AP-RX71M-0A
LCD-KIT サンプルプログラム解説

第 5.1 版　2026 年 04 月 08 日

目 次

1. 概要 1

1.1 概要 1
1.2 本サンプルプログラムについて 1
1.3 開発環境について 2

2. サンプルプログラムの構成 3

2.1 フォルダ構成 3
2.2 ファイル構成 4

3. LCD-KIT サンプルプログラム 6

3.1 動作説明 6
3.2 LCD-KIT の選択・設定 8
3.2 メモリマップ 9
3.3 サンプルプログラムのダウンロード 10
3.4 LCD ダイレクトドライブ設定 11
3.5 デバッガ使用時の注意事項 12
3.6 プログラミングの注意事項 12

4. 開発環境使用時の各設定値 13

©2026 Alpha Project Co., Ltd.

1. 概要

1.1 概要

本アプリケーションノートでは、AP-RX71M-0A(RX71M)を用いて静電容量式 LCD キット（LCD-KIT-D02、LCD-KIT-D03）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
AP-RX71M-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）
	I2C 通信(RIIC0)	LCD-KIT との通信
	周期タイマ(CMT0,MTU2)	時間管理
LCD-KIT ・LCD-KIT-D02 ・LCD-KIT-D03	ＬＣＤパネル	各種画像の表示
	タッチパネル(静電容量式)	タッチ検出
	バックライト	バックライトの点灯
	スイッチ	各モード切り替え
	スピーカ（LCD-KIT-D02）	音声出力
	ブザー （LCD-KIT-D03）	ブザー音出力

1.2 本サンプルプログラムについて

本サンプルプログラムは、ルネサス エレクトロニクス株式会社提供のミドルウェアおよびドライバを AP-RX71M-0A に移植しています。
各ミドルウェアおよびドライバの詳細については、以下の資料を参照してください。
入手につきましては、ルネサス社ウェブサイトの下記のページにて、検索を行ってください。
FIT モジュールにつきましては、Smart Configurator から入手することも可能です。

ルネサス エレクトロニクス社 RX71M サンプルコード
https://www.renesas.com/jp/ja/products/microcontrollers-microprocessors/rx-32-bit-performance-efficiency-mcus/rx71m-high-performance-real-time-engine-32-bit-microcontrollers-industrial-equipment#documents

©2026 Alpha Project Co., Ltd.

● BSP
・資料名 RX ファミリ ボードサポートパッケージモジュール Firmware Integration Technology 機能名称：BSP　　<R01AN1685　Rev 5.20>
● CMT
・資料名 RX ファミリ CMT モジュール Firmware Integration Technology 機能名称：タイマ　　<R01AN1856　Rev 4.00>
● GPIO
・資料名 RX ファミリ GPIO モジュール Firmware Integration Technology 機能名称：I/O 設定　　<R01AN1721　Rev 3.00>
● MPC
・資料名 RX ファミリ MPC モジュール Firmware Integration Technology 機能名称：端子設定　　<R01AN1724　Rev3.00>
● I2C
・資料名 RX ファミリ I2C バスインタフェース(RIIC)モジュール Firmware Integration Technology 機能名称：I2C バス　　<R01AN1692　Rev 2.41>

（※）資料をダウンロードする際にはルネサス エレクトロニクス株式会社の My Renesas への登録が必要となります。

1.3　開発環境について

本サンプルプログラムは、統合開発環境「CS+」を用いて開発されています。
本サンプルプログラムに対応する開発環境、コンパイラのバージョンは次の通りです。

ソフトウェア	バージョン	備考
CS+	v8.13.00	–
RX 用コンパイラ　CC-RX	V2.08.01	–
Smart Configurator	v2.2.0	–

©2026 Alpha Project Co., Ltd.

