

LCD-KIT+VS-CAM-01 サンプルプログラム解説 (RX64M)

第1版 2015年 9月29日

目次

1. 概要.....	1
1.1 概要.....	1
2. サンプルプログラムの構成.....	2
2.1 フォルダ構成.....	2
2.2 ファイル構成.....	3
3. LCD-KIT サンプルプログラム.....	4
3.1 動作説明.....	4
3.2 サンプルプログラムのダウンロード.....	6
3.3 開発環境使用時の各設定値.....	6
3.4 メモリマップ.....	8
3.5 LCD ダイレクトドライブ設定.....	9
3.6 デバッガ使用時の注意事項.....	10
3.7 プログラミングの注意事項.....	10

1. 概要

1.1 概要

本アプリケーションノートでは、AP-RX64M-0A (RX64M) を用いて、マルチタッチ対応静電容量式 LCD キット「LCD-KIT-D01」と CMOS カメラ「VS-CAM-01」を動作させるサンプルプログラムについて解説します。

本サンプルプログラムで使用する主な機能を以下に記します。

	機能	動作内容
LCD-KIT-D01	LCDパネル	カメラ画像の表示
	タッチパネル(静電容量式)	ポインタの移動
	バックライト	バックライトの点灯
	スイッチ	各種音声の出力開始
	スピーカ	音声出力
VS-CAM-01	CMOS カメラ	画像の取り込み
AP-RX64M-0A	ダイレクトメモリアクセス (EXDMAC0)	グラフィック表示 (画像データ転送)
	タイマパルスユニット (MTU1, 2, 3)	グラフィック表示 (同期信号)
	簡易 I2C 通信 (SC12)	LCD-KIT との通信
	PDC	CMOS カメラからの画像データ転送
	周期タイマ (CMT0, MTU2)	時間管理

2. サンプルプログラムの構成

2.1 フォルダ構成

サンプルプログラムは下記のようなフォルダ構成になっています。



2.2 ファイル構成

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。

<¥ap_rx64m_0a_lcdkit_cam フォルダ内>

ap_rx64m_0a_lcdkit_cam.mtpj … CS+用プロジェクトファイル

<¥ap_rx64m_0a_lcdkit_cam¥DefaultBuild フォルダ内>

ap_rx64m_0a_lcdkit_cam.abs … elf 形式オブジェクトファイル
 ap_rx64m_0a_lcdkit_cam.mot … モトローラ S フォーマット形式ファイル
 ap_rx64m_0a_lcdkit_cam.map … マップファイル

<¥ap_rx64m_0a_lcdkit_cam¥src フォルダ内>

AP_RX64M_0A.c … メイン処理
 cmoscsm.c … CMOS カメラ処理
 exdma.c … DMA 処理
 i2c.c … I2C 処理
 lcdkit.c … LCD-KIT 操作入力制御処理
 lcd_int.c … LCD-KIT 割り込み処理
 tmr.c … タイマ処理
 common.h … 共通ヘッダファイル
 vect.h … 割り込み関数プロトタイプ

r_bsp BSP フォルダ
 r_cmt_rx CMT モジュールフォルダ
 r_config 各モジュールの設定ファイルフォルダ
 r_dmac_api_rx DMA モジュールフォルダ
 r_pdc_rx PDC モジュールフォルダ
 r_riic_rx I2C モジュールフォルダ

<¥ap_rx64m_0a_lcdkit_cam¥data フォルダ内>

Sound0.bin … 音声出力用バイナリデータ 0
 Sound1.bin … 音声出力用バイナリデータ 1
 Sound2.bin … 音声出力用バイナリデータ 2

