

μPD70F3769  
USB シリアル変換サンプル・プログラム  
CDC サンプル・ドライバ  
ユーザーズ・マニュアル

(第 1.0 版)

テセラ・テクノロジー株式会社

- 本資料の内容は予告なく変更することがあります。
- 文書による当社の承諾なしに本資料の転載複製を禁じます。
- 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
- 本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。

## 目次

1. 概要.....	3
1.1. 制限事項.....	エラー! ブックマークが定義されていません。
2. 構成.....	4
2.1. システム構成.....	4
2.2. ファイル構成.....	5
3. 開発環境.....	6
4. ROM・RAM サイズ .....	7
5. コミュニケーションデバイスクラス (CDC) .....	8
5.1. CDC の位置づけ .....	9
6. サンプル・アプリケーション .....	10
6.1. サンプル・アプリケーションの位置づけ.....	10
7. 処理フロー .....	11
7.1. 初期化処理 .....	11
7.1.1. CPU 初期化.....	12
7.1.2. USB 初期化処理.....	14
7.2. エンドポイント監視処理.....	16
7.2.1. エンドポイント0 監視処理.....	16
7.2.2. エンドポイント1 監視処理.....	18
7.3. サンプル・アプリケーション処理 .....	19
8. 関数仕様.....	21
9. レジスタ設定.....	26
9.1. CPU 関連レジスタの設定 .....	27
9.2. USB 関連レジスタの設定 .....	29
9.3. ディスクリプタ設定.....	36
10. USB シリアル変換サンプル・プログラムの構成.....	45
10.1. ドライバのインストール.....	46

# 1. 概要

本書は  $\mu$ PD70F3769 内蔵の USB ファンクション・コントローラ用サンプル・プログラム、コミュニケーションデバイスクラス(CDC)サンプル・ドライバについて説明するものです。

サンプル・ドライバの主な機能を以下に示します。

- ・ FS(フル・スピード: 12Mbps)デバイスとして動作します。
- ・ バス・パワー・デバイスとして動作します。
- ・ ホスト接続時、コミュニケーションクラス(仮想 COM)として認識されます。
- ・ ターミナルソフトにて仮想 COM ポートを指定することで、通信を行うことができます。
- ・ ファンクション側で受信した通信データは、サンプル・アプリケーション処理部により大文字 ⇄ 小文字変換され、再度、仮想 COM として USB 出力されます。
- ・ ターミナルソフトからボーレート、ストップビット、データ長、パリティビットの変更が可能です。(サンプル・ドライバでは、これらの設定値は使用しません。)

## 1.1. 注意事項

本「CDC サンプル・ドライバ」をお客様の装置に組み込む場合は、お客様が所有するベンダーIDに変更してください。

## 2. 構成

### 2.1. システム構成

サンプル・プログラムのシステム構成を次に示します。

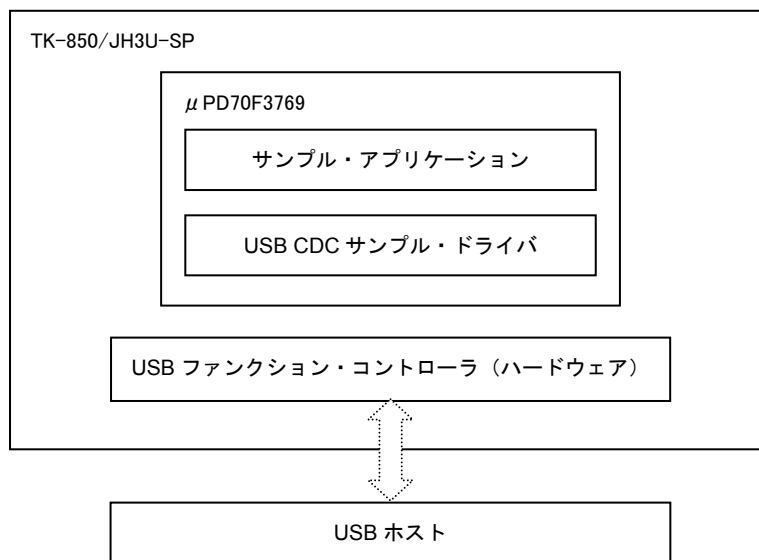


図 2-1 サンプル・プログラムのシステム構成

## 2.2. ファイル構成

サンプル・プログラムのファイル構成を次に示します。

### 【source】

ファイル名	処理内容
main.c	メインルーチン、初期化
usb850.c	USB 初期化、エンドポイント制御、バルク転送、コントロール転送
usb850_communication.c	CDC 固有処理

### 【include】

ファイル名	処理内容
errno.h	エラーコード定義
main.h	main.c 関数プロトタイプ宣言
RegDef.h	レジスタ定義
Types.h	ユーザ型宣言
usb850.h	usb850.c 関数プロトタイプ宣言
usb850_communication.h	usb850_communication.c 関数プロトタイプ宣言
usb850_desc.h	ディスクリプタ定義
usb850_sfr.h	USB ファンクション・コントローラ用レジスタアクセス用マクロ定義

### 【inf file】

ファイル名	処理内容
JG3H_CDC_VISTA.inf	Windows Vista 用 inf ファイル
JG3H_CDC_XP.inf	Windows XP 用 inf ファイル

### 3. 開発環境

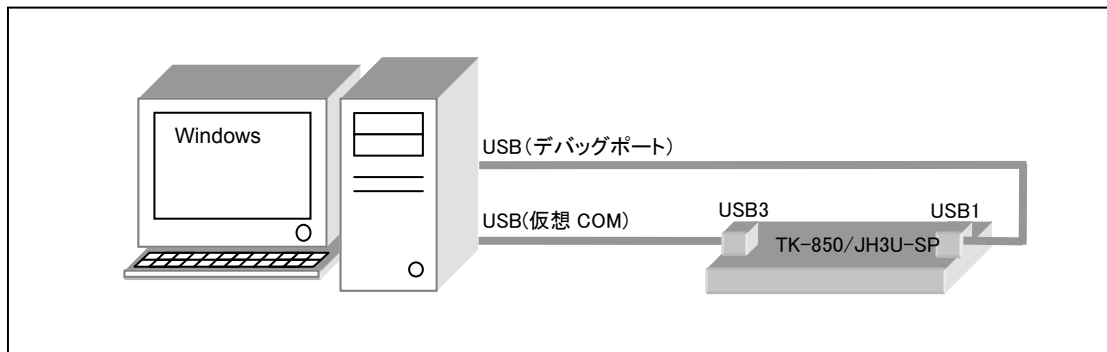


図 3-1 開発環境構築例(ハードウェア)

## 4. ROM・RAM サイズ

μ PD70F3769 の ROM/RAM 容量は以下の通りです。

ROM 容量: 512 [Kbyte]

RAM 容量: 56 [Kbyte]

サンプル・プログラム(サンプル・ドライバ部+サンプル・アプリケーション部)で使用する ROM/RAM サイズは以下の通りです。※ベクタテーブルサイズは除く

ROM サイズ: 約 3.5 [Kbyte]

RAM サイズ: 約 1.1 [Kbyte]

## 5. コミュニケーションデバイスクラス(CDC)

USB のコミュニケーションデバイスクラス(CDC)仕様に関しては、USB CDC 規格書「Universal Serial Bus Class Definitions for Communication Devices Version1.1」を参照してください。

サンプル・ドライバは、CDC の Abstract Control Model で、対応するクラスリクエストは次の通りです。

### SendEncapsulatedCommand

コミュニケーションクラス・インターフェースの制御プロトコルのフォーマットでコマンドを発行する為のリクエストです。

### GetEncapsulatedResponse

コミュニケーションクラス・インターフェースの制御プロトコルのフォーマットで応答を要求する為のリクエストです。

### SetLineCoding

シリアル通信の通信フォーマットを指定する為のリクエストです。

### GetLineCoding

デバイス側の現在の通信フォーマット設定を取得する為のリクエストです。

### SetControlLineState

RS-232/V.24 形式の制御信号のためのリクエストです。



## 5.1. CDC の位置づけ

サンプル CDC は以下の様な位置づけとなっています。

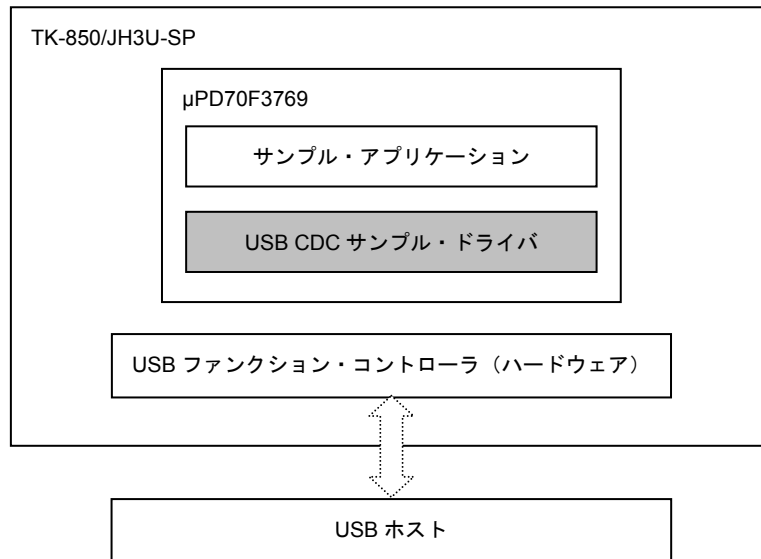


図 5-1 CDC の位置づけ

## 6. サンプル・アプリケーション

サンプル・アプリケーションでは、 $\mu$ PD70F3769 内蔵の USB ファンクション・コントローラで受信したユーザーデータを読み出し、大文字 $\leftrightarrow$ 小文字変換 (ASCII 文字コード) を行い、変換したデータを再度、仮想 COM として USB 出力します。

