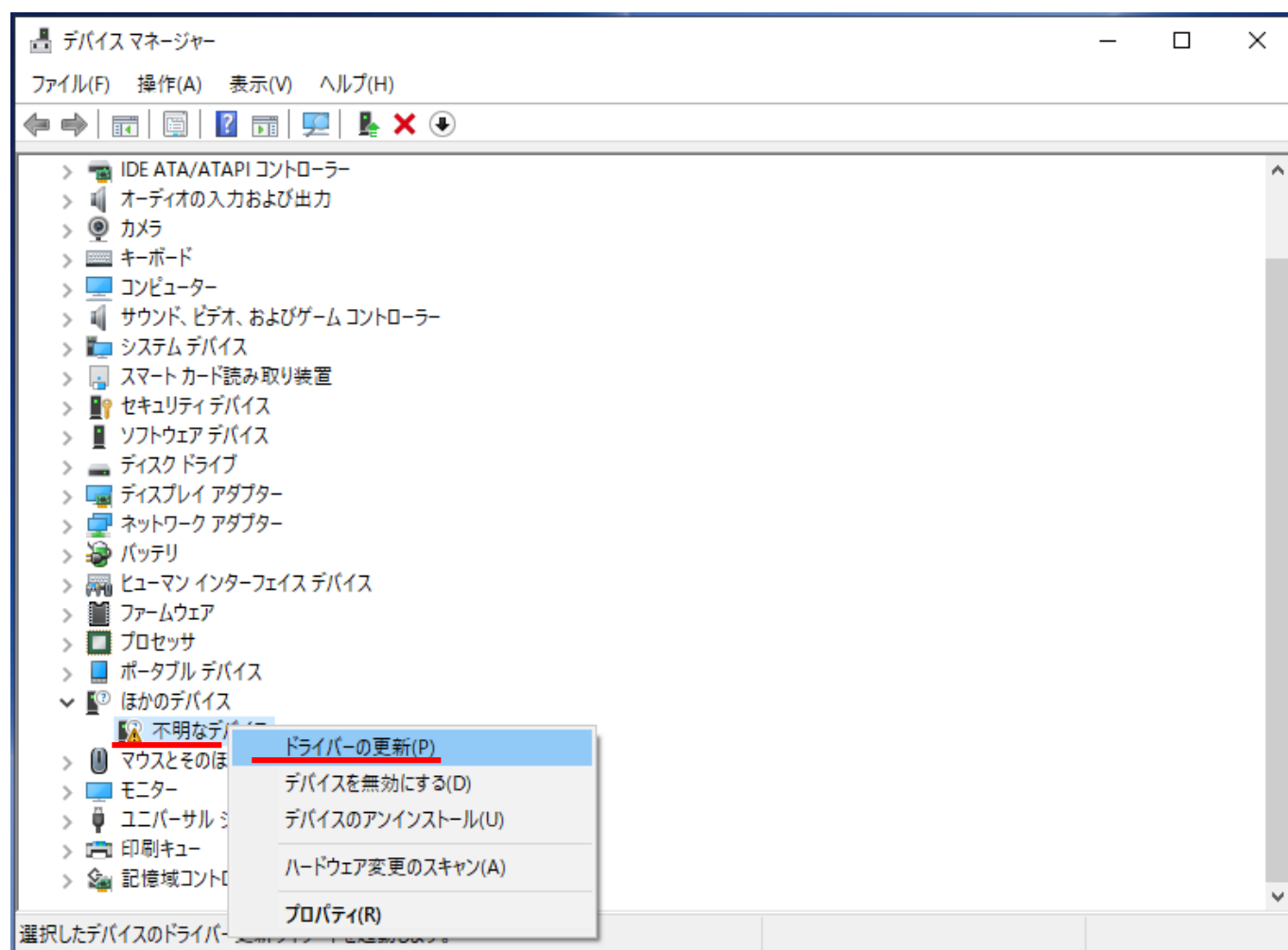
7.1. Windows10 の場合

まい USB のソフトウェア CD を CD ドライブにセットしてください。

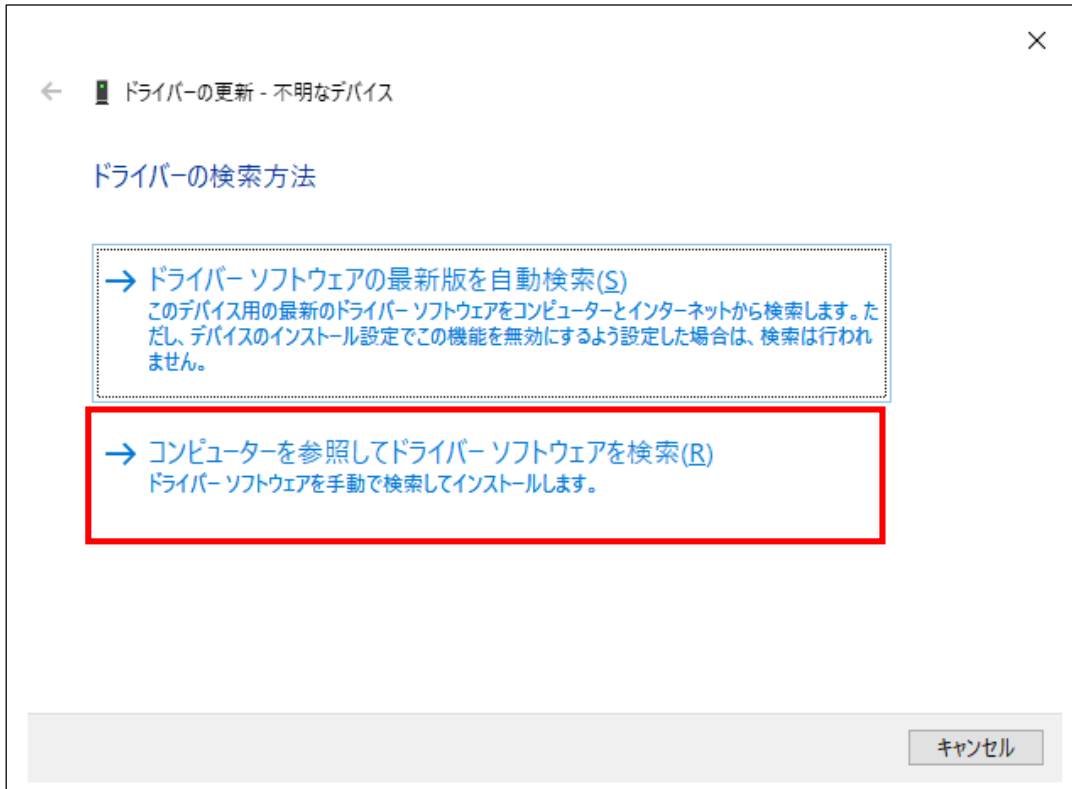
まい USB のファームウェアが書き込まれたターゲットボードを PC に接続してください。