3. LCD-KIT サンプルプログラム

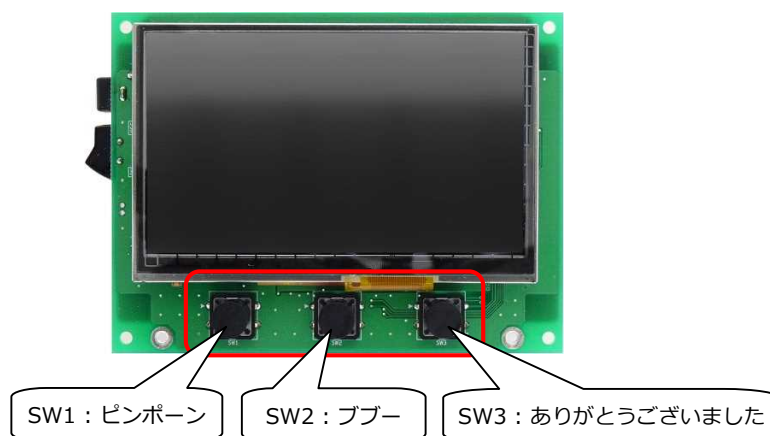
3.1 動作説明

本サンプルプログラムは下記の動作を行ないます。

- ① SW1～3 スイッチに対応する音声をLCD-KITのスピーカへ出力します
- ② LCDパネル CMOSカメラから読み込んだ画像をLCDへ表示します。
 CMOSカメラ
- ③ タッチパネル タッチパネル入力に対するポインタの表示を行います
 4点検出(青、赤、黄、緑のポインタ)

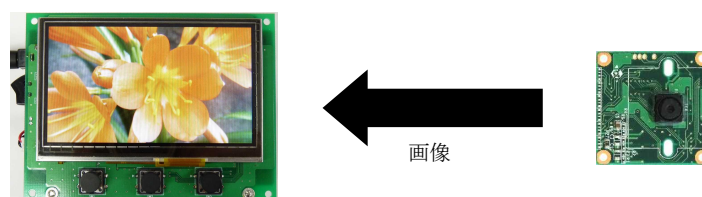
① SW1～3

SW1～3を押すと、スピーカから音声が出力されます。



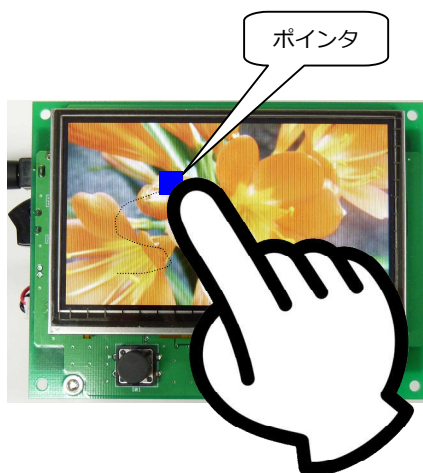
② LCD + CMOS カメラ

CMOSカメラで読み込んだ画像をLCDへ表示します。



③ タッチパネル

画面をタッチするとポインタの表示を行います。最大4点のマルチタッチ検出を行い、青・赤・緑・黄の順でポインタが表示されます。



3.2 サンプルプログラムのダウンロード

サンプルプログラムを CPU ボード上で実行するためには、ビルドしたサンプルプログラムの実行ファイルを CPU ボードにダウンロードする必要があります。

サンプルプログラムのビルド方法および CPU ボードにサンプルプログラムをダウンロードする方法については、アプリケーションノート「AN1526 RX 開発環境の使用法 (CS+, Renesas Flash Programmer)」に詳細な手順が記されていますので参照してください。

3.3 開発環境使用時の各設定値

開発環境を使用する際の、AP-RX64M-0A 固有の設定を以下に示します。

表内の「項目番号」はアプリケーションノート「AN1526 RX 開発環境の使用法 (CS+, Renesas Flash Programmer)」内で示されている項目番号を示していますので、対応したそれぞれの設定値を参照してください。