### 6.1. サンプル・アプリケーションの位置づけ

サンプル・アプリケーションは以下の様な位置づけになっています。

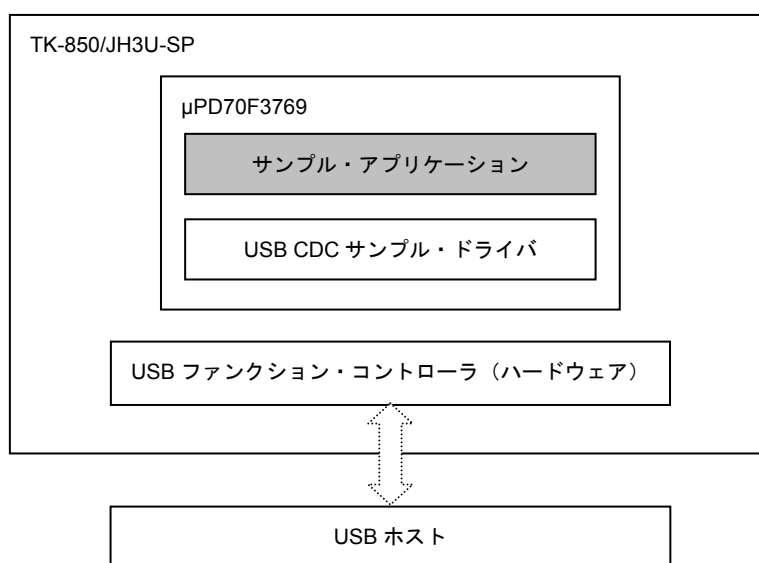


図 6-1 サンプル・アプリケーションの位置づけ

## 7. 処理フロー

サンプル・プログラムで実装されている処理を以下に示します。

- ・ 初期化処理
- ・ エンドポイント監視処理
- ・ サンプル・アプリケーション処理

※サンプル・ドライバでは、エンドポイント監視を割り込みベクタではなくポーリングで各エンドポイントの割り込みフラグを監視することで行っています。

### 7.1. 初期化処理

初期化処理では以下の処理を行っています。

- ・ CPU 初期化
- ・ USB 初期化

### 7.1.1. CPU 初期化

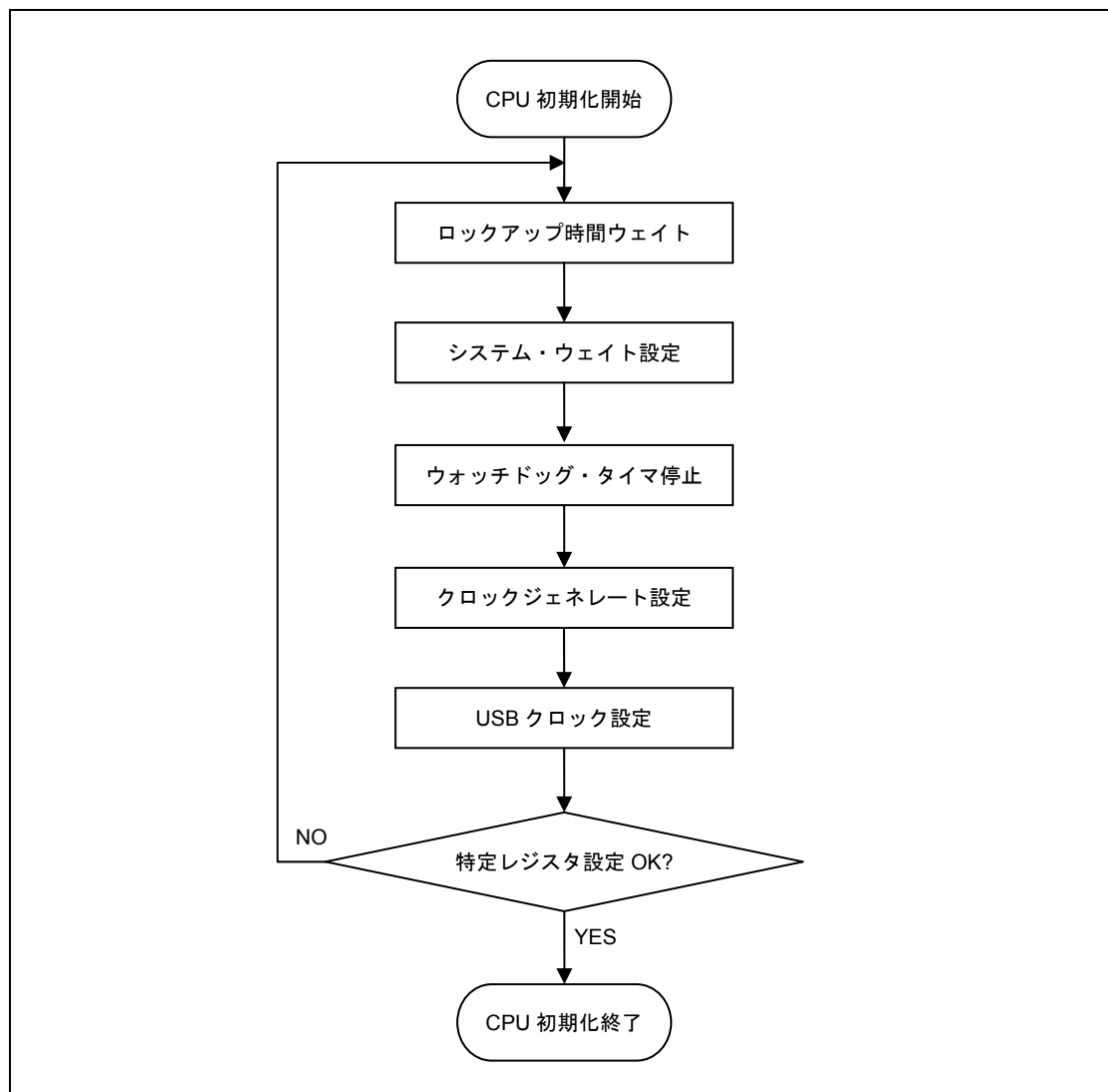


図 7-1 CPU 初期化フロー

CPU 初期化で実行する処理内容を次に示します。

- ・ ロックアップ時間ウェイト  
LOCKR レジスタを監視し、周波数が安定する(ロック状態)までウェイトします。
- ・ システム・ウェイト設定  
VSWC レジスタで内蔵周辺 I/O レジスタに対するバス・アクセスのウェイトを制御します。
- ・ ウォッチドッグ・タイマ停止  
WDTM2 の WDM21・WDM20 ビットを”0”にすることでウォッチドッグ・タイマの動作モードを停止に設定します。
- ・ クロックジェネレート設定  
CKC レジスタでメイン・クロックの逡倍率を指定します。  
PLLCTL レジスタで PLL モードの ON/OFF 設定を指定します。  
PCG レジスタで CPU クロックの分周率や、メイン/サブクロックでの動作を指定します。
- ・ USB クロック設定  
UCKSEL レジスタ、UFCKMSK レジスタ、UHCKMSK レジスタで USB クロックを制御します。
- ・ 特定レジスタ設定時のエラー発生確認  
CKC レジスタ、PCG レジスタは特定レジスタである為、操作後に SYS レジスタの PRERR ビットにてプロテクション・エラーが発生していないか確認する必要があります。