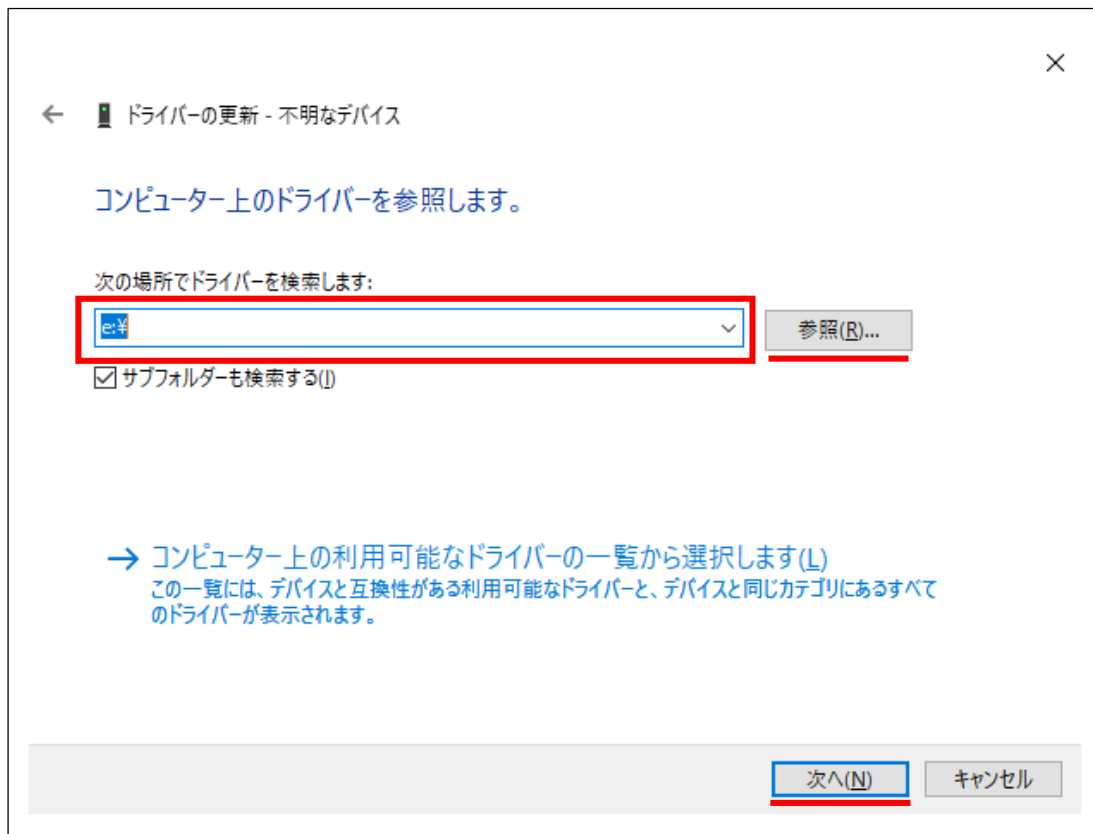
OS の標準ドライバーでは、まい USB が書き込まれたターゲットボードは自動認識しませんので、この時点では「不明なデバイス」として認識されます。



デバイスマネージャを開き、「ほかのデバイス」-「不明なデバイス」を右クリックし、「ドライバーの更新」を選択。



「コンピューターを参照してドライバー ソフトウェアを検索」を選択。



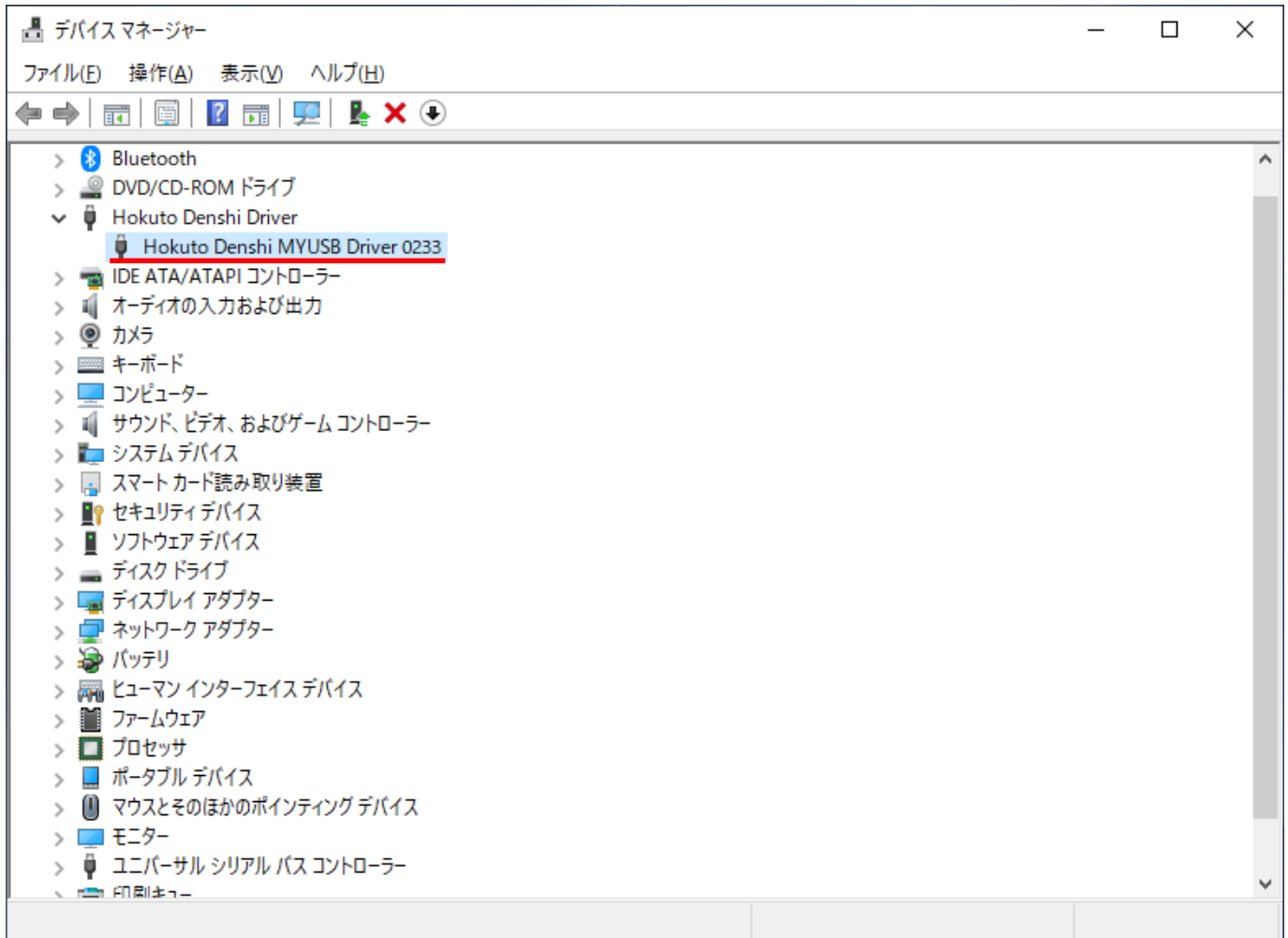
「参照」を押し、CDドライブ内の ¥software¥Host¥Windows10_64bit を選択して、「次へ」を押ししてください。