ビルド・動作確認方法		
項目名	項目番号	設定値
出力フォルダ	2-2	ap_rx64m_0a_lcdkit_cam¥DefaultBuild
モトローラファイル名	2-3	ap_rx64m_0a_lcdkit_cam¥DefaultBuild¥ap_rx64m_0a_lcdkit_cam.mot
アブソリュートファイル名	2-4	ap_rx64m_0a_lcdkit_cam¥DefaultBuild¥ap_rx64m_0a_lcdkit_cam.abs
マップファイル	2-5	ap_rx64m_0a_lcdkit_cam¥DefaultBuild¥ap_rx64m_0a_lcdkit_cam.map

Renesas Flash Programmer を使用した Flash 書き込み方法		
項目名	項目番号	設定値
ボード設定 (Flash 書き込み)	3-1	Fig 3-1 を参照
デバイス名	3-2	Generic BOOT Device
Flash に書き込むファイル	3-3	ap_rx64m_0a_lcdkit_cam¥DefaultBuild¥ap_rx64m_0a_lcdkit_cam.mot
ボード設定 (動作)	3-4	Fig 3-2 を参照

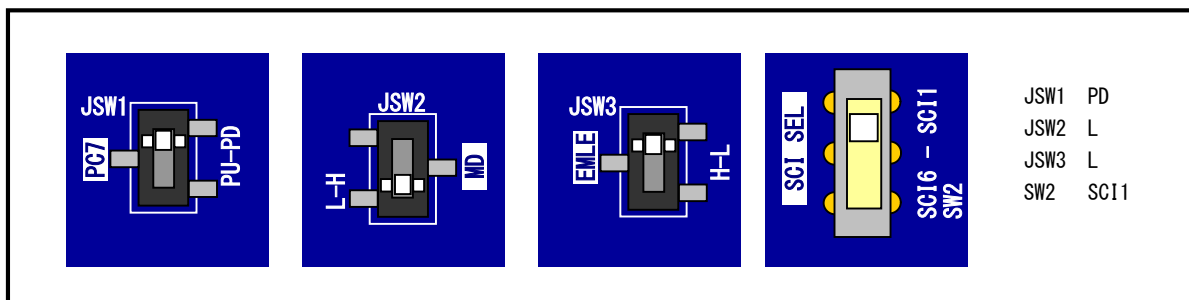


Fig 3-1 Flash 書き込み時のボード設定

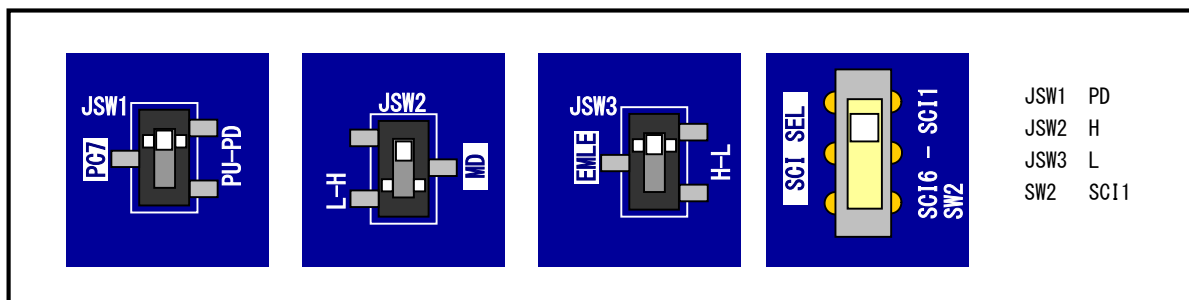


Fig 3-2 サンプルプログラム動作時のボード設定

E1 エミュレータを使用したデバッグ方法		
項目名	項目番号	設定値
ボード設定	4-1	Fig 3-3 を参照
JTAG クロック	4-10	16.5 (MHz)
EXTAL クロック	4-11	24 (MHz)