## 7.1.2. USB 初期化処理

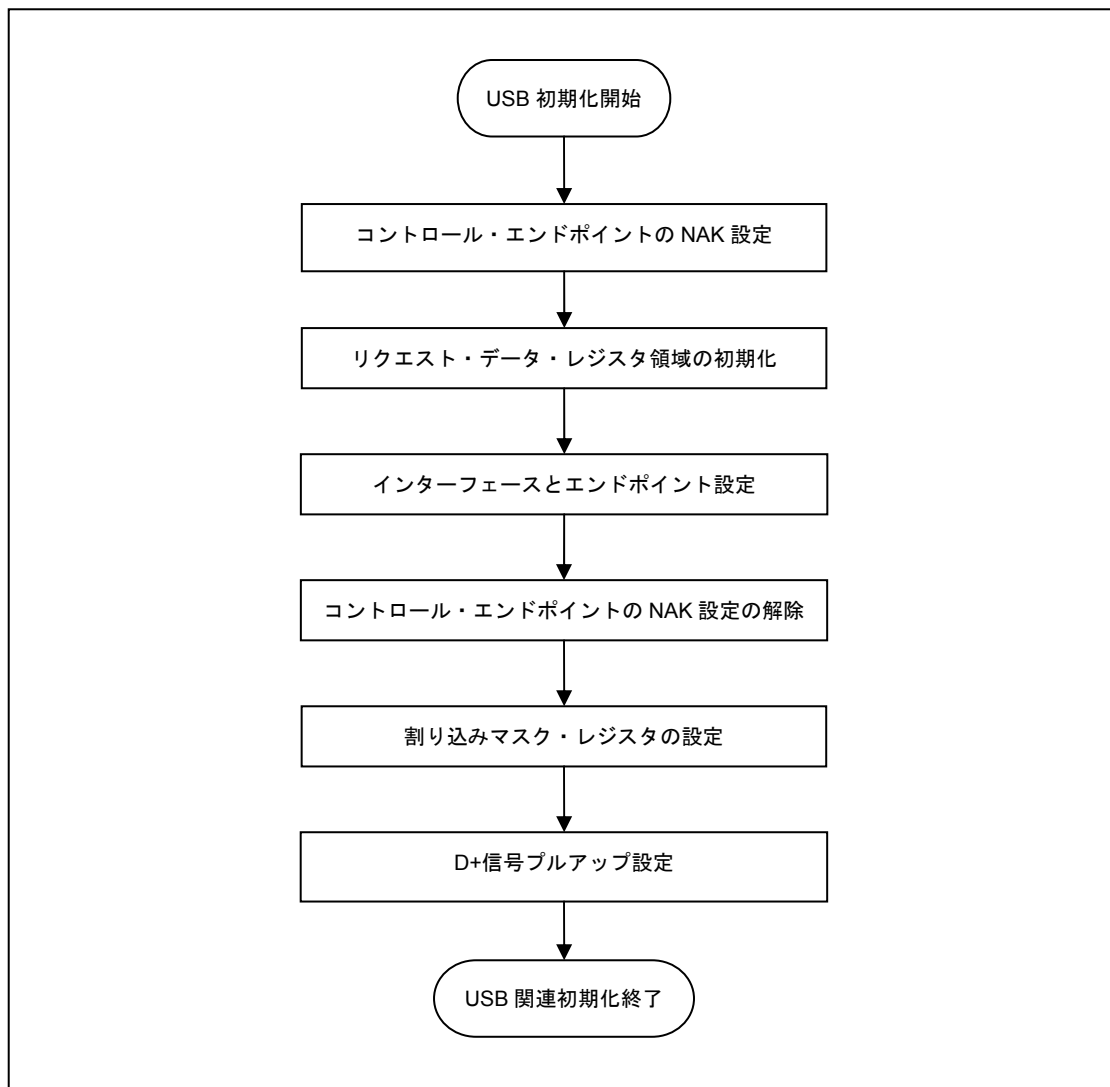


図 7-2 USB 初期化フロー

USB 初期化で実行すべき処理内容を次に示します。

- ・ コントロール・エンドポイントの NAK 設定  
自動実行リクエストを含む全てのリクエストに NAK 応答します。  
自動実行リクエストで使用するデータの登録が完了するまでハードウェアが自動実行リクエストに意図しないデータを返さないように設定します。
- ・ リクエスト・データ・レジスタ領域の初期化  
Get Descriptor リクエストに응答するためのディスクリプタ・データなどをレジスタに登録します。  
登録するデータは、デバイス・ステータス、エンドポイント 0 ステータス、Device Descriptor、Configuration Descriptor、Interface Descriptor、Endpoint Descriptor です。
- ・ インターフェースとエンドポイントの設定  
サポートするインターフェースの数、オルタナティブ設定の状態、インターフェースとエンドポイントの関係などの情報をレジスタに設定します。
- ・ コントロール・エンドポイントの NAK 設定の解除  
自動実行リクエスト用のデータ登録が終わったところで、コントロール・エンドポイントの NAK 設定を解除します。
- ・ 割り込みマスク・レジスタの設定  
USB ファンクション・コントローラの割り込みステータス・レジスタに示される割り込み要因ごとのマスクを設定します。
- ・ D+信号プルアップ設定  
D+信号をプルアップし、ホスト側にデバイスが接続されたことを認識させます。

## 7.2. エンドポイント監視処理

エンドポイント監視処理では、エンドポイント 0(コントロール転送用エンドポイント)、エンドポイント 1(バルクアウト用エンドポイント)の FIFO にデータあるかどうかを監視しています。

### 7.2.1. エンドポイント 0 監視処理

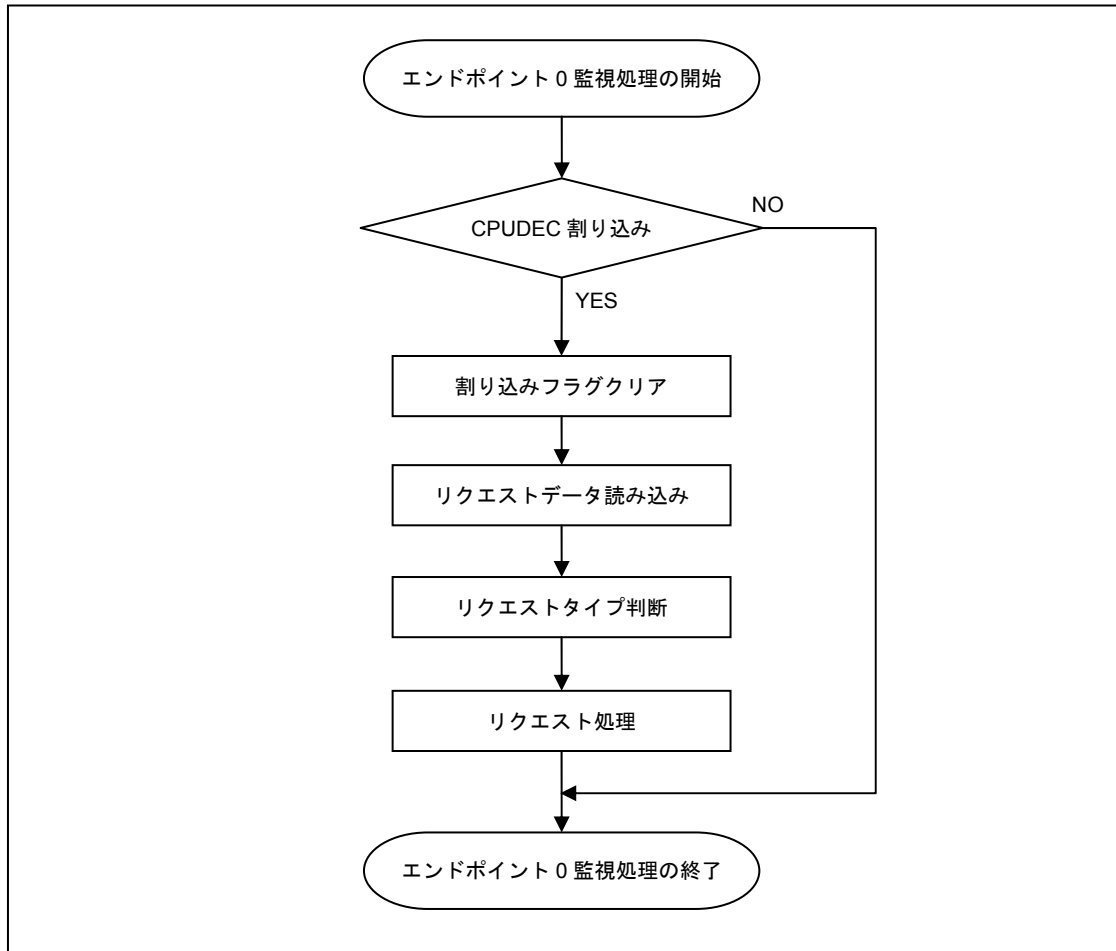


図 7-3 エンドポイント 0 監視処理フロー



エンドポイント 0 監視処理で実行すべき処理内容を次に示します。

エンドポイント 0 はコントロール転送用のエンドポイントですが、プラグイン時のエニュメレーションで使用するような標準デバイスリクエストの一部はハードウェアで自動応答する為、ここで監視する必要があるのはハードウェアで自動応答しない標準リクエスト、クラスリクエスト、ベンダリクエストです。

- ・ CPUDEC 割り込み判断  
UF0IS1 の CPUDEC ビットが ON(1)していることで、割り込みが発生していると判断します。
- ・ 割り込みフラグクリア  
UF0IC1 の CPUDECC ビットを OFF(0)することで割り込み要因をクリアします。
- ・ リクエストデータ読み込み  
受信データを FIFO から読み込み、リクエストデータを構成します。
- ・ リクエストタイプ判断  
リクエストデータが、(ハードウェアで)自動実行しない標準リクエストかクラスリクエストかベンダリクエストかを判断します。
- ・ リクエスト処理  
リクエストタイプによりそれぞれのリクエスト処理を実行します。