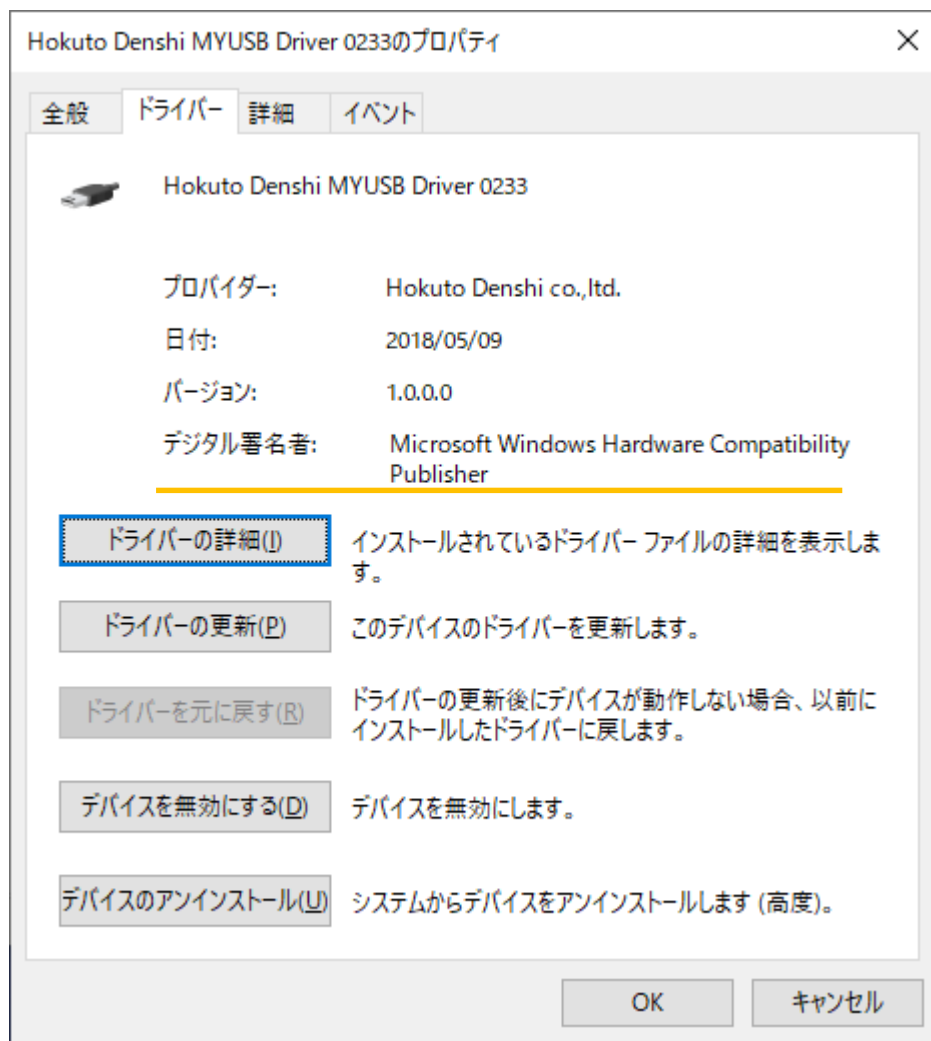
上記画面が出ますので、「閉じる」を押ししてください。

※0233 のところは、PID 番号となり、製品毎に異なります。

[参考]



ドライバーがインストールされると、「不明なデバイス」から「Hokuto Denshi MYUSB Driver XXXX」に変わります。



ドライバーのプロパティを開き、「ドライバー」タブを見ると、デジタル署名者「Microsoft...」となっています。
 (Windows10 向け Microsoft 構成証明署名のファイルとなります)

※まい USB では、Windows Server(及び、~Windows 8.1, Windows10)に適用可能な、HLKドライバーの提供はありません

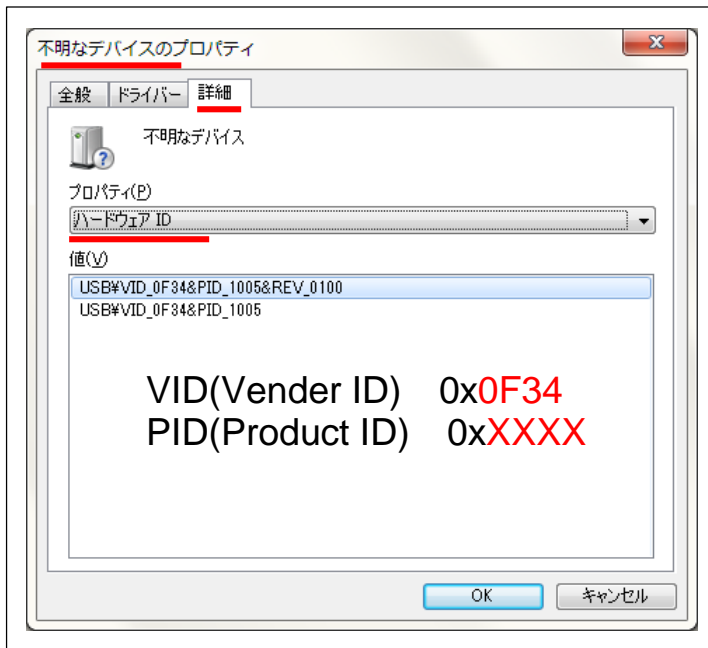
※ドライバーのインストールに失敗した場合

不明なデバイスのプロパティを開いてください。

詳細－ハードウェア ID

を確認し、

USB¥VID_0F34&PID_XXXX



上記の VID ではない場合は、まい USB が書き込まれたターゲットボードではなく、PC に接続されている他のハードウェアが不明なデバイスとなっています。

PID は製品毎に異なり、CD にシールで記載されています。

※Windows10 のデバイスセキュリティに関して

設定 — Windows セキュリティ — デバイスセキュリティ — コア分離 — メモリ整合性



はオフ設定で使用してください

本設定がオンになっている時は、ドライバーがエラーとなり動作致しません。

8. API 関数(ユーザインタフェース)

8.1. USB ホスト(Windows 側)

USB ホストの API 関数の実体は DLL に実装されています。

アプリケーションから関数をコールする場合、DLL をロードなどしてご使用ください。

※DLL は、32bit アプリケーション向けとなっていますので、アプリケーションのターゲットは x86 を選択してください

8.1.1. API 関数一覧

USB ホストの API 関数一覧を表 8-1 に示します。

表 8-1 USB ホスト API 関数一覧

関数名	説明
MyUsbHostInitDll	初期化関数
MyUsbHostTermDll	終了関数
MyUsbHostSendData	データ送信関数
MyUsbHostRecvData	データ受信関数
MyUsbHostGetLastEvent	最終イベント情報取得関数
MyUsbHostGetRecvPointer	受信バッファポインタ取得関数
MyUsbHostGetLastError	最終エラー状態取得関数