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。

¥ ap_rx71m_0a_lcdkit	LCD-KIT サンプルプログラム(AP-RX71M-0A)フォルダ
├ ¥ src	ソースフォルダ
│ ├ ¥ smc_gen	Smart Configurator 生成フォルダ
│ │ ├ ¥ general	general フォルダ
│ │ ├ ¥ r_bsp	BSP モジュールフォルダ
│ │ ├ ¥ r_cmt_rx	CMT モジュールフォルダ
│ │ ├ ¥ r_gpio_rx	GPIO モジュールフォルダ
│ │ ├ ¥ r_mpc_rx	MPC モジュールフォルダ
│ │ ├ ¥ r_riic_rx	I2C モジュールフォルダ
│ │ ├ ¥ r_pincfg	端子設定ファイルフォルダ
│ │ └ ¥ r_config	各モジュールの設定ファイルフォルダ
│ └ ¥ lcdkit	LCD-KIT 用ファイルフォルダ
├ ¥ DefaultBuild	ワークフォルダ
├ ¥ .settings	Smart Configurator 設定フォルダ
└ ¥ Data	音声および画像のバイナリデータフォルダ

©2026 Alpha Project Co., Ltd.

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。
本章では、ミドルウェア・ドライバ等の既存のファイルに関しては説明を省略しています。

<¥Sample¥ap_rx71m_0a_lcdkit フォルダ内>

ap_rx71m_0a_lcdkit.mtpj	･･･	CS+用プロジェクトファイル
ap_rx71m_0a_lcdkit.rcpe	･･･	e2studio 用プロジェクトファイル
ap_rx71m_0a_lcdkit.scfg	･･･	Smart Configurator 用ファイル （CS+上から Smart Configurator を起動できます。）
AP-RX71M-0A_lcdkit_sample_V2.0.bdf	･･･	Board Description File （本プログラムのクロック周波数、端子設定を Smart Configurator にインポートできます。）

<¥Sample¥ap_rx71m_0a_lcdkit ¥DefaultBuild フォルダ内>

ap_rx71m_0a_lcdkit.abs	･･･	elf 形式オブジェクトファイル
ap_rx71m_0a_lcdkit.mot	･･･	モトローラ S フォーマット形式ファイル
ap_rx71m_0a_lcdkit.map	･･･	マップファイル

<¥Sample¥ap_rx71m_0a_lcdkit ¥src フォルダ内>

smc_gen	･･･	Smart Configurator により生成されたモジュールフォルダ
lcdkit	･･･	LCD-KIT 用ファイルフォルダ
ap_rx71m_0a.c	･･･	メイン処理ソースファイル
cmt_dev.c	･･･	タイマドライバソースファイル
image_data.c	･･･	画像表示データ設定ソースファイル
lcd_ctrl_exdma.c	･･･	LCD ダイレクトドライブ EXDMA ドライバソースファイル
lcd_ctrl_mtu.c	･･･	LCD ダイレクトドライブ MTU ドライバソースファイル
lcd_disp.c	･･･	LCD 表示処理ソースファイル
i2c_dev.c	･･･	I2C ドライバソースファイル
sdram_dev.c	･･･	SDRAM ドライバソースファイル
cmt_dev.h	･･･	タイマドライバヘッダファイル
lcd_ctrl.h	･･･	LCD ダイレクトドライブドライバヘッダファイル
lcd_disp.h	･･･	LCD 表示処理ヘッダファイル
i2c_dev.h	･･･	I2C ドライバヘッダファイル

<¥Sample¥ap_rx71m_0a_lcdkit¥src¥lcdkit フォルダ内>

lcdkit_app.c	･･･	LCD-KIT 制御処理ソースファイル
lcdkit_d02.c	･･･	LCD-KIT-D02 ドライバソースファイル
lcdkit_d03.c	･･･	LCD-KIT-D03 ドライバソースファイル
lcdkit.h	･･･	LCD-KIT ヘッダファイル
lcdkit_app.h	･･･	LCD-KIT 制御処理ヘッダファイル
lcdkit_d02.h	･･･	LCD-KIT-D02 ドライバヘッダファイル
lcdkit_d03.h	･･･	LCD-KIT-D03 ドライバヘッダファイル

©2026 Alpha Project Co., Ltd.