## 7.2.2. エンドポイント 1 監視処理

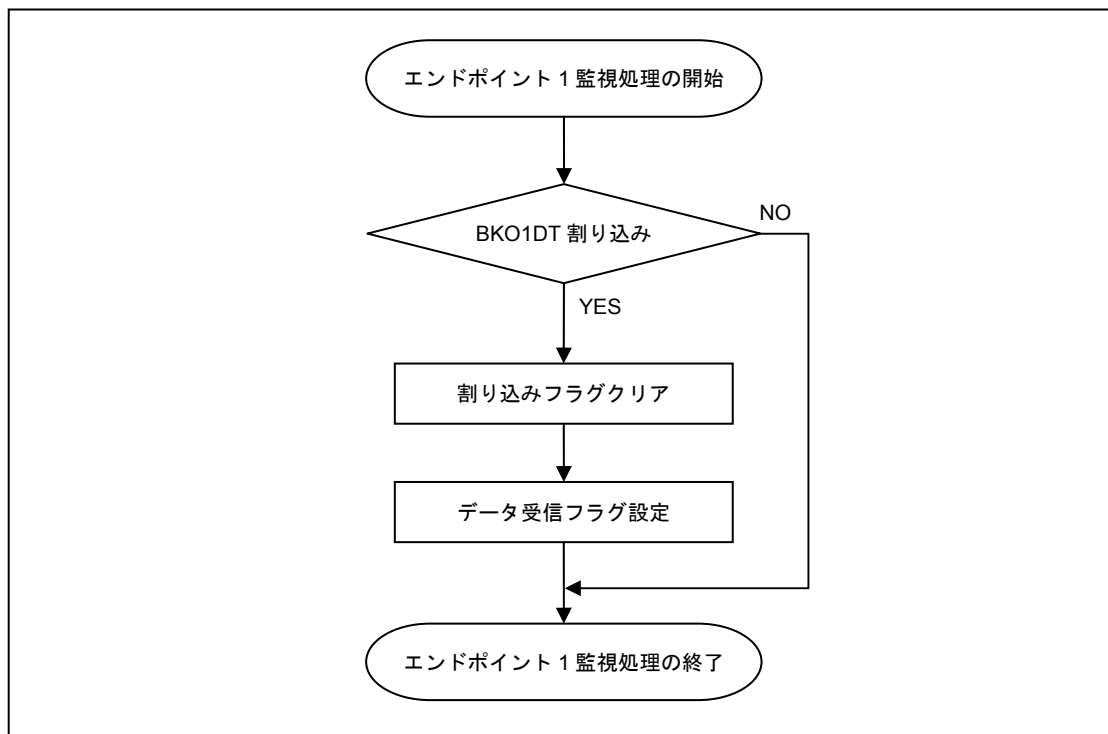


図 7-4 エンドポイント 1 監視処理フロー

エンドポイント 1 監視処理で実行すべき処理内容を次に示します。

- ・ BKO1DT 割り込み判断  
UF0IS3 レジスタの BKO1DT ビットが ON(1)していることで正常受信完了割り込みが発生したと判断します。
- ・ 割り込みクリア  
UF0IC3 レジスタの BKO1DTC ビットを OFF(0)することで割り込み要因をクリアします。
- ・ データ受信フラグ設定  
データ受信フラグ(usb850\_rdata\_flg)をセット(1)します。

### 7.3. サンプル・アプリケーション処理

サンプル・アプリケーション処理では、データ受信フラグ(usb850\_rdata\_flg)を監視しています。

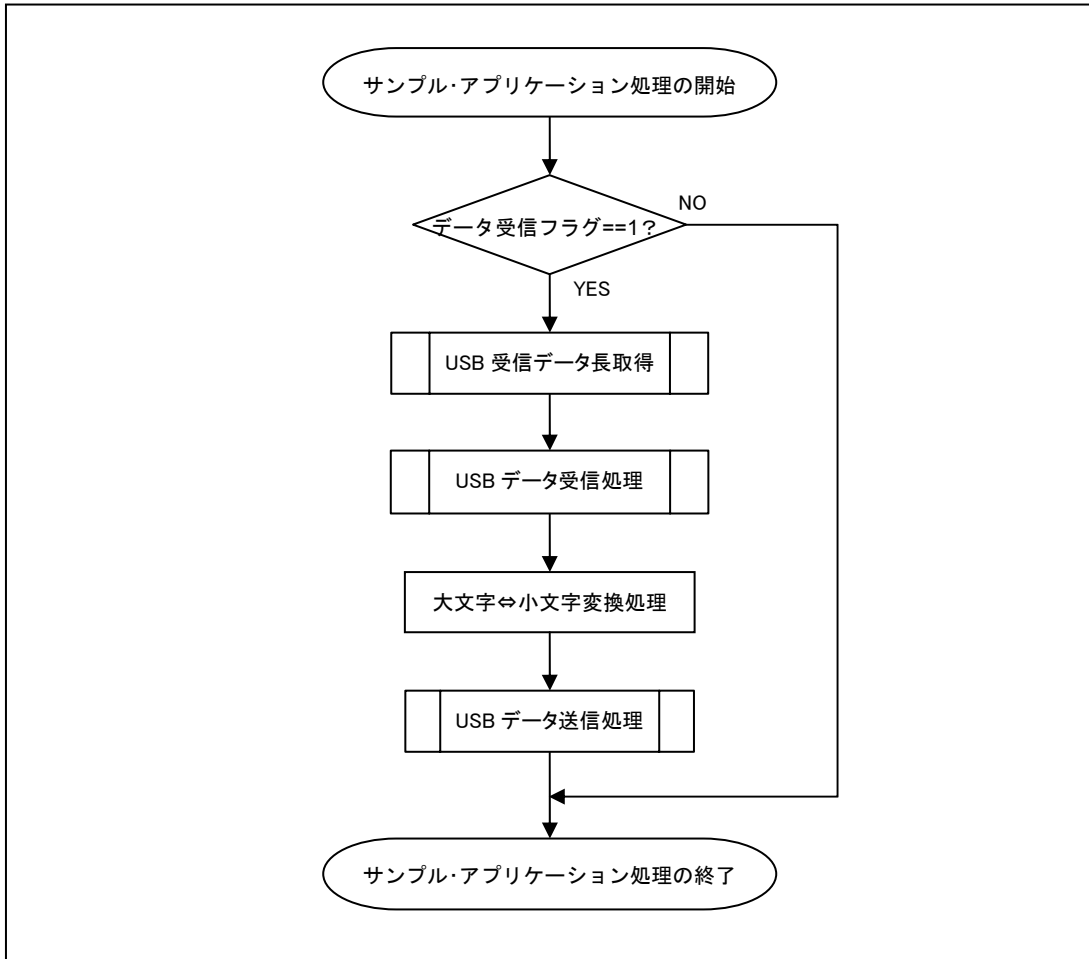


図 7-5 サンプル・アプリケーション処理フロー

サンプル・アプリケーション処理で実行する処理内容を次に示します。

- ・ データ受信フラグ判断  
データ受信フラグ(usb850\_rdata\_flg)を確認します。このフラグは、BK01DT 割り込みにより設定されます。フラグがセット(1)されているときは、受信データが存在することを意味します。
- ・ USB 受信データ長取得  
受信データ長を取得します。この API は、サンプル・ドライバから提供されます。
- ・ USB データ受信処理  
受信データを格納するバッファ、受信するデータ長、および、受信用エンドポイント番号を引数に受信処理を呼び出します。この API は、サンプル・ドライバから提供されます。
- ・ 大文字⇔小文字変換処理  
USB の受信 FIFO から読み出したデータの ASCII 文字コードにおいて、大文字を小文字に、小文字を大文字に変換します。大文字、小文字以外は変換しません。変換する文字コードは、'A' ~ 'Z'、'a' ~ 'z' です。
- ・ USB データ送信処理  
変換の終了したデータを送信します。送信したいデータのいったバッファと、そのデータ長、送信に利用するエンドポイント番号を引数に送信処理を呼び出します。この API は、サンプル・ドライバから提供されます。

## 8. 関数仕様

サンプル・ドライバで使用する関数仕様を次に示します。

main.c

void main( void )	
関数概要	メイン処理
パラメータ	void
戻り値	void
処理概要	・初期化
	・エンドポイント 0 監視
	・エンドポイント 1 監視
	・サンプル・アプリケーション処理

void init( void )	
関数概要	初期化処理
パラメータ	void
戻り値	void
処理概要	・CPU 初期化
	・ROM 化パッケージ用初期化
	・USB 初期化

void cpu_init( void )	
関数概要	CPU 初期化処理
パラメータ	void
戻り値	void
処理概要	・ロックアップ時間ウェイト
	・システム・ウェイト設定
	・ウォッチドッグ・タイマ設定
	・PLL ON(システムクロックジェネレート)
	・クロック分周設定
	・USB クロック設定

void romp_init( void )	
関数概要	ROM 化パッケージ用初期化処理
パラメータ	void
戻り値	void
処理概要	<ul style="list-style-type: none"> <li>・コピー関数実行</li> </ul> _rcopy(&label, number)は、label の示すアドレス以降に存在する rompsec セクション内の情報をもとに、コピーしたいセクション番号 number の初期値データ、または RAM に配置するテキストを RAM 領域に 1 バイトずつコピーします。number に-1 を指定した場合、rompsec セクション内のすべてのセクションをコピーします。

void app_main( void )	
関数概要	サンプル・アプリケーション処理
パラメータ	void
戻り値	void
処理概要	<ul style="list-style-type: none"> <li>・受信データ長取得</li> <li>・データ受信</li> <li>・大文字⇔小文字変換</li> <li>・データ送信</li> </ul>

#### usb850.c

void usb_init( void )	
関数概要	USB ファンクション・コントローラ初期化処理
パラメータ	void
戻り値	void
処理概要	<ul style="list-style-type: none"> <li>・USB ファンクション・コントローラ関連レジスタの初期設定 詳細は別紙「レジスタ設定(CDC).xls」参照</li> <li>・CDC クラスリクエスト関数登録</li> <li>・D+信号プルアップ</li> </ul>

void intusb0b( void )	
関数概要	エンドポイント 0 監視処理
パラメータ	void
戻り値	void
処理概要	<ul style="list-style-type: none"> <li>・CPUDEC 割り込みの監視</li> <li>・クラスリクエストの該当処理実行</li> </ul>

void intusb1b( void )	
関数概要	エンドポイント 1 監視処理
パラメータ	void
戻り値	void
処理概要	・BKO1DT 割り込みの監視 ・データ受信フラグのセット