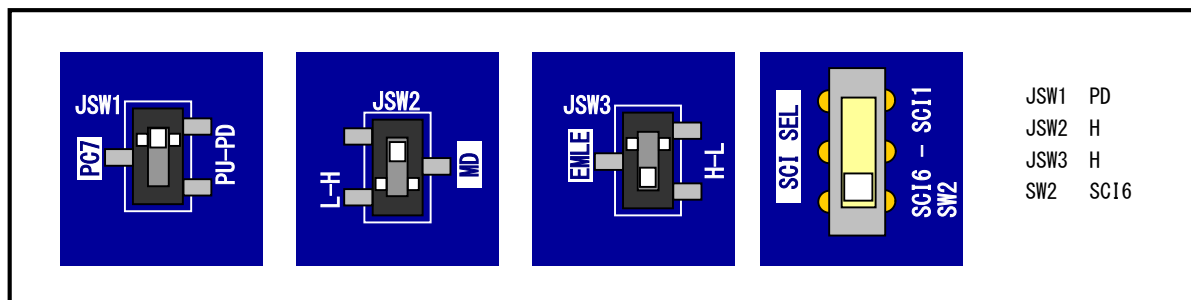
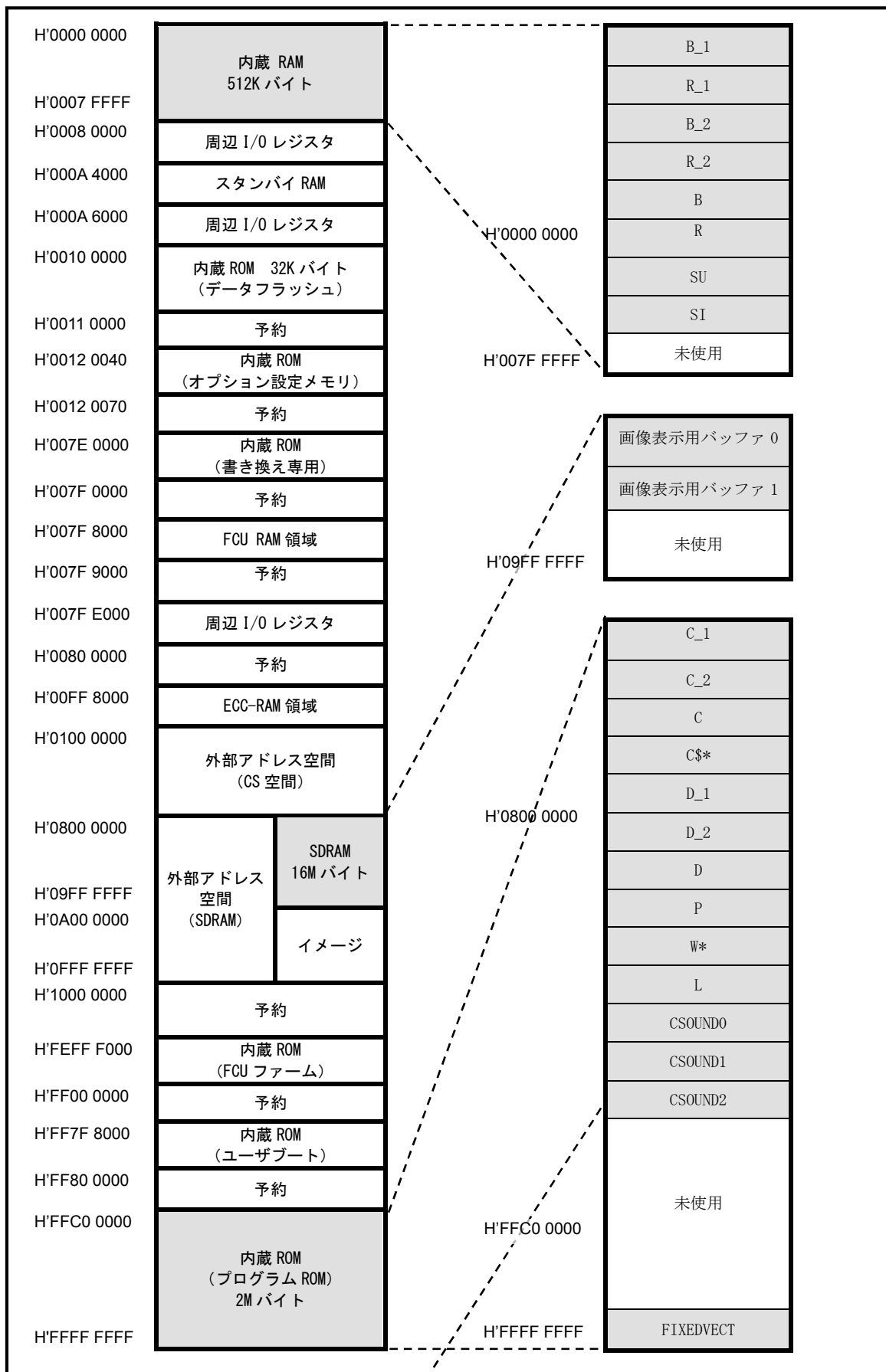