8.1.2. API 関数詳細

(1) MyUsbHostInitDll

初期化関数

書式	unsigned char* MyUsbHostInitDll (HWND hWnd, unsigned long messageStsNum, HANDLE hStsEvent, BOOL bSendBlock)		
引数	HWND hWnd		イベントメッセージを受取る Window ハンドル
	unsigned long messageStsNum		状態に変化があったとき(USB 切断時及び、データ送信完了時)に送信するメッセージ ID
	HANDLE hStsEvent		状態に変化があったとき(USB 切断時及び、データ送信完了時)に送信するイベントハンドル
	BOOL bSendBlock		データ送信ブロック指定 TRUE: 送信完了するまでデータ送信関数を終了しない FALSE: 送信完了を待たずにデータ送信関数を終了する

戻り値 受信データバッファポインタ

但し、以下の場合には「NULL」が返ります。

- ・ PIPE が開けなかった場合(USB 切断中又は PIPE がすでに開いている)
- ・ 内部リソースを取得できなかった場合(リソース不足)

詳細 使用する全ての PIPE を開き、正常終了した場合には、受信データのバッファポインタを返します。異常終了した場合には、NULL を返します。

※本関数は、USB 接続時に呼び出してください。

本関数の正常終了後、USB 通信が可能となります。

●引数について

○USB 切断

以下の条件のとき、ステータスに変化があったことをイベントメッセージにてアプリケーションに通知させることができます。

- ✓ hWnd が「NULL 以外」の場合には、hWnd のメッセージキューに messageStsNum をポストします。
- ✓ hStsEvent が「NULL 以外」の場合には、hStsEvent をシグナル状態にします。

○データ送信完了

以下の条件のとき、ファンクションヘデータを送信完了したことをアプリケーションに通知させることができます。

- ✓ bSendBlock が「FALSE」かつ hWnd が「NULL 以外」の場合には、hWnd のメッセージキューに messageStsNum をポストします。
- ✓ bSendBlock が「FALSE」かつ hStsEvent が「NULL 以外」の場合には、hStsEvent をシグナル状態にします。

※アプリケーションへの通知を行わないようにする場合には、bSendBlock を「TRUE」に設定してください。

●イベント発生について

イベントを発生させる場合、イベント発生を契機にアプリケーションにて

MyUsbHostGetLastEvent 関数をコールすることによってイベントの種類を知ることができます。

(2) MyUsbHostTermDll

終了関数

書式	void MyUsbHostTermDll (void)
引数	なし
戻り値	なし
詳細	MyUsbHostInitDll で生成したリソースを開放し、全ての PIPE を閉じます。

アプリケーション終了時、及び USB 切断時(MyUsbHostGetLastEvent 関数の戻り値が EVENT_DISCONNECT)には必ず本関数をコールしてください。

(3) MyUsbHostSendData

データ送信関数

書式	BOOL MyUsbHostSendData(unsigned char* pData, unsigned long len)
引数	unsigned char* pData 送信データバッファポインタ unsigned long len 送信データサイズ(1~4096)
戻り値	TRUE : 正常終了 FALSE : 異常終了 <ul style="list-style-type: none"> ・ PIPE が開いていない。 ・ 送信データバッファが NULL ・ 送信データ長が 0 又は 4097 以上 ・ 接続が切れた場合 ・ データ送信中の場合 (MyUsbHostInitDll 関数で bSendBlock を「FALSE」に設定している場合)

詳細 指定されたデータをデータサイズ分、ファンクションへ送信します。
ファンクション側では MyUsbFunctionGetRecvData 関数でデータを読み出してください。

一回のデータ送信サイズは 4096Byte までですので、4096Byte を超えるデータを送信する場合には、複数回コールしてください。

ホストはファンクションの受信 FIFO があふれていてもデータ送信を行うことができます。
ファンクションの受信 FIFO がオーバーフローを起こさないようにする場合、アプリケーションで制御を行ってください。

(4) MyUsbHostRecvData

データ受信関数

書式	BOOL MyUsbHostRecvData(unsigned long* pLength)
引数	unsigned long* pLength 0 受信データサイズポインタ

以下の場合には、「0」が返ります。

- ・ PIPE が開いていない。
- ・ 受信データがない場合
- ・ 接続が切れた場合

戻り値	TRUE : 正常終了
	FALSE : 異常終了
	・ PIPE が開いていない。
	・ 接続が切れた場合

詳細 ファンクションから MyUsbFunctionSetSendData 関数で送信されたデータを受信し、受信データバッファ(MyUsbHostInitDll 関数/MyUsbHostGetRecvPointer 関数の戻り値)にデータを格納します。

アプリケーション側で受信データサイズ分、受信バッファからデータを読み出してください。

ホストで本関数を呼び出さないとファンクションの送信 FIFO があふれる可能性があります。

ファンクションの送信 FIFO がオーバーフローを起こさないようにする場合、アプリケーションで制御を行ってください。

(5) MyUsbHostGetLastEvent

最終イベント情報取得関数

書式	unsigned long MyUsbHostGetLastEvent(void)
引数	なし
戻り値	EVENT_NON (0) : イベントなし(初期値)
	EVENT_FINISH_SEND (1) : 送信完了イベント
	EVENT_DISCONNECT (0xFFFFFFFF) : 切断イベント

詳細 最後に発生したイベント情報を取得することができます。

(6) MyUsbHostGetRecvPointer

受信バッファポインタ取得関数

書式	unsigned char* MyUsbHostGetRecvPointer(void)
引数	なし
戻り値	受信データバッファポインタ
詳細	初期化が正常終了している場合に、受信データのバッファポインタを返します。異常終了している場合、未初期化時又は、MyUsbHostTermDll 関数をコールした場合には、NULL を返します。

本関数は、MyUsbHostInitDll 関数で確保した受信バッファポインタを返すための関数であり、受信バッファポインタのデータを更新するものではありません。
データ更新するには、MyUsbHostRecvData 関数をコールしてください。