<¥Sample¥ap_rx71m_0a_lcdkit ¥Data フォルダ内>

Image_scale.bin	・・・	LCD 表示用バックライト調整画像バイナリデータ 1
Image_start1.bin	・・・	LCD 表示用音声再生画像バイナリデータ 1
Image_start2.bin	・・・	LCD 表示用音声再生画像バイナリデータ 2
Image_start3.bin	・・・	LCD 表示用音声再生画像バイナリデータ 3
Image0.bin	・・・	LCD 表示用写真画像バイナリデータ 0
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

©2026 Alpha Project Co., Ltd.

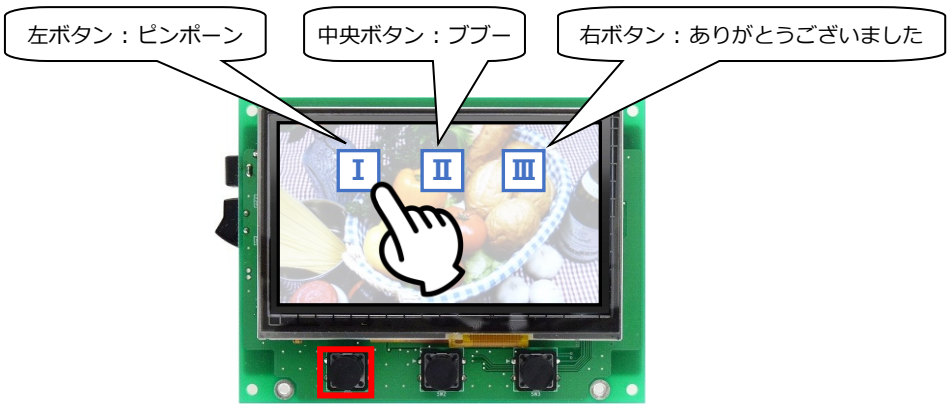
3.　LCD-KIT サンプルプログラム

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。

- LCD 画面上に画像を表示します。（480*272px 画像）
- LCD-KIT のハードボタンを押すことで以下の 3 つのモードに切り替えることができます。
  （サンプルプログラム開始時は SW2 を押したときに切り替わる「タッチ確認モード」が動作します。）
  ・SW1：音声確認モード/ブザー確認モード（LCD キットの種類で動作が変わります）
  ・SW2：タッチ確認モード
  ・SW3：輝度確認モード

- 音声確認モード（LCD-KIT-D02）
  LCD-KIT の SW1 を押すことで音声確認モードが動作します。
  本モードでは、画面上部に三つのボタンを表示します。ユーザは各ボタンをタッチすることでボタンに対応した音声を再生することが可能です。



SW1 押す

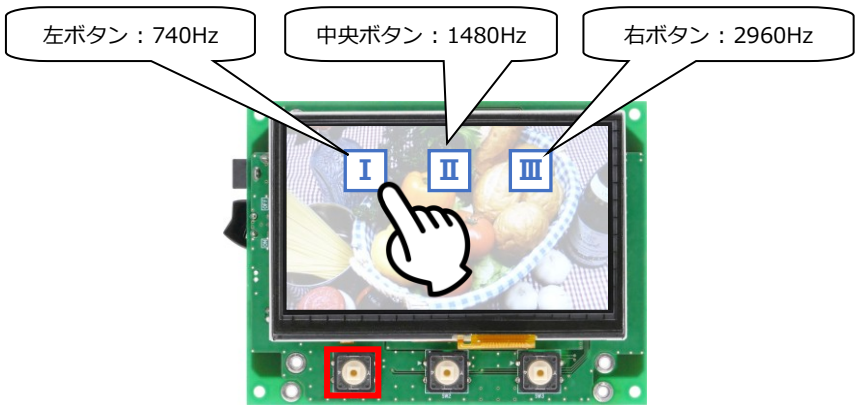
ボタン	動作
左ボタン（Ⅰ）	「ピンポーン」という音声再生
中央ボタン（Ⅱ）	「ブブー」という音声再生
右ボタン（Ⅲ）	「ありがとうございました」という音声再生

©2026 Alpha Project Co., Ltd.