INT32 usbf850_data_send( UINT8* data, INT32 len, INT8 ep )			
関数概要	USB データ送信処理		
パラメータ	UINT8 *	data	:送信データバッファポインタ
	INT32	len	:データ長
	INT8	ep	:データ送信エンドポイント指定(エンドポイント番号)
戻り値	処理結果		
	DEV_OK	:正常終了	
	DEV_ERROR	:異常終了	
処理概要	・送信データバッファに格納されているデータを、1byte ずつ指定されたエンドポイント用のデータ送信 FIFO に格納します。		

INT32 usbf850_data_receive( UINT8* data, INT32 len, INT8 ep )			
関数概要	USB データ受信処理		
パラメータ	UINT8 *	data	:受信データ格納バッファポインタ
	INT32	len	:データ長
	INT8	ep	:データ受信エンドポイント指定(エンドポイント番号)
戻り値	処理結果		
	DEV_OK	:正常終了	
	DEV_ERROR	:異常終了	
処理概要	・指定されたエンドポイント用のデータ受信 FIFO から 1byte ずつデータを読み込み、受信データ格納バッファに格納します。		

void usbf850_rdata_length( INT32* len, INT8 ep )			
関数概要	USB 受信データ長取得		
パラメータ	INT32 *	len	:受信データ長格納ポインタ
	INT8	ep	:データ受信エンドポイント指定(エンドポイント番号)
戻り値	void		
処理概要	・指定されたエンドポイントの受信データ長を取得します。		

void usbf850_sendnullEP0( void )	
関数概要	エンドポイント 0 用 NULL パケット送信処理
パラメータ	void
戻り値	void
処理概要	・FIFO クリア
	・エンドポイント 0 用データエンドビット ON

void usbf850_sendstallEP0( void )	
関数概要	エンドポイント 0 用 STALL 送信処理
パラメータ	void
戻り値	void
処理概要	・エンドポイント 0 用 STALL ビット ON

void usbf850_standardreq( void )	
関数概要	USB ファンクション・コントローラが自動応答しない標準リクエスト処理
パラメータ	void
戻り値	void
処理概要	・リクエストが GET_DESCRIPTOR の場合は GET_DESCRIPTOR 処理を実行
	・それ以外はエンドポイント 0 用 STALL 処理を実行

void usbf850_getdesc( void )	
関数概要	GET_DESCRIPTOR 処理
パラメータ	void
戻り値	void
処理概要	・StringDescriptor を要求している場合は StringDescriptor をエンドポイント 0 から送信します。
	・それ以外はエンドポイント 0 用 STALL 処理を実行

void usbf850_sstall_ctrl( void )	
関数概要	STALL 制御処理
パラメータ	void
戻り値	void
処理概要	・エンドポイント 0 用 STALL 処理を実行



usb850\_communication.c

void usb850_send_encapsulated_command( void )	
関数概要	Send_Encapsulated_Command 処理
パラメータ	void
戻り値	void
処理概要	・エンドポイント 0 で受信したデータをバルクイン・エンドポイントで送信

void usb850_set_line_coding( void )	
関数概要	Set_Line_Coding 処理
パラメータ	void
戻り値	void
処理概要	・エンドポイント 0 でシリアルポート設定値を受信 ・エンドポイント 0 用 NULL パケット送信処理実行

void usb850_get_line_coding( void )	
関数概要	Get_Line_Coding 処理
パラメータ	void
戻り値	void
処理概要	・シリアルポート設定値をエンドポイント 0 から送信

void usb850_set_control_line_state( void )	
関数概要	Set_Control_Line_State 処理
パラメータ	void
戻り値	void
処理概要	・エンドポイント 0 用 NULL パケット送信処理実行

void usb850_setfunction_communication( void )	
関数概要	CDC クラスリクエスト関数登録処理
パラメータ	void
戻り値	void
処理概要	・各クラスリクエスト関数のアドレスを登録

## 9. レジスタ設定

## 9.1. CPU 関連レジスタの設定

アドレス	レジスタ名称	レジスタ略称	設定値	備考
0xFFFF F820	ロック・レジスタ	LOCKR	-	LOCK(0):PLL のロック状態 0:ロック状態 1:アンロック状態
0xFFFF F06E	システム・ウェイト・コントロールレジスタ	VSWC	0x12	内蔵周辺 I/O レジスタに対するバス・アクセスのウェイト制御 12H:ウェイト数 3
0xFFFF F9FC	オンチップ・デバッグ・モード設定レジスタ	OCDM	0x00	<b>特定レジスタ</b> OCDM(0):動作モード _DRST 端子がロウ・レベルの場合: 通常動作モード(オンチップ・デバッグ・モード用端子として使用) _DRST 端子がハイ・レベルの場合: オンチップ・デバッグ・モード(オンチップ・デバッグ・モード用端子として使用)
0xFFFF F6D0	ウォッチドッグ・タイマ・モードレジスタ 2	WDTM2	0x00	00H:タイマ停止
0xFFFF F822	クロック・コントロール・レジスタ	CKC	0x0B	<b>特定レジスタ</b> CKDIV0(0):PLL モード時の内部システムクロック 1:8 通倍 ※CKC レジスタには必ず 0BH を設定すること。
0xFFFF F82C	PLL コントロール・レジスタ	PLLCTL	0x03	SELPLL(1):CPU 動作クロック選択レジスタ 1:PLL モード PLLON(0):PLL 動作停止レジスタ 1:PLL 動作(PLL 動作開始後、周波数が安定するまでの所定のロックアップ時間が必要)
0xFFFF F828	プロセッサ・クロック・コントロール・レジスタ	PCG	0x00	<b>特定レジスタ</b> FRC(7):サブクロック発振回路の内蔵帰還抵抗の制御 使用する(0)

				MCK(6):メインクロック発振回路の制御 発振する(0)
				MFRC(5):メイン・クロック発振回路の内蔵帰還抵抗の制御 使用する(0)
				CK3-0(3-0):内部システムクロック(fclk)/CPUクロック(fcpu)の選択 fxx(0000)
0xFFFF	F484	データ・ウェイト・コントロール・レジスタ 0	DWC0	0x7777
				DW32-30(14-12):CS3 の挿入ウェイト・ステート数 1H:7
				DW22-20(10-8):CS2 の挿入ウェイト・ステート数 1H:7
				DW12-10(6-4):CS1 の挿入ウェイト・ステート数 7H:7 ※CS1 は内部で使用する為、値固定
				DW02-00(3-0):CS0 の挿入ウェイト・ステート数 1H:7
0xFFFF	FE40	USB クロック選択レジスタ	UCKSEL	0x02
0xFFFF	FE41	UFUCLK マスクレジスタ	UFCKMSK	0x00
0xFFFF	FE42	UHPCLK/UFHUCLK マスクレジスタ	UHCKMSK	0x00
				UHUCLK:UCLK、UHPCLK:UHPCLK、UFUCLK:UCLK、UHPCLK:UHUCLK USB Host バッファ&許可:USBH 使用、USB Function バッファ&許可:USBF 使用

## 9.2. USB 関連レジスタの設定

アドレス	マクロ名称	内容	初期化処理後	
			値	設定
0x00200002	UF0E0NA	UF0 EP0NAKALL レジスタ	0x01	[ビット 0:EP0NKA] Endpoint0 への SETUP トランザクション以外の NAK を送信する (1)
0x00200144	UF0DSTL	UF0 デバイス・ステータス・レジスタ L	0x00	[ビット 1:RMWK] デバイスによるリモート・ウェイクアップ機能の使用禁止(0) [ビット 0:SFPW] デバイスがバス・パワード(0)
0x0020014C	UF0E0SL	UF0 EP0 ステータス・レジスタ L	0x00	[ビット 0:E0HALT] Endpoint0 のストール状態クリア(0)
0x00200150	UF0E1SL	UF0 EP1 ステータス・レジスタ L	0x00	[ビット 0:E1HALT] Endpoint1 のストール状態クリア(0)
0x00200154	UF0E2SL	UF0 EP2 ステータス・レジスタ L	0x00	[ビット 0:E2HALT] Endpoint2 のストール状態クリア(0)
0x002001A0	UF0DSCL	UF0 ディスクリプタ・レングス・レジスタ	0x20	[ビット 7-0:DPL7-0] GET_DESCRIPTOR Configuration リクエストで返信する全ディスクリプタのバイト数-1 の値
0x002001A2	UF0DD0	UF0 デバイス・ディスクリプタ・レジスタ	0x12	bLength
0x002001A4	UF0DD1		0x01	bDescriptorType
0x002001A6	UF0DD2		0x00	bcdUSB Lo
0x002001A8	UF0DD3		0x02	bcdUSB Hi
0x002001AA	UF0DD4		0x02	bDeviceClass
0x002001AC	UF0DD5		0x00	bDeviceSubClass
0x002001AE	UF0DD6		0x00	bDeviceProtocol
0x002001B0	UF0DD7		0x40	bMaxPacketSize0
0x002001B2	UF0DD8		0x09	idVendor Lo
0x002001B4	UF0DD9		0x04	idVendor Hi
0x002001B6	UF0DD10		0xD0	idProduct Lo
0x002001B8	UF0DD11		0x01	idProduct Hi
0x002001BA	UF0DD12		0x01	bcdDevice Lo
0x002001BC	UF0DD13		0x00	bcdDevice Hi