(7) MyUsbHostGetLastError

最終エラー状態取得関数

書式	unsigned long MyUsbHostGetLastError(void)
引数	なし
戻り値	エラー番号
	0 : エラーなし
	100~199 : 初期化系エラー
	200~299 : データ送信系エラー
	300~399 : データ受信系エラー
詳細	DLL 内部で最後に発生したエラー状態を取得することができます。

※本関数は USB 転送機能には一切影響ありません。アプリケーション作成時のデバッグ用としてご使用ください。

8.2. USB ファンクション(ターゲット基板側)

USB ファンクションはライブラリ形式となっています。

アプリケーションからは、ライブラリをリンクすることで、ライブラリの API 関数をコールすることができます。

最大 64Byte × 69 個のバッファリングが可能な受信 FIFO をもっており、ホストからの送信データを「まい USB」にて受信 FIFO に格納し、MyUsbFunctionGetRecvData 関数でデータを読み出すことができます。

ホストへのデータ送信は、アプリケーションにて MyUsbFunctionSetSendData 関数コール時に設定したデータを「まい USB」にて送信します。

【注意事項】

- ・ 割り込みベクタテーブルにUSB割り込み関数「usb_function_interrupt」をご登録ください。
(※V850E2/ME3, H8SX1655, RX用, R8C用のものは、ライブラリ内で登録しておりますので、登録の必要はございません。RAでは、「intr_usbfs_usbi」を定義してください。)
- ・ USBを制御するにあたりターゲット基板に一部制約がある場合がございます。
詳細は、「9章」をご参照ください。

8.2.1. API 関数一覧

USB ファンクションの API 関数一覧を表 8-2、表 8-3 に示します。

表 8-2 USB ファンクション API 関数一覧

関数名	説明
MyUsbFunctionInit	初期化関数
MyUsbFunctionIsConnection	USB デバイス状態取得関数
MyUsbFunctionSetSendData	データ送信関数
MyUsbFunctionGetSendStatus	データ送信状態取得関数
MyUsbFunctionGetRecvData	データ受信関数
MyUsbFunctionGetRecvDataLength	受信データ長取得関数
MyUsbFunctionGetRecvStatus	データ受信状態取得関数
MyUsbFunctionClearRecvErrorStatus	データ受信状態クリア関数
MyUsbFunctionGetRecvFifoCounter	受信 FIFO 数取得関数
MyUsbFunctionSetPriority	USB 割り込み優先度設定関数
MyUsbFunctionSetUserVcc5	USB VCC 設定関数
MyUsbFunctionSetRcvFifo	受信 FIFO 数設定関数
MyUsbFunctionVdcenSet	USB 電源レギュレータ有効化関数

表 8-3 USB ファンクション割り込み処理関数一覧

関数名	説明
usb_function_interrupt	USB ファンクション割り込み処理関数
intr_usbfs_usbi	USB ファンクション割り込み処理関数(RA)

8.2.2. API 関数詳細

以下の API 関数の戻り値として記載しています「BOOL」は「unsigned char」を定義したものです。

TRUE は「1」、FALSE は「0」を示します。

尚、関数定義及び、定数定義については、「usb_function_api.h」に記載しています。

(8) MyUsbFunctionInit

初期化関数

書式 void MyUsbFunctionInit(void)
 引数 なし
 戻り値 なし
 詳細 USB ライブラリを初期化します。

※本関数は、ターゲット基板起動時のみ呼び出してください。
 正しくホストと接続するために約 2 秒ほどの時間がかかります。

※H8S/2456 の場合

本関数をコールする前に、USB 用 PLL 回路周波数通倍率を発振の入力クロック周波数に合わせてください。

発振器の入力クロック 周波数	USB 専用クロック (cku:48MHz)	USB 用 PLL コントロール レジスタ (USPLLCR)
8MHz	EXTAL × 6	11
12MHz	EXTAL × 4	10
16MHz	EXTAL × 3	01

※R8C/34U、R8C/34K、R8C/3MK の場合

本関数をコールする前に、USB VCC 状態及び、受信 FIFO 数を設定してください。

※RA 向け

書式 void MyUsbFunctionInit(unsigned char usbfs_usbi_interrupt_num)
 引数 USBFS_USBI 割り込み番号
 戻り値 なし
 詳細 USB ライブラリを初期化します。

※RA の場合

引数として、FSP で定義した interrupts 番号を与えて呼び出してください。

(9) MyUsbFunctionIsConnection

USB デバイス状態取得関数

書式 BOOL MyUsbFunctionIsConnection(void)
 引数 なし
 戻り値 TRUE : 接続中
 FALSE : 切断中
 解説 USB デバイス状態を取得します。
 接続中になると USB 通信が可能になります。

(10) MyUsbFunctionSetSendData

データ送信関数

書式	BOOL MyUsbFunctionSetSendData(unsigned char *pData, unsigned long length)	
引数	char * pData	送信データバッファポインタ
	unsigned long length	送信データサイズ(1~4096)
戻り値	TRUE	: 正常終了
	FALSE	: 異常終了
		<ul style="list-style-type: none"> ・ 送信データバッファが NULL ・ 送信データ長が 0 又は 4097 以上 ・ 接続が切れている場合 ・ 送信 FIFO がいっぱい

解説 指定されたデータをデータサイズ分、「まい USB」にてホストへ送信します。ホスト側では MyUsbHostRecvData 関数でデータを読み出してください。

MyUsbFunctionGetSendStatus 関数をコールし“SENDING”でないことを確認してからデータバッファの内容を更新し、本関数を実行してください。

一回のデータ送信サイズは 4096Byte までですので、4096Byte を超えるデータを送信する場合には、複数回コールしてください。

(11) MyUsbFunctionGetSendStatus

データ送信状態取得関数

書式	unsigned char MyUsbFunctionGetSendStatus(void)	
引数	なし	
戻り値	SEND_FINISH (0)	: 初期状態/送信完了
	SENDING (1)	: 送信中
	SEND_FAIL (2)	: 送信失敗

解説 データ送信状態を返します。

データ送信時には、本関数をコールし“SENDING”ではないことを確認してから MyUsbFunctionSetSendData 関数コールしてください。

データ送信中に USB ケーブル切断されたときに、SEND_FAIL が返ります。再接続時にも SEND_FAIL のままとなります。

(12) MyUsbFunctionGetRecvData

データ受信関数

書式	unsigned char MyUsbFunctionGetRecvData(unsigned char *pData)
引数	char * pData ○ 受信バッファポインタ*
戻り値	受信データサイズ 以下の場合には、「0」が返ります。 <ul style="list-style-type: none">・ 受信バッファポインタが「NULL」・ 接続が切れた場合・ 受信データがない場合
解説	ホストから MyUsbHostSendData 関数によって送信されたデータを「まい USB」にて受信 FIFO に格納します。 本関数は、受信 FIFO に入っているデータを読み出し、受信データバッファにデータを格納します。 ※本関数を呼び出すときには、アプリケーションで静的に 64Byte 以上を確保してから本関数をコールするか、MyUsbFunctionGetRecvDataLength 関数をコールし、存在している受信データ長分のバッファを確保してから本関数をコールしてください。