- ブザー確認モード（LCD-KIT-D03）

  LCD-KIT の SW1 を押すことでブザー確認モードが動作します。

  本モードでは、画面上部に三つのボタンを表示します。ユーザは各ボタンをタッチすることでボタンに対応した周波数のブザーが鳴ります。



SW1 押す

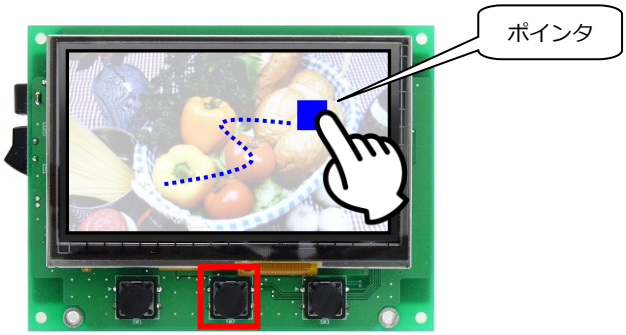
ボタン	動作
左ボタン（Ⅰ）	約 740Hz
中央ボタン（Ⅱ）	約 1480Hz
右ボタン（Ⅲ）	約 2960Hz

- タッチ確認モード

  サンプルプログラムの起動直後、あるいは、LCD-KIT の SW2 を押すことでタッチ確認モードが動作します。

  本モード中に LCD-KIT の画面をタッチすると四角形のカーソルを表示します。

  最大 5 点のマルチタッチ検出を行い、青・赤・緑・黄・白の順でポインタが表示されます。



SW2 押す

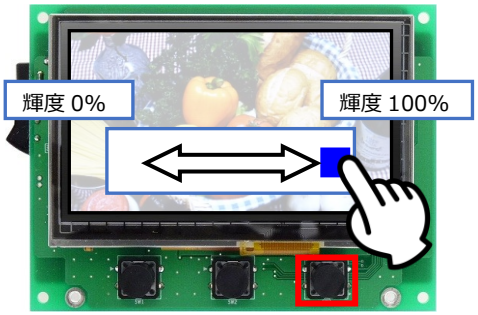
©2026 Alpha Project Co., Ltd.

- 輝度確認モード

  LCD-KIT の SW3 を押すことで輝度確認モードが動作します。

  本モードでは、LCD-KIT の下部に長方形の輝度調節スペースを表示します。

  ユーザは輝度調節スペースをタッチすることで画面輝度を変更できます。輝度調節スペースの右端が画面輝度 100%、左端が画面輝度 0%です。



SW3 押す

輝度を変更後に LCD-KIT の SW1, 2, 3 を押すと、輝度は 100%に戻ります。

3.2　LCD-KIT の選択・設定

本サンプルプログラムは下記の動作を行います。

本サンプルプログラムは、使用する LCD-KIT によって、下記ソースファイルの変更が必要な場合があります。

変更後は、e2 studio 上で [プロジェクトをクリーンにする]を行ってから [プロジェクトのビルド]を選択して、

必ずプロジェクトを再ビルドしてください。

デフォルトでは LCD-KIT-D03 を使用する設定になっています。

¥Sample¥ap_rx71m_0a_lcdkit¥src¥lcdkit¥lcdkit.h（23～28 行目）

23 :	#define LCDKIT_NONE　(0)	
24 :	#define LCDKIT_D02　(LCDKIT_NONE + 1)	←LCD-KIT-D02 と接続時の設定値
25 :	#define LCDKIT_D03　(LCDKIT_D02 + 1)	←LCD-KIT-D03 と接続時の設定値
26 :	#define LCDKIT_COUNT　(LCDKIT_D03 + 1)	
27 :		
28 :	#define LCDKIT　(LCDKIT_D03)　　/* <- Please specify the LCD-	←赤字の値を適切に設定してください

©2026 Alpha Project Co., Ltd.