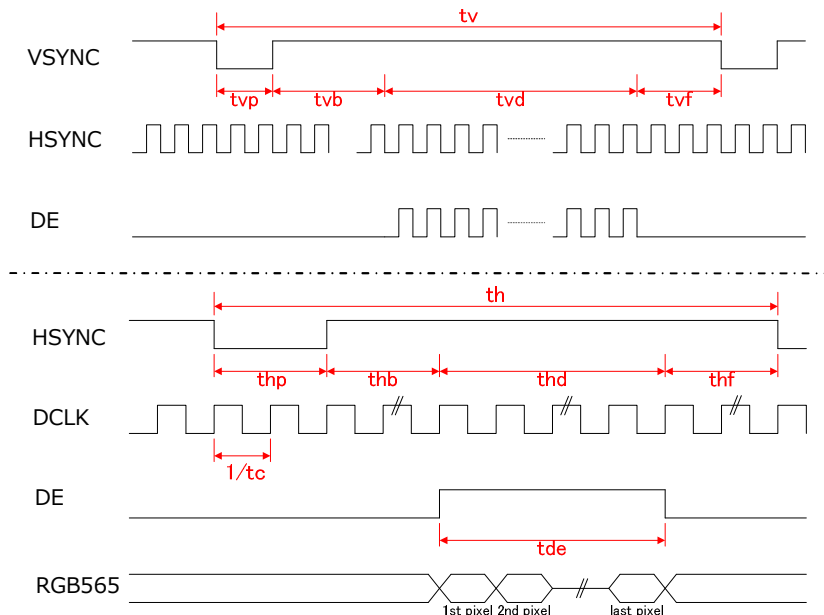
Fig 3-3 E1 エミュレータデバッグ時のボード設定

3.4 メモリマップ



3.5 LCD ダイレクトドライブ設定

サンプルプログラムでは、LCD ダイレクトドライブ方式を使用して、LCD の表示を行っています。
 LCD ダイレクトドライブの端子、及びタイミングは以下の設定されています。



・ 端子の設定

信号名	説明	使用ポート	端子設定	備考
DCLK	ドットクロック	P26	MTIOC2A	PWM モード 1
HSYNC	水平同期信号	PJ3	MTIOC3C	PWM モード 1
VSYNC	垂直同期信号	PJ5	PJ5	
DE	データイネーブル信号	P14	MTIOC3C	PWM モード 1
B[5..1]	Blue データ	PD[4..0]	D[4..0]	EXDMAC
G[5..0]	Green データ	PE[2..0]、PD[7..5]	D[10..5]	EXDMAC
R[5..1]	Red データ	PE[7..3]	D[15..11]	EXDMAC

・ タイミングの設定 (サンプルプログラム common.h 内)