* ホスト側から送信したデータがマイコンの FIFO サイズより大きい場合は、MyUsbFunctionGetRecvDataLength の戻り値が “0” になるまで MyUsbFunctionGetRecvData を複数回呼び出してすべてのデータを取得して下さい。

(13) MyUsbFunctionGetRecvDataLength

受信データ長取得関数

書式	unsigned char MyUsbFunctionGetRecvDataLength(void)
引数	なし
戻り値	受信したデータ長
解説	MyUsbFunctionGetRecvData 関数を呼び出した時に得られる受信 FIFO 上の受信バッファポインタが示すデータの長さをバイト単位で返します。

(14) MyUsbFunctionGetRecvStatus

データ受信状態取得関数

書式	unsigned char MyUsbFunctionGetRecvStatus (void)
引数	なし
戻り値	RECV_OK (1) : 正常 RECV_OVERFLOW (0) : 受信 FIFO がオーバーフローしている (ホストからの送信データを取りこぼしている。)
解説	データ受信状態を返します。

本関数にて受信 FIFO がオーバーフローしていたかどうかを取得することができます。
アプリケーションでホストから再送してもらうかどうかを知る手段としてご使用になれます。

(15) MyUsbFunctionClearRecvErrorStatus

データ受信状態クリア関数

書式	void MyUsbFunctionClearRecvErrorStatus (void)
引数	なし
戻り値	なし
解説	データ受信状態を正常に戻します。

本関数をコールするまで、データ受信状態が保持されます。

(16) MyUsbFunctionGetRecvFifoCounter

受信 FIFO 数取得関数

書式	unsigned long MyUsbFunctionGetRecvFifoCounter(void)
引数	なし
戻り値	受信データ数 (0~69)
解説	ライブラリ内で管理している受信 FIFO 内に格納している受信データの個数を返します。

(17) MyUsbFunctionSetPriority

USB 割り込み優先度設定関数

書式	void MyUsbFunctionSetPriority(unsigned short priority)
引数	unsigned short priority USB 割り込み優先度 (レベル:0~15)

戻り値 なし

解説 指定された USB 割り込み優先度を設定します。
15 以上の値を設定した場合には、「15」を設定します。

※MyUsbFunctionInit 関数をコールした場合には「15」が設定されますので、USB 割り込み優先度を変更する場合には、MyUsbFunctionInit 関数をコールした後に本関数をコールしてください。

※H8S/2472 の場合

USB 割り込みレベルの設定ではなく、割り込みコントロールレベル(優先(1)/非優先(0))を設定します。

1 以上の値を設定した場合には、「1」を設定します。

※MyUsbFunctionInit 関数をコールした場合には「1」が設定されますので、USB 割り込みコントロールレベルを変更する場合には、MyUsbFunctionInit 関数をコールした後に本関数をコールしてください。

※H8S/2456、V850E2/ME3、H8SX/1655、R8C/34U、R8C/34K、R8C/3MK の場合

USB 割り込み優先度が 0(最低)~7(最高)となります。

7 以上の値を設定した場合には、「7」を設定します。

※MyUsbFunctionInit 関数をコールした場合には「7」が設定されますので、USB 割り込み優先度を変更する場合には、MyUsbFunctionInit 関数をコールした後に本関数をコールしてください。

※RA6/RA4 の場合

初期値は 0(最高優先度)です。優先度は、0(最高)~15(最低)の範囲内で設定可能です。

※RA2L1 の場合

初期値は 0(最高優先度)です。優先度は、0(最高)~3(最低)の範囲内で設定可能です。

(18) MyUsbFunctionSetUserVcc5

USB VCC 設定関数

書式	void MyUsbFunctionSetUserVcc5(BOOL flag)
引数	BOOL flag TRUE:ターゲット基板電源 5V FALSE: ターゲット基板電源 3.3V

戻り値 なし
 解説 本関数は R8C 専用の関数となっております。

R8C に搭載されている USB モジュールの電源を制御するための関数であり、ターゲット基板が 5V の場合、TRUE
 ターゲット基板が 3.3V の場合、FALSE
 を設定してください。

尚、本関数をコールした後の MyUsbFunctionInit 関数にて、本関数の設定が有効になります。

(19) MyUsbFunctionSetRcvFifo

受信 FIFO 数設定関数

書式	BOOL MyUsbFunctionSetRcvFifo(unsigned long num)
引数	unsigned long num 受信 FIFO 数(3~)
戻り値	TRUE : 正常終了 FALSE : 異常終了

解説 本関数は R8C 専用の関数となっております。

R8C では RAM 領域が小さいため、初期状態で受信データ 64Byte 分のデータのみライブラリにて確保できるようにしております。
 多くのデータを送受信できるようにユーザーにて受信データを確保できるようにしております。

1つの受信 FIFO サイズは 68Byte となっておりますので、HEAP 領域を確保する際には、 $68 \times \text{受信 FIFO 数} + 2\text{Byte}$ を確保してください。

受信 FIFO 数が 3 未満の場合には、HEAP 領域を確保せず初期状態のまま動作いたします。

尚、本関数をコールした後の MyUsbFunctionInit 関数にて、本関数の設定が有効になります。

(20) MyUsbFunctionVdcenSet

USB VDCEN 設定関数

書式	void MyUsbFunctionVdcenSet(void)
引数	なし
戻り値	なし
解説	USB 電源レギュレータ有効化関数

VDCEN=1 を設定する関数です。USB 電源レギュレータを持っているマイコンで有効です。

尚、本関数をコールした後の MyUsbFunctionInit 関数にて、本関数の設定が有効になります。

VDCEN=1 に設定する場合、MyUsbFunctionInit 関数を呼び出す前に、本関数をよびだしてください。

(21) usb_function_interrupt

USB 割り込み処理関数

書式	void usb_function_interrupt(void)
引数	なし
戻り値	なし
解説	USB 割り込み処理を行います。 <ul style="list-style-type: none">・ USB ホストへの接続検出・ USB コネクタからの切断検出・ 認証処理・ 送信 FIFO に格納されているデータをホストへ送信する処理・ ホストからデータを受信し、受信 FIFO の格納する処理

※本関数は、API 関数ではございませんので、アプリケーションからの関数コールは行わないでください。

(22) intr_usbfs_usbi

USB 割り込み処理関数(RA)