0x002001BE	UF0DD14		0x01	iManufacture
0x002001C0	UF0DD15		0x02	iProduct
0x002001C2	UF0DD16		0x03	iSerialNumber
0x002001C4	UF0DD17		0x01	bNumConfigurations
0x002001C6	UF0CIE0	UF0 コンフィギュレーション/ インターフェース/ エンドポイント・ディスクリプタ・レジスタ  (コンフィギュレーションディスクリプタ部)	0x09	bLength
0x002001C8	UF0CIE1		0x02	bDescriptorType
0x002001CA	UF0CIE2		0x30	wTotalLength Lo
0x002001CC	UF0CIE3		0x00	wTotalLength Hi
0x002001CE	UF0CIE4		0x02	bNumInterface
0x002001D0	UF0CIE5		0x01	bConfigurationValue
0x002001D2	UF0CIE6		0x00	iConfiguration
0x002001D4	UF0CIE7		0x80	bmAttributes
0x002001D6	UF0CIE8		0x1B	MaxPower
0x002001D8	UF0CIE9		UF0 コンフィギュレーション/ インターフェース/ エンドポイント・ディスクリプタ・レジスタ  (インターフェース 0 部)	0x09
0x002001DA	UF0CIE10	0x04		bDescriptorType
0x002001DC	UF0CIE11	0x00		bInterfacenumber
0x002001DE	UF0CIE12	0x00		bAlternateSetting
0x002001E0	UF0CIE13	0x01		bNumEndpoints
0x002001E2	UF0CIE14	0x02		bInterfaceClass
0x002001E4	UF0CIE15	0x02		bInterfaceSubClass
0x002001E6	UF0CIE16	0x00		bInterfaceProtocol
0x002001E8	UF0CIE17	0x00		iInterface
0x002001EA	UF0CIE18	UF0 コンフィギュレーション/ インターフェース/ エンドポイント・ディスクリプタ・レジスタ  (エンドポイント1部)		0x07
0x002001EC	UF0CIE19		0x05	bDescriptorType
0x002001EE	UF0CIE20		0x87	bEndpointAddress
0x002001F0	UF0CIE21		0x03	bmAttributes
0x002001F2	UF0CIE22		0x08	wMaxPacketSize Lo
0x002001F4	UF0CIE23		0x00	wMaxPacketSize Hi

0x002001F6	UF0CIE24		0x0A	bInterval
0x002001F8	UF0CIE25	UF0 コンフィギュレーション/ インターフェース/ エンドポイント・ディスクリプタ・レジスタ  (インターフェース 0 部)	0x09	bLength
0x002001FA	UF0CIE26		0x04	bDescriptorType
0x002001FC	UF0CIE27		0x01	bInterfacenumber
0x002001FE	UF0CIE28		0x00	bAlternateSetting
0x00200200	UF0CIE29		0x02	bNumEndpoints
0x00200202	UF0CIE30		0x0A	bInterfaceClass
0x00200204	UF0CIE31		0x00	bInterfaceSubClass
0x00200206	UF0CIE32		0x00	bInterfaceProtocol
0x00200208	UF0CIE33		0x00	iInterface
0x0020020A	UF0CIE34		UF0 コンフィギュレーション/ インターフェース/ エンドポイント・ディスクリプタ・レジスタ  (エンドポイント1部)	0x07
0x0020020C	UF0CIE35	0x05		bDescriptorType
0x0020020E	UF0CIE36	0x81		bEndpointAddress
0x00200210	UF0CIE37	0x02		bmAttributes
0x00200212	UF0CIE38	0x40		wMaxPacketSize Lo
0x00200214	UF0CIE39	0x00		wMaxPacketSize Hi
0x00200216	UF0CIE40	0x00		bInterval
0x002001F8	UF0CIE41	UF0 コンフィギュレーション/ インターフェース/ エンドポイント・ディスクリプタ・レジスタ  (エンドポイント 2 部)	0x07	bLength
0x002001FA	UF0CIE42		0x05	bDescriptorType
0x002001FC	UF0CIE43		0x02	bEndpointAddress
0x002001FE	UF0CIE44		0x02	bmAttributes
0x00200200	UF0CIE45		0x40	wMaxPacketSize Lo
0x00200202	UF0CIE46		0x00	wMaxPacketSize Hi
0x00200204	UF0CIE47		0x00	bInterval
0x00200074	UF0MODC	UF0 モード・コントロール・レジスタ	0x00	[ビット 6:GDCGDST] GET_DESCRIPTOR Configuration リクエストは自動のまま(0)
0x00200080	UF0AIFN	UF0 アクティブ・インターフェース・ナンバー・レジスタ	0x00	[ビット 7:ADDIF] Interface 0 だけをサポート(0)
				[ビット 1-0:IFNO1、0] サポートする Interface 番号

0x00200084	UF0AAS	UF0 アクティブ・オルタナティブ・セッティング・レジスタ	0x00	[ビット 3-1:AL5ST3,2,1] Alternative Setting0(0)
0x00200086	UF0E1IM	UF0 エンドポイント 1 インターフェース・マッピング・レジスタ	0x40	[ビット 7-5:E1EN2,1,0] Interface 1, Alternative Setting(010B)
				[ビット 4:E12AL1] CONF ビット=1 で Alternate Setting1 に設定されても有効にならない(0)
				[ビット 3-0:E15AL4-1] CONF ビット=1 で Alternate Setting n に設定されても有効にならない(0)
0x00200088	UF0E2IM	UF0 エンドポイント 2 インターフェース・マッピング・レジスタ	0x40	[ビット 7-5:E2EN2,1,0] Interface 1, Alternative Setting(010B)
				[ビット 4:E22AL1] CONF ビット=1 で Alternate Setting1 に設定されても有効にならない(0)
				[ビット 3-0:E25AL4-1] CONF ビット=1 で Alternate Setting n に設定されても有効にならない(0)
0x00200092	UF0E7IM	UF0 エンドポイント 7 インターフェース・マッピング・レジスタ	0x20	[ビット 7-5:E7EN2,1,0] Interface 0, Alternative Setting(001B)
				[ビット 4:E72AL1] CONF ビット=1 で Alternate Setting1 に設定されても有効にならない(0)
				[ビット 3-0:E75AL4-1] CONF ビット=1 で Alternate Setting n に設定されても有効にならない(0)
0x0020003C	UF0IC0	UF0 割り込みクリアレジスタ 0	0x20	[ビット 7:BUSRSTC] Bus Reset 割り込み要求クリア(0)
				[ビット 6:RSUSPDC] Resume/Suspend 割り込み要求クリア(0)
				[ビット 4:SHORTC] Short 割り込み要求をクリア(0)
				[ビット 3:DMAEDC] DMA_END 割り込み要求をクリア(0)
				[ビット 2:SETRQC] SET_RQ 割り込み要求をクリア(0)
				[ビット 1:CLRRQC] CLR_RQ 割り込み要求をクリア(0)
				[ビット 0:EPHALTC] EP_Halt 割り込み要求をクリア(0)
0x0020003E	UF0IC1	UF0 割り込みクリアレジスタ 1	0x80	[ビット 6:E0INC] EP0IN 割り込み要求をクリア(0)



				[ビット 5: E0INDTC] EP0INDT 割り込み要求をクリア(0)
				[ビット 4: E0ODTC] EP0OUTDT 割り込み要求をクリア(0)
				[ビット 3: SUCESC] Success 割り込み要求をクリア(0)
				[ビット 2: STGC] Stg 割り込み要求をクリア(0)
				[ビット 1: PROTC] Protect 割り込み要求をクリア(0)
				[ビット 0: CPUDEC] CPUDEC 割り込み要求をクリア(0)
0x00200040	UF0IC2	UF0 割り込みクリアレジスタ 2	0x0C	[ビット 7: BKI2INC] BLK21IN 割り込み要求をクリア(0)
				[ビット 6: BKI2DTC] BLKI2DT 割り込み要求をクリア(0)
				[ビット 5: BKI1INC] BLKI1IN 割り込み要求をクリア(0)
				[ビット 4: BKI1DTC] BLKI1DT 割り込み要求をクリア(0)
				[ビット 1: IT2DTC] INT2DT 割り込み要求をクリア(0)
				[ビット 0: IT1DTC] INT1DT 割り込み要求をクリア(0)
0x00200042	UF0IC3	UF0 割り込みクリアレジスタ 3	0x00	[ビット 7: BKO2FLC] BLKO2FL 割り込み要求をクリア(0)
				[ビット 6: BKO2NLC] BLKO2NL 割り込み要求をクリア(0)
				[ビット 5: BKO2NAKC] BLKO2NK 割り込み要求をクリア(0)
				[ビット 0: BKO2DTC] BLKO2DT 割り込み要求をクリア(0)
				[ビット 3: BKO1FLC] BLKO1FL 割り込み要求をクリア(0)
				[ビット 2: BKO1NLC] BLKO1NL 割り込み要求をクリア(0)
				[ビット 1: BKO1NAKC] BLKO1NK 割り込み要求をクリア(0)
				[ビット 0: BKO1DTC] BLKO1DT 割り込み要求をクリア(0)
0x00200044	UF0IC4	UF0 割り込みクリアレジスタ 4	0xDF	[ビット 5: SETINTC] SET_INT 割り込み要求をクリア
0x00200060	UF0FIC0	UF0 FIFO クリアレジスタ 0	0xFF	[ビット 7: BKI2SC] UF0BI2 レジスタの SIE 側 FIFO のみクリア(1)
				[ビット 6: BKI2CC] UF0BI2 レジスタの CPU 側 FIFO のみクリア(1)
				[ビット 5: BKI1SC] UF0BI1 レジスタの SIE 側 FIFO のみクリア(1)
				[ビット 4: BKI1CC] UF0BI1 レジスタの CPU 側 FIFO のみクリア(1)
				[ビット 3: ITR2C] UF0INT2 レジスタクリア(1)
				[ビット 2: ITR1C] UF0INT1 レジスタクリア(1)
				[ビット 1: EP0WC] UF0E0W レジスタクリア(1)