3.3　メモリマップ

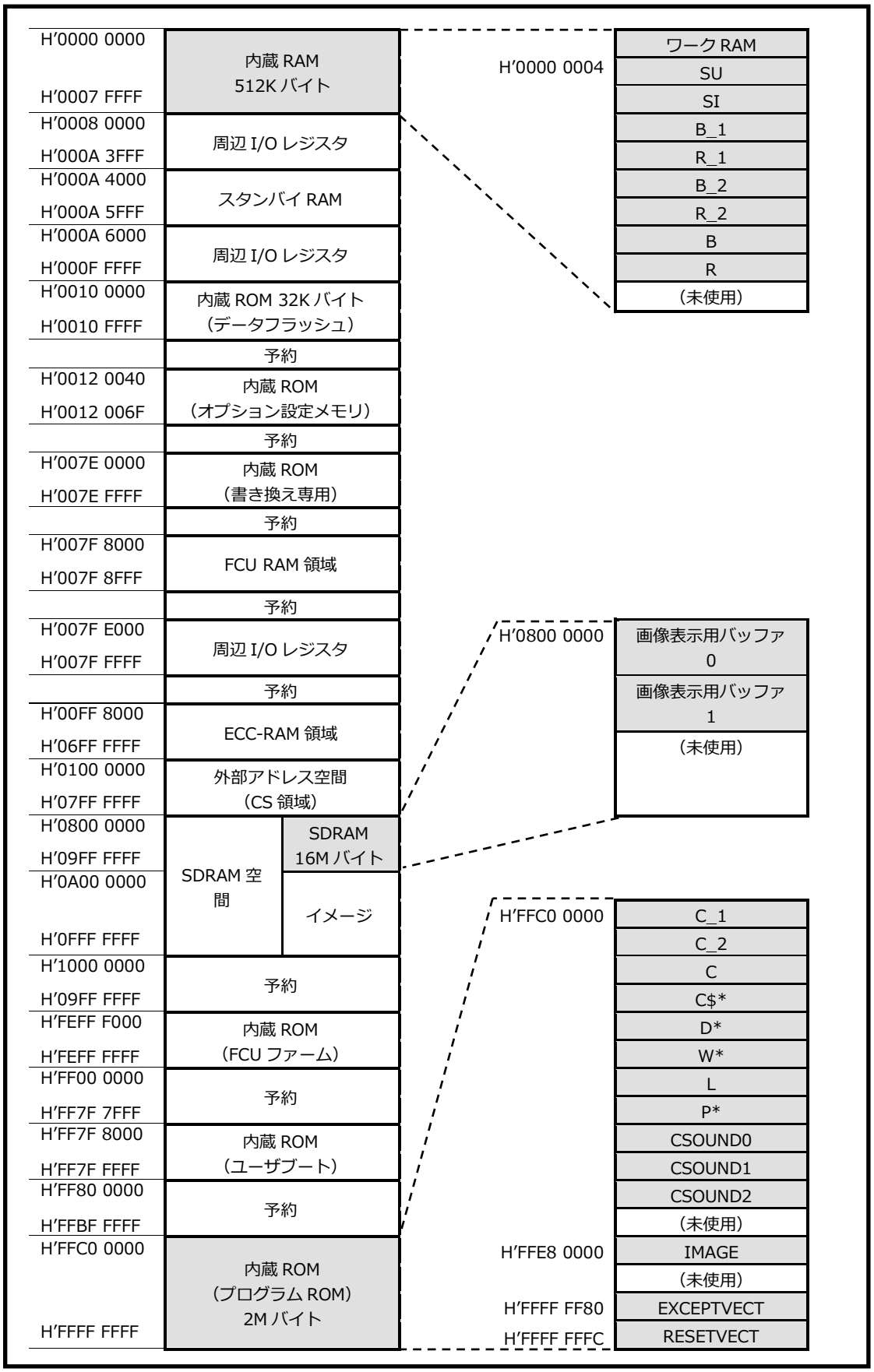


Fig 3.2-1　LCD-KIT サンプルプログラム(AP-RX71M-0A) メモリマップ

©2026 Alpha Project Co., Ltd.