書式	void intr_usbfs_usbi(void)
引数	なし
戻り値	なし
解説	USB 割り込み処理を行います。 <ul style="list-style-type: none">・ USB ホストへの接続検出・ USB コネクタからの切断検出・ 認証処理・ 送信 FIFO に格納されているデータをホストへ送信する処理・ ホストからデータを受信し、受信 FIFO の格納する処理

※本関数は、API 関数ではございませんので、アプリケーションからの関数コールは行わないでください。

RA 向けの割り込み関数です。FSP の割り込み設定で、USBFS - USBFS INT の割り込み関数名として、"intr_usbfs_usbi"を定義してください。

9. ターゲット基板別制約

OSH7216 の場合

USB の D+制御をポート B の PB15 に接続したハードウェアを作成してください。

OSH7285、SH7286 の場合

USB の D+制御をポート B の PB10 に接続したハードウェアを作成してください。

OH8S/2472 の場合

USB の D+制御を PULLUP 端子に接続したハードウェアを作成してください。

OH8S/2456 の場合

USB の D+制御をポート 2 の P20 に接続したハードウェアを作成してください。

OV850E2/ME3 の場合

USB の D+制御をポート 5 の P50 に接続したハードウェアを作成してください。

USB の VBUS 制御をポート 5 の P51 に接続したハードウェアを作成してください。

USB の UCLK 制御をポート 1 の P10 に接続したハードウェアを作成してください。

OH8SX/1655 の場合

USB の D+制御をポート M の PM4 に接続したハードウェアを作成してください。

ORX621、RX62N の場合

USB の D+制御を P14 に接続したハードウェアを作成してください。

USB の VBUS 制御を P16 に接続したハードウェアを作成してください。

ORX630 の場合

USB の D+制御を P14 に接続したハードウェアを作成してください。

USB の VBUS 制御を P16 に接続したハードウェアを作成してください。

ORX631/RX63N の場合 (USB0 又は USB1 どちらかを使用)

USB0 の D+制御を P14、P23 又は P31 に接続したハードウェアを作成してください。

USB0 の VBUS 制御を P16 に接続したハードウェアを作成してください。

USB1 の D+制御を P15 に接続したハードウェアを作成してください。

USB1 の VBUS 制御を P17 に接続したハードウェアを作成してください。

※まい USB ではポート設定を行なっておりませんので、ユーザにてポート設定を行なって下さい。

ORX231 の場合

電源電圧、USB_VCC端子の接続に応じ、USBの電源レギュレータをONする場合、MyUsbFunctionVdcenSet()を、まいUSB初期化(MyUsbFunctionInit()コール)前に行ってください

ORX64MT/RX71M の場合(USB0 又は USB A どちらかを使用)

USB0: VBUS 制御を P16 に接続したハードウェアを作成してください。

USB A: VBUS 制御を P11 に接続したハードウェアを作成してください。

ORX72T/RX66T の場合

USB の VBUS 制御を PD2 に接続したハードウェアを作成してください。

ORX72M/RX72N/RX66N の場合

USB の VBUS 制御を P16 に接続したハードウェアを作成してください。

ORA の場合

USB の VBUS 制御を P407 に接続したハードウェアを作成してください。

ORA4M1/RA2A1 の場合

電源電圧、USB_VCC端子の接続に応じ、USBの電源レギュレータをONする場合、MyUsbFunctionVdcenSet()を、まいUSB初期化(MyUsbFunctionInit()コール)前に行ってください

10.付録

CDに付属していますサンプルアプリケーションについて説明します。

尚、サンプルアプリケーションは、北斗電子製のボードで動作するアプリケーションです。

サンプルアプリケーションの実行環境を図10に示します。

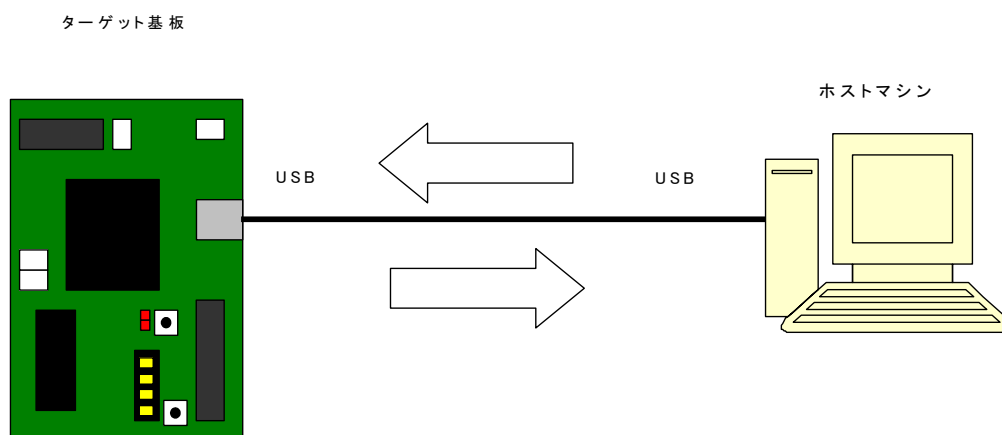


図10 サンプルアプリケーションの実行環境

10.1. サンプルアプリケーションについて

サンプルアプリケーションは、CDROMの「sample フォルダ」もしくは「demo フォルダ」に収録されております。ご参考としてお使い下さい。

※製品によっては、どちらか一方のご用意となります。

■実行方法

サンプルアプリケーションを実行するための手順を以下に示します。

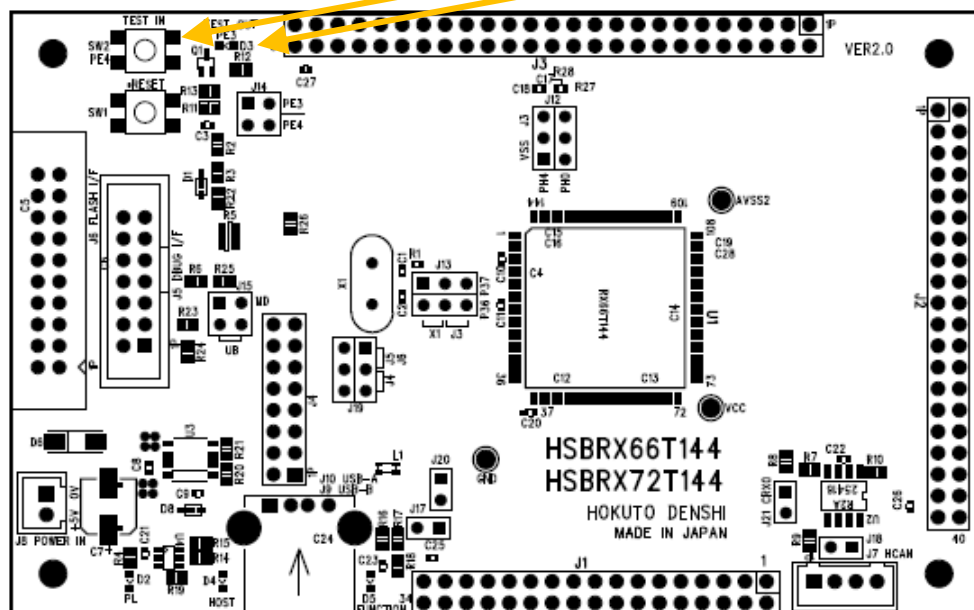
- (1) CDに入っています「Sample/Host/Demo」をホストマシンにコピーしてください。
- (2) CDに入っています「sample.mot」もしくは「demo.mot」をお手持ちのWriterでターゲット基板に書き込んでください。
- (3) ホストマシンとターゲット基板をUSBケーブルで接続し、ターゲット基板の電源を投入してください。
- (4) 7章に従いホストマシン上にドライバーソフトをインストールしてください
- (5) インストール完了後、①でコピーした「sample.exe」もしくは「demo.exe」を起動してください。

ORX66T の場合

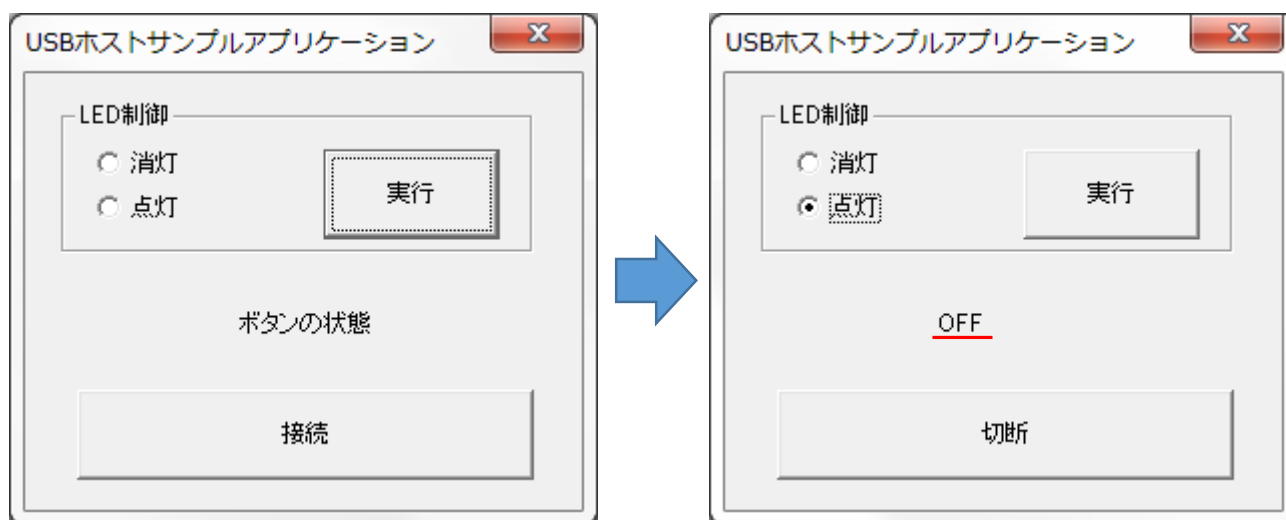
SW2
評価用 SW

D3
評価用 LED

ターゲットボード: HSBRX66T144



ホストマシン上アプリケーション(sample.exe)



(1)接続を押す

「接続」ボタンが「切断」に変われば接続は成功しています

※「接続」のまま変わらないときは、ドライバーのインストールが成功しているか、確認を行ってください

(2)「点灯」にチェックを入れ「実行」を押す

→ターゲットボードの D3 が点灯します

(3)ターゲットボードの SW2 を押してください

→OFF の表示が ON に変わります

(4)「切断」ボタンを押してください

11.注意事項

11.1. 64bit 版の注意事項

まい USB(64bit 版)はドライバーソフト(myusb_XXXX.sys)が 64bit 版のドライバーとなります。
Windows アプリケーション作成時に用いる DLL(myusb_XXXX.dll)は 32bit 版となりますので、64bit 環境においてもプログラムのビルドは 32bit アプリケーションとして行ってください。

11.2.「まい USB」と「まい USBLite」の違いについて

「まい USBLite」では、以下の違いがございます。

- ・ USB ホスト、USB ファンクションの間のデータ送受信機能

「まい USBLite」では、API 関数の引数制限はございませんが、通信速度と通信データ長に制限が入っております。通信データ長は 64Byte 固定で送信され、先頭 1Byte のみアプリケーションから入力したデータとなります。

- ・通信時のウェイト

「まい USBLite」では、ホスト→ターゲット(ファンクション)通信時に 500ms のウェイトが挿入されます。
(ホスト上で動作するサンプルプログラム上で LED ON/OFF のボタンを押した際、ターゲット側での反応が遅い動作となりますが、製品版の「まい USB」では、そのような制約はありません。

12.著作権・免責に関して

本書の内容は予告無く変更する場合があります。本書は著作権により保護されています。

株式会社北斗電子(以下当社)の文書による事前の許諾無しに、本書を複写または複製、転載する事は禁じられています。

当社は本書の内容について万全を期して作成し、正確と確信しておりますが、当社による本書に関する保証は一切なく、特定の目的の市販性、正当性、適合性に関する黙示の保証に対する責任を否認します。本書の内容は予告無しに変更する事があります。本書の中に誤りがある場合でも、当社はいかなる責任も負いません。

本書の内容に関するお問い合わせはご容赦下さい。

本ソフトウェアをご利用になる際、別紙「ソフトウェア使用権許諾契約書」を熟読ください。

本ソフトウェアの価格にはサポート料は含まれておりません。

お問合せ窓口

最新情報については弊社ホームページをご活用ください。

ご不明点は弊社サポート窓口までお問合せください。

株式会社 **北斗電子**

〒060-0042 札幌市中央区大通西 16 丁目 3 番地 7

TEL 011-640-8800 FAX 011-640-8801

e-mail: support@hokutodenshi.co.jp (サポート用)、order@hokutodenshi.co.jp (ご注文用)

URL: <https://www.hokutodenshi.co.jp>

商標等の表記について

- ・ 全ての商標及び登録商標はそれぞれの所有者に帰属します。
- ・ パーソナルコンピュータを PC と称します。

ルネサス エレクトロニクス社 マイコン向けソフトウェア製品

まい USB 取扱説明書

株式会社 **北斗電子**

©2009-2022 北斗電子 Printed in Japan 2022 年 3 月 30 日改訂 REV.1.3.0.0 (220330)