				[ビット0:EP0RC] UF0E0Rレジスタクリア(1)
0x00200062	UF0FIC1	UF0 FIFO クリアレジスタ 1	0x0F	[ビット3:BKO2C] UF0B02レジスタのSIE側/CPU側の両FIFOをクリア(1)
				[ビット2:BKO2CC] UF0B02レジスタのCPU側だけのFIFOをクリア(1)
				[ビット1:BKO1C] UF0B01レジスタのSIE側/CPU側の両FIFOをクリア(1)
				[ビット0:BKO1CC] UF0B01レジスタのCPU側だけのFIFOをクリア(1)
0x0020002E	UF0IM0	UF0 割り込みマスクレジスタ 0	0xDF	[ビット7:BUSRSTM] Bus Reset 割り込みをマスク(1)
				[ビット6:RSUSPDM] Resume/Suspend 割り込みをマスク(1)
				[ビット4:SHORTM] Short 割り込みをマスク(1)
				[ビット3:DMAEDM] DMA_END 割り込みをマスク(1)
				[ビット2:SETRQM] SET_RQ 割り込みをマスクしない(0)
				[ビット1:CLRRQM] CLR_RQ 割り込みをマスク(1)
[ビット0:EPHALTM] EP_Halt 割り込みをマスク(1)				
0x00200030	UF0IM1	UF0 割り込みマスクレジスタ 1	0x7E	[ビット6:E0INM] EP0IN 割り込みをマスク(1)
				[ビット5:E0INDTM] EPINDT 割り込みをマスク(1)
				[ビット4:E0ODTM] EP0OUTDT 割り込みをマスク(1)
				[ビット3:SUCESM] Success 割り込みをマスク(1)
				[ビット2:STGM] Stg 割り込みをマスク(1)
				[ビット1:PROTM] Protect 割り込みをマスク(1)
				[ビット0:CPUDECM] CPUDEC 割り込みをマスクしない(0)
0x00200032	UF0IM2	UF0 割り込みマスクレジスタ 2	0xF3	[ビット7:BK21INM] BLK12IN 割り込みをマスク(1)
				[ビット6:BK12DTM] BLK2DT 割り込みをマスク(1)
				[ビット1:IT2DTM] INT2DT 割り込みをマスクしない(0)
				[ビット5:BK11INM] BLK11IN 割り込みをマスク(1)
				[ビット4:BK11DTM] BLK1DT 割り込みをマスク(1)
				[ビット0:IT1DTM] INT1DT 割り込みをマスクしない(0)
0x00200034	UF0IM3	UF0 割り込みマスクレジスタ 3	0xFE	[ビット7:BKO2FLM] BLKO2FL 割り込みをマスク(1)
				[ビット6:BKO2NLM] BLKO2NL 割り込みをマスク(1)
				[ビット5:BLKO2NAKM] BLKO2NK 割り込みをマスク(1)

				[ビット4:BKO2DTM] BLKO2DT 割り込みをマスクしない(0)
				[ビット3:BKO1FLM] BLKO1FL 割り込みをマスク(1)
				[ビット2:BKO1NLM] BLKO1NL 割り込みをマスク(1)
				[ビット1:BLKO1NAKM] BLKO1NK 割り込みをマスク(1)
				[ビット0:BKO1DTM] BLKO1DT 割り込みをマスクしない(0)
0x00200036	UF0IM4	UF0 INT Mask4 Register	0x20	[ビット5:SETINTM] SET_INT 割り込みをマスクする(1)
0xFFFFF428	PM4	ポート 4 モード・レジスタ	0x FC	[ビット2:PM42] 入力モード(1)
				[ビット1:PM41] 出力モード(0)
				[ビット0:PM40] 出力モード(0)
0xFFFFF408	P4	ポート 4・レジスタ	0x02	[ビット2:P42] (出力モード時)0 出力(0)
				[ビット1:P41] (出力モード時)1 出力(1)
				[ビット0:P40] (出力モード時)1 出力(1)

### 9.3. ディスクリプタ設定

#### デバイス・ディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x12
bDescriptor	1	ディスクリプタタイプ	0x01
bcdUSB	2	BCD 表現の USB 仕様リリース番号	0x0200
bDeviceClass	1	クラス・コード 0x00: クラス無し、0x01~0xFE: 特定、0xFF: ベンダ	0x02
bDeviceSubClass	1	サブ・クラスコード	0x00
bDeviceProtocol	1	プロトコル・コード 0x00: 固有プロトコル使用せず、0xFF: ベンダ固有	0x00
bMaxPacketSize0	1	エンドポイント 0 の最大パケットサイズ	0x40
idVendor	2	ベンダ ID(USB IF が割り当て)	0x0409
idProduct	2	プロダクト ID(ベンダが割り当てる)	0x01D0
bcdDevice	2	BCD 表現のデバイスのリリース番号	0x0001
iManufacture	1	製造者を表すstring・ディスクリプタへのインデックス	0x01
iProduct	1	製品を表すstring・ディスクリプタへのインデックス	0x02
iSerialNumber	1	デバイスの製造番号を表すstring・ディスクリプタへのインデックス	0x03
bNumConfigurations	1	構成可能な数	0x01

コンフィグレーション・ディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x09
bDescriptor	1	ディスクリプタタイプ	0x02
wTotalLength	2	構成全体(構成、インターフェース、エンドポイント、そのほかのディスクリプタ)の長さ	0x0030
bNumInterfaces	1	構成のもつインターフェースの数	0x02
bConfigurationValue	1	SetConfiguration リクエストで、この構成を選択するための引数値(1 以上)	0x01
iConfiguration	1	構成を表すstring・ディスクリプタへのインデックス	0x00
bmAttributes	1	構成の特性、ビット単位で意味づけ D7: "1" D6: 自己電源 D5: リモート・ウェークアップ D4-D0: 予約(0)	0x80
bMaxPower	1	最大バス電力消費量を 2mA 単位で指定	0x1B

## インターフェース・ディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x09
bDescriptor	1	ディスクリプタタイプ	0x04
bInterfaceNumer	1	構成の中で、このインターフェースを表すインデックス番号(0 ベース)	0x00
bAlternateSetting	1	SetInterface リクエストで、代替設定を選択するための引数値	0x00
bNumEndpoints	1	(エンドポイント 0 を除く)インターフェースの持つエンドポイント数	0x01
bInterfaceClass	1	クラス・コード 0x00: クラスなし、0x01~0xFE: 特定、0xFF: ベンダ	0x02
bInterfaceSubClass	1	サブクラス・コード	0x02
bInterfaceProtocol	1	プロトコル・コード 0x00: 固有プロトコル使用せず、0xFF ベンダ固有	0x00
iInterface	1	このインターフェースを表すストリング・ディスクリプタへのインデックス	0x00

エンドポイント・ディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x07
bDescriptor	1	ディスクリプタタイプ	0x05
bEndpointAddress	1	エンドポイント・アドレス。ビット単位で意味づけ D7: 方向 0: OUT 1: IN D6-D4: 予約(0) D4-D0: エンドポイント番号	0x87
bmAttributes	1	属性(ビット単位で意味づけ) D1-D0: 転送タイプ 0: コントロール 1: アイソクロナス 2: バルク 3: インタラプト ※D5~D2 はアイソクロナス・エンドポイントのみで使用 D3-D2: 同期タイプ 0: 同期なし 1: 非同期 2: アダプティブ 3: 同期 D5-D4: ユーセージ・タイプ 0: データ・エンドポイント 1: フィードバック・エンドポイント 2: 従属的なフィードバック・エンドポイント 3: (予約)	0x03

wMaxPacketSize	2	<p>ペイロード・サイズ指定(ビットで意味づけ)  D10-D0: 最大パケット・サイズ  D12-D11: <math>\mu</math> フレームあたりの追加的なトランザクション数(HS のアイソクロナスとインタラプトのみ)</p> <p>0: 追加なし(1トランザクション/<math>\mu</math> フレーム)  1: 1 つ(2トランザクション/<math>\mu</math> フレーム)  2: 2 つ(3トランザクション/<math>\mu</math> フレーム)  3: 未使用(予約)</p>	0x0008
bInterval	1	<p>データ転送のエンドポイントをポーリング間隔  フル/ロー・スピード・インタラプト: ms 単位(フレーム数)で指定  ハイ・スピード・アイソクロナス/インタラプト: <math>\mu</math> フレーム単位で 2 の(N-1)乗の N を指定  (例えば、bInterval が 4 の場合、8 <math>\mu</math> フレームに 1 回ポーリング)  フル・スピード・アイソクロナス: 1ms 単位(フレーム数)で 2 の(N-1)乗の N を指定  ハイ・スピード・バルク/コントロール: エンドポイントの最大 NAK レートを <math>\mu</math> フレーム単位で指定  値 0 は OUT/DATA トランザクションで NAK 応答しないことを意味</p>	0x0A