27 行	#define LCD_CLKNUM 10	1/tc	$P\Phi / 10 = 120 \text{ MHz} / 4 = 12 \text{ MHz}$
28 行	#define LCD_DE_START 43	thp + thb	43 clk
29 行	#define LCD_DE_LENGTH 480	tde	480 clk
30 行	#define LCD_HSYNC_CYCLE 575	th	575 clk
31 行	#define LCD_HSYNC_LENGTH 41	thp	41 clk
33 行	#define LCD_VSYNC_LENGTH 10	tvp	10 HSYNC
34 行	#define LCD_VSYNC_BP (10 + 2)	tvp + tvb	10 + 2 = 12 HSYNC
35 行	#define LCD_VSYNC_DP (10 + 2 + 272)	tvp + tvb + tvd	10 + 2 + 272 = 284 HSYNC
36 行	#define LCD_VSYNC_FP (10 + 2 + 272 + 132)	tvp + tvb + tvd + tvf	10 + 2 + 272 + 132 = 416 HSYNC
37 行	#define LCD_VSYNC_CYCLE 416	tv	416 HSYNC

3.6 デバッグ使用時の注意事項

本サンプルプログラムを E1 エミュレータ等のデバッグを使用してデバッグする際の注意事項を下記に記述します。

① フリッカ

ブレークをかけて処理を中断し、その後処理を再開させると LCD にフリッカが発生することがあります。

本現象は LCD の仕様に依存するもので、プログラムに問題はありません。

プログラムの実行を継続していくと次第にフリッカはなくなります。

デバッグを使用せずに動作させる場合には、上記の現象は発生しません。

② SDRAM のデータ化け

ブレークをかけて処理を中断した場合、SDRAM 内のデータが化けてしまうことがあります。

これは、LCD への DMA 転送を行っている間、リフレッシュが発生すると転送のタイミングにズレが発生し、画面が乱れてしまうため、SDRAM へのリフレッシュを停止しているためです。

ブレークをかけない状態であれば、DMA 転送を行っていない期間にリフレッシュを行っていますので、通常動作させている状態では、上記の現象は発生しません。

3.7 プログラミングの注意事項

本サンプルプログラムを参考にプログラムを作成される場合の注意事項を下記に記述します。

① SDRAM へのアクセス

DMA 転送を行っているときに SDRAM へのアクセスを行うと DMA のタイミングがズレてしまい、画面が乱れることがあります。

DMA への転送を行っていない vtd 以外の期間でアクセスを行ってください。

② 割り込み

LCD の表示のタイミングは、MTU の割り込みを基準に行っております。

そのため、この割り込み処理に影響を与えるような長い割り込み禁止等がある場合、画像に乱れが発生します。

また、優先順位の高い割り込みや多重割り込みの処理を行っていない割り込みなどを使用した場合にも割り込み禁止と同じ状態となりますので、ご注意ください。

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。
- ・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技術サポートは一切受け付けておりません。
- ・本サンプルプログラムに関して、ルネサスエレクトロニクスへのお問い合わせはご遠慮ください。
- ・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、弊社およびルネサスエレクトロニクスでは一切責任を負いませんのでご了承下さい。
- ・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。万一ご不審な点、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。
- ・本文書の内容は、将来予告なしに変更されることがあります。

商標について

- ・RX 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。
- ・High-performance Embedded Workshop 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。
- ・Windows®の正式名称は Microsoft®Windows®Operating System です。
- ・Microsoft、Windows は、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。
- ・SuperH 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。

本文書では下記のように省略して記載している場合がございます。ご了承下さい。

- ・Windows®7 は Windows7 もしくは Win7
- ・Windows®Vista は Windows Vista もしくは WinVista
- ・Windows®XP は Windows XP もしくは WinXP
- ・High-performance Embedded Workshop は HEW

- ・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
静岡県浜松市東区積志町834
<http://www.apnet.co.jp>
E-MAIL : query@apnet.co.jp