

# LCD-KIT-D02 サンプルプログラム解説 (RX64M)

第2版 2018年 1月9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.....	5
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)を用いて静電容量式の LCD-KIT-D02 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-D02	L C Dパネル	各種画像の表示
	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
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(SCI2)	LCD-KIT との通信
	周期タイマ(CMT0,MTU2)	時間管理
	シリアル通信(SCI0)	標準出力

## 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 フォルダ内>	
ap_rx64m_0a_lcdkit.mtpj	… CS+用プロジェクトファイル
<¥ap_rx64m_0a_lcdkit¥DefaultBuild フォルダ内>	
ap_rx64m_0a_lcdkit.abs	… elf 形式オブジェクトファイル
ap_rx64m_0a_lcdkit.mot	… モトローラ S フォーマット形式ファイル
ap_rx64m_0a_lcdkit.map	… マップファイル
<¥ap_rx64m_0a_lcdkit¥src フォルダ内>	
AP_RX64M_0A.c	… メイン処理
exdma.c	… DMA 処理
i2c.c	… I2C 処理
image.c	… LCD 表示画像描画処理
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_riic_rx	I2C モジュールフォルダ
<¥ap_rx64m_0a_lcdkit¥data フォルダ内>	
Image0.bin	… LCD 表示用写真画像バイナリデータ 0
Image1.bin	… LCD 表示用写真画像バイナリデータ 1
Image2.bin	… LCD 表示用写真画像バイナリデータ 2
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

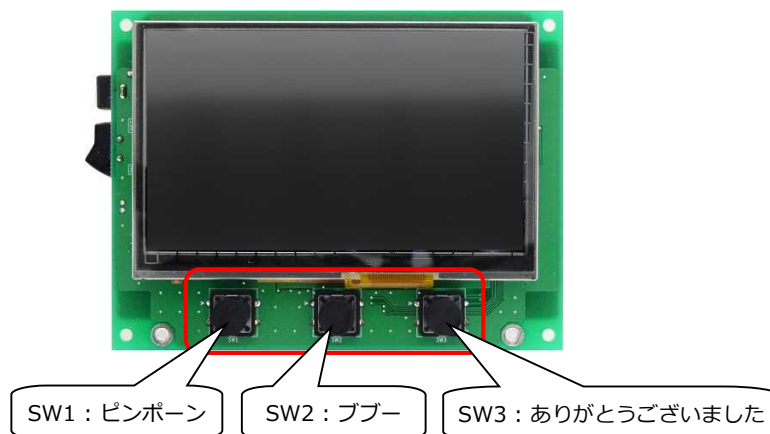
#### 3.1 動作説明

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 のスピーカーへ出力します
- ② LCD パネル       3 種類の画像を 5 秒毎に切り替えて表示します
- 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  
5 点検出(青、赤、黄、緑、白のポインタ)

##### ① SW1～3

SW1～3 を押すと、スピーカーから音声が出力されます。



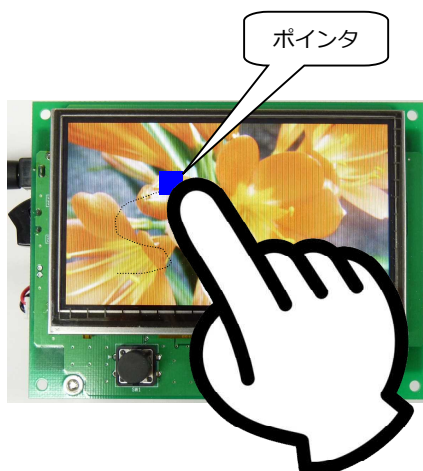
##### ② LCD

画像を 5 秒おきに切り替えて表示します。  
3 種類の写真が以下のように切り替わります。



### 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5点のマルチタッチ検出を行い、青・赤・黄・緑・白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 開発環境の使用方法(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 開発環境の使用方法(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¥DefaultBuild
モトローラファイル名	2-3	ap_rx64m_0a_lcdkit¥DefaultBuild¥ap_rx64m_0a_lcdkit.mot
アブソリュートファイル名	2-4	ap_rx64m_0a_lcdkit¥DefaultBuild¥ap_rx64m_0a_lcdkit.abs
マップファイル	2-5	ap_rx64m_0a_lcdkit¥DefaultBuild¥ap_rx64m_0a_lcdkit.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¥DefaultBuild¥ap_rx64m_0a_lcdkit.mot
ボード設定 (動作)	3-4	Fig 3-2 を参照