## インターフェース・ディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x09
bDescriptor	1	ディスクリプタタイプ	0x04
bInterfaceNumer	1	構成の中で、このインターフェースを表すインデックス番号(0 ベース)	0x01
bAlternateSetting	1	SetInterface で、代替設定を選択するための引数値	0x00
bNumEndpoints	1	(エンドポイント 0 を除く)インターフェースの持つエンドポイント数	0x02
bInterfaceClass	1	クラス・コード 0x00: クラスなし、0x01~0xFE: 特定、0xFF: ベンダ	0x0A
bInterfaceSubClass	1	サブクラス・コード	0x00
bInterfaceProtocol	1	プロトコル・コード 0x00: 固有プロトコル使用せず、0xFF ベンダ固有	0x00
iInterface	1	このインターフェースを表すストリング・ディスクリプタへのインデックス	0x00

エンドポイント・ディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x07
bDescriptor	1	ディスクリプタタイプ	0x05
bEndpointAddress	1	エンドポイント・アドレス。ビット単位で意味づけ D7: 方向 0: OUT 1: IN D6-D4: 予約(0) D4-D0: エンドポイント番号	0x81
bmAttributes	1	属性(ビット単位で意味づけ) D1-D0: 転送タイプ 0: コントロール 1: アイソクロナス 2: バルク 3: インタラプト ※D5~D2 はアイソクロナス・エンドポイントのみで使用 D3-D2: 同期タイプ 0: 同期なし 1: 非同期 2: アダプティブ 3: 同期 D5-D4: ユーセージ・タイプ 0: データ・エンドポイント 1: フィードバック・エンドポイント 2: 従属的なフィードバック・エンドポイント 3: (予約)	0x02

wMaxPacketSize	2	<p>ペイロード・サイズ指定(ビットで意味づけ)  D10-D0: 最大パケット・サイズ  D12-D11: <math>\mu</math> フレームあたりの追加的なトランザクション数(HS のアイソクロナスとインタラプトのみ)</p> <p>0: 追加なし(1トランザクション/<math>\mu</math> フレーム)  1: 1 つ(2トランザクション/<math>\mu</math> フレーム)  2: 2 つ(3トランザクション/<math>\mu</math> フレーム)  3: 未使用(予約)</p>	0x0040
bInterval	1	<p>データ転送のエンドポイントをポーリング間隔  フル/ロー・スピード・インタラプト: ms 単位(フレーム数)で指定  ハイ・スピード・アイソクロナス/インタラプト: <math>\mu</math> フレーム単位で 2 の(N-1)乗の N を指定  (例えば、bInterval が 4 の場合、8 <math>\mu</math> フレームに 1 回ポーリング)  フル・スピード・アイソクロナス: 1ms 単位(フレーム数)で 2 の(N-1)乗の N を指定  ハイ・スピード・バルク/コントロール: エンドポイントの最大 NAK レートを <math>\mu</math> フレーム単位で指定  値 0 は OUT/DATA トランザクションで NAK 応答しないことを意味</p>	0x00

ストリングディスクリプタ

フィールド	サイズ	説明	設定値
bLength	1	ディスクリプタサイズ	0x07
bDescriptor	1	ディスクリプタタイプ	0x05
bString	-	Language Code:0x09,0x04	
		Manufacture:"NEC Electronics Co."	
		Product:"CDCDrv"	
		Serial Number:"0_98765432"	

## 10. USB シリアル変換サンプル・プログラムの構成

サンプル・プログラムのディレクトリ構成を次に示します。

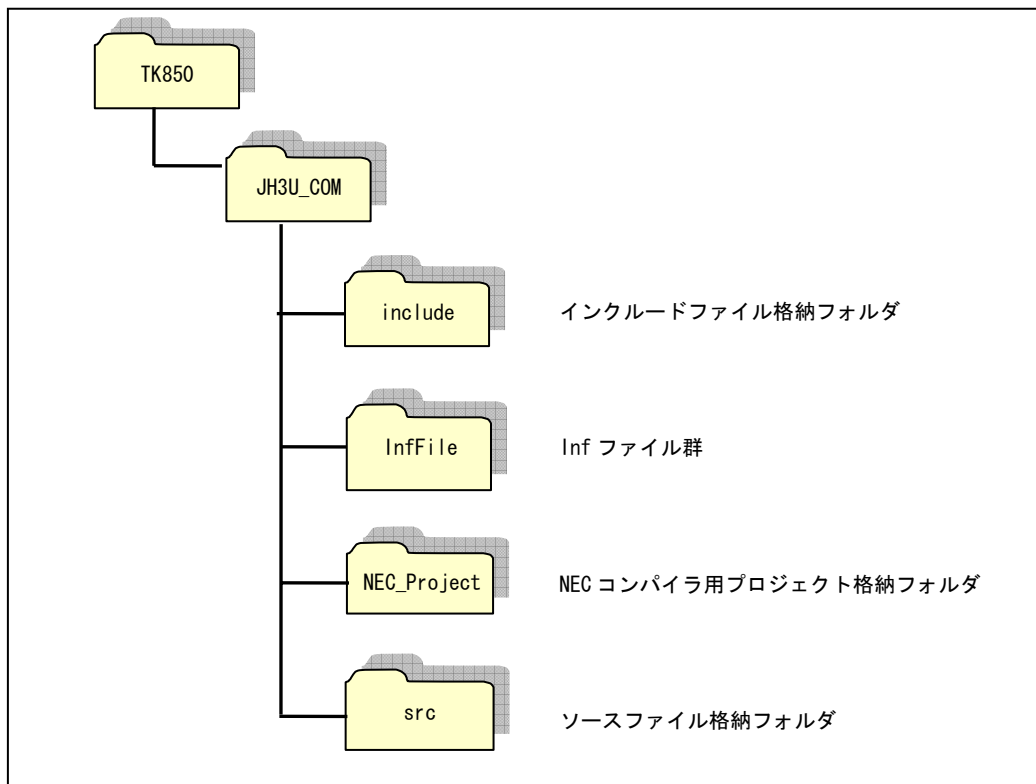


図 10-1 ディレクトリ構成

## 10.1. ドライバのインストール

デバッグポート(USB1)、 $\mu$  PD70F3769 USB ポート(USB3)はホスト PC 接続時にそれぞれ別のドライバが必要です。ドライバをインストールするために必要なファイルを次に示します。ドライバインストール時に該当するファイルを指定してください。

### [デバッグポート] (USB1 コネクタ)

MQB2SALL.inf 開発ツールのインストール時に”Starter Kit USB Driver”をインストール事で、デバッグポート用ドライバが「C:\Program Files\NECElectronicsTools\TK-driver」にインストールされます。接続時には「NECElectronics Starter Kit Virtual UART」として認識されません。

### [ $\mu$ PD70F3769 仮想 COM ポート] (USB3 コネクタ)

JG3H\_CDC\_XP.inf USB シリアル変換サンプル・プログラム用のドライバです。USB シリアル変換サンプル・プログラム(TK850\JH3U\_COM\InfFile)に同梱されています。使用する OS が WindowsXP の場合「XP\JG3H\_CDC\_XP.inf」、WindowsVista の場合「VISTA\JG3H\_CDC\_VISTA.inf」を選択してください。接続時には「NEC Electronics Jx3H Virtual UART」として認識されます。

以下はハードウェアの環境構築例です。

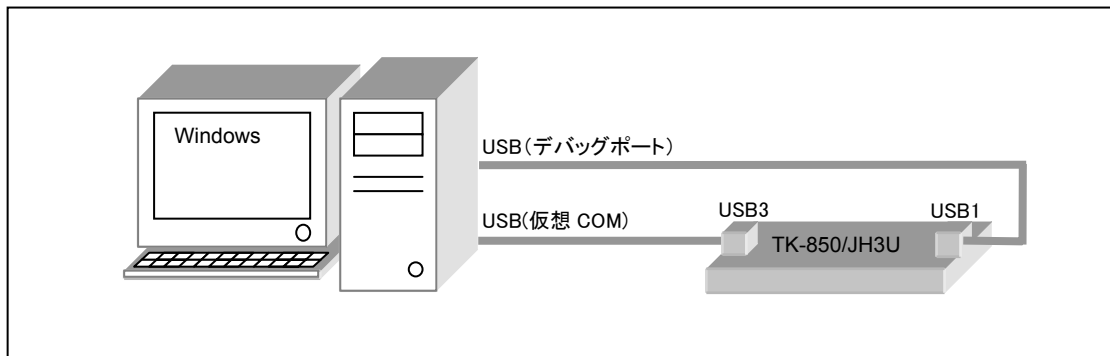


図 10-2 開発環境構築例(ハードウェア)

改版履歴

版数	日付	内容
1.0	2008/10/10	初版