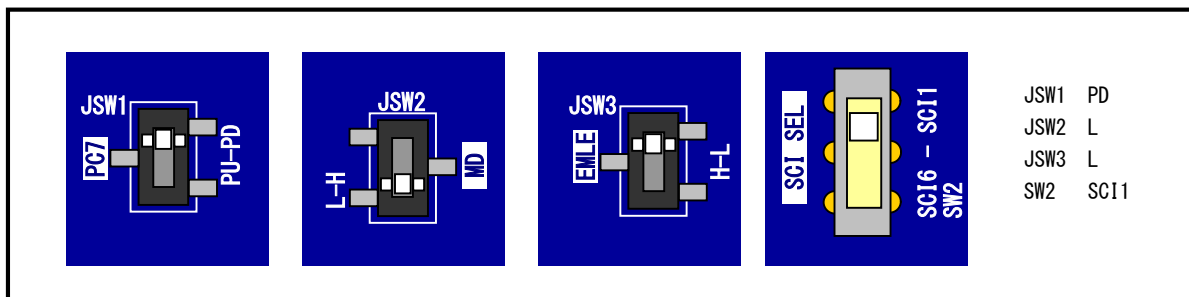


Fig 3-1 Flash 書き込み時のボード設定

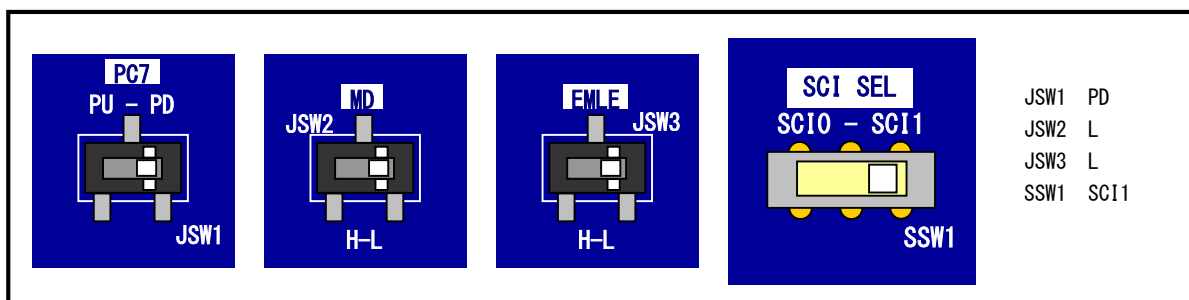


Fig 3-2 サンプルプログラム動作時のボード設定

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)

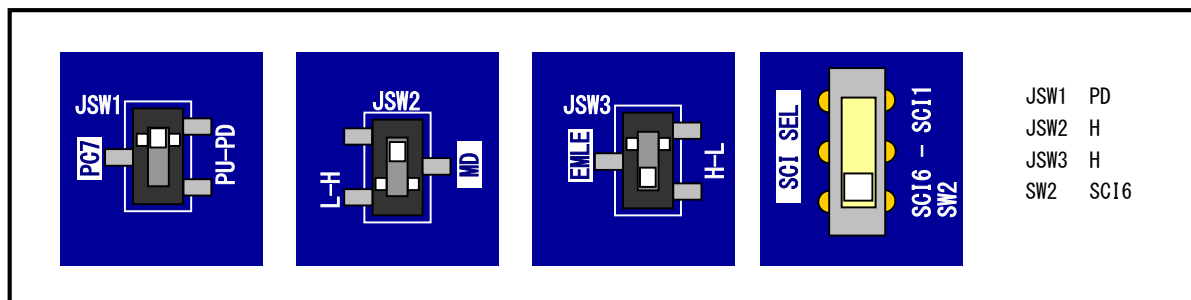
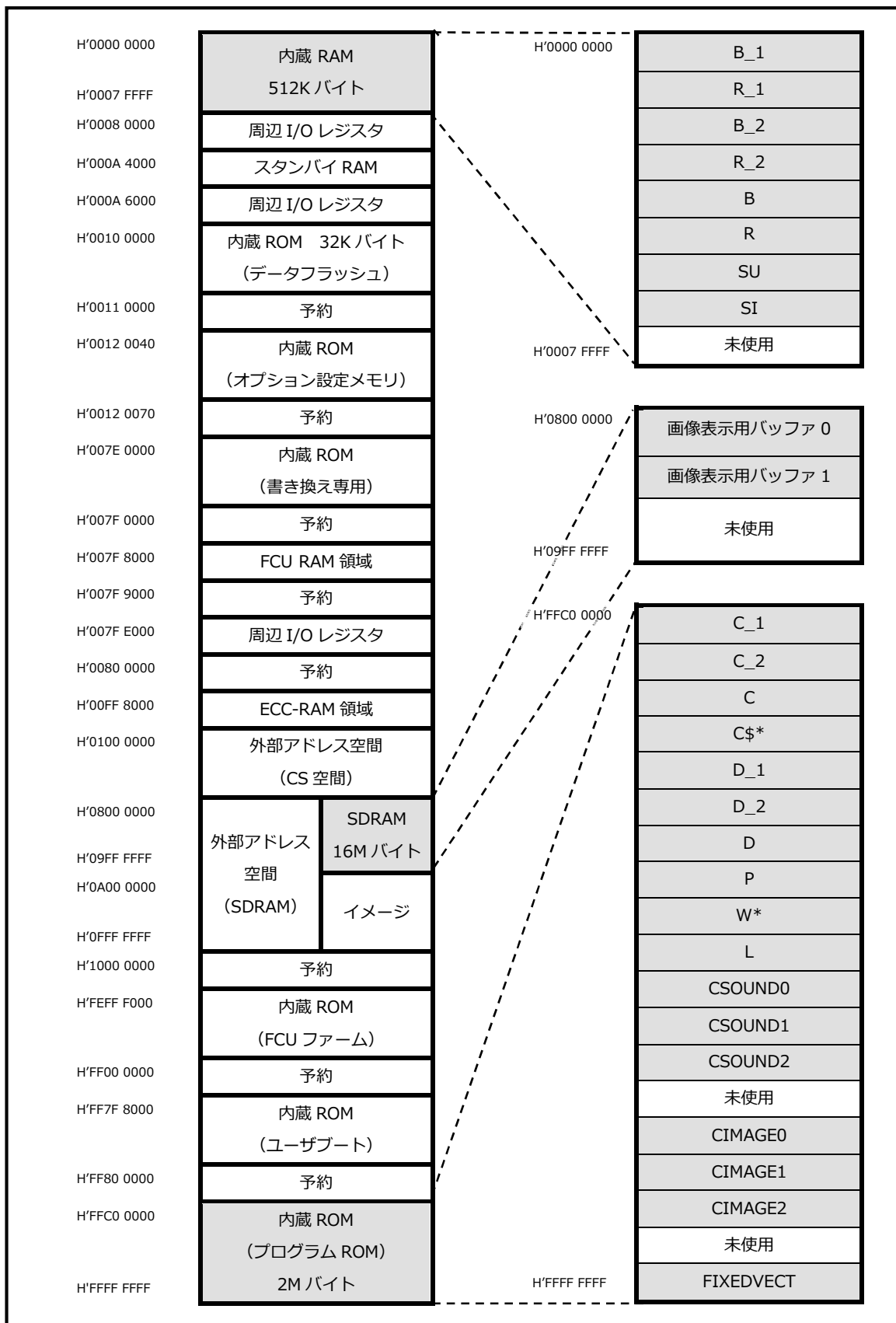


Fig 3-3 E1 エミュレータデバッグ時のボード設定

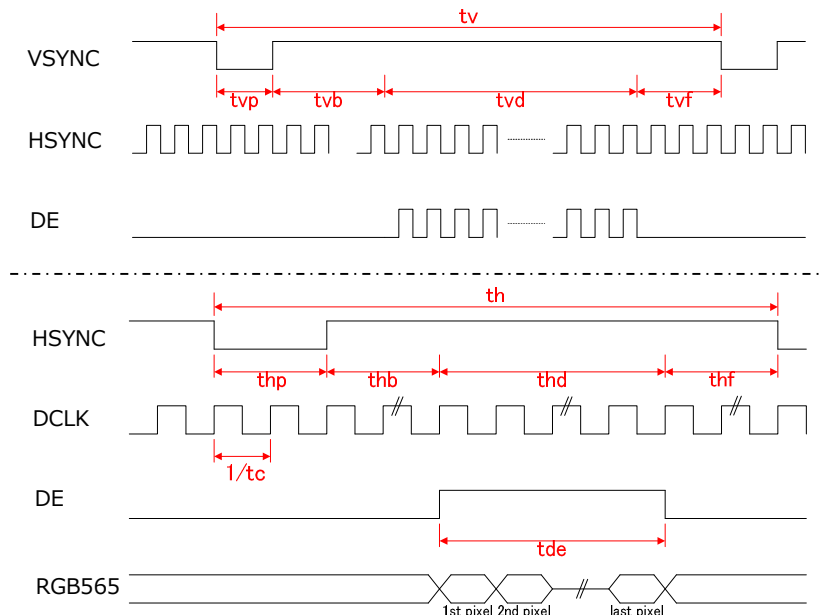


3.4 メモリマップ



### 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	MTIOC3A	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 <b>10</b>	1/tc	$P\Phi / 10 = 120 \text{ MHz} / 10 = 12 \text{ MHz}$
28 行	#define LCD_DE_START <b>43</b>	thp + thb	<b>43</b> clk
29 行	#define LCD_DE_LENGTH <b>480</b>	tde	<b>480</b> clk
30 行	#define LCD_HSYNC_CYCLE <b>575</b>	th	<b>575</b> clk
31 行	#define LCD_HSYNC_LENGTH <b>41</b>	thp	<b>41</b> clk
33 行	#define LCD_VSYNC_LENGTH <b>10</b>	tvp	<b>10</b> HSYNC
34 行	#define LCD_VSYNC_BP ( <b>10</b> + <b>2</b> )	tvp + tvb	<b>10</b> + <b>2</b> = <b>12</b> HSYNC
35 行	#define LCD_VSYNC_DP ( <b>10</b> + <b>2</b> + <b>272</b> )	tvp + tvb + tvd	<b>10</b> + <b>2</b> + <b>272</b> = <b>284</b> HSYNC
36 行	#define LCD_VSYNC_FP ( <b>10</b> + <b>2</b> + <b>272</b> + <b>132</b> )	tvp + tvb + tvd + tvf	<b>10</b> + <b>2</b> + <b>272</b> + <b>132</b> = <b>416</b> HSYNC
37 行	#define LCD_VSYNC_CYCLE <b>416</b>	tv	<b>416</b> HSYNC

### 3.6 デバッガ使用時の注意事項

本サンプルプログラムを E1 エミュレータ等のデバッガ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。

デバッガ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。
- ・本サンプルプログラムで使用されているミドルウェア及びドライバの著作権はルネサスエレクトロニクス株式会社が保有します。
- ・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。
- ・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

- ・RX64M 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。
- ・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。
- ・CS+ 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、Windows NT 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。  
Windows®10、Windows®8、Windows®7、Windows®Vista は、米国 Microsoft Corporation.の商品名称です。  
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。ご了承ください。  
Windows®10 は Windows 10 もしくは Win10  
Windows®8 は Windows 8 もしくは Win8  
Windows®7 は Windows 7 もしくは Win7  
Windows®Vista は Windows Vista もしくは WinVista
- ・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](mailto:query@apnet.co.jp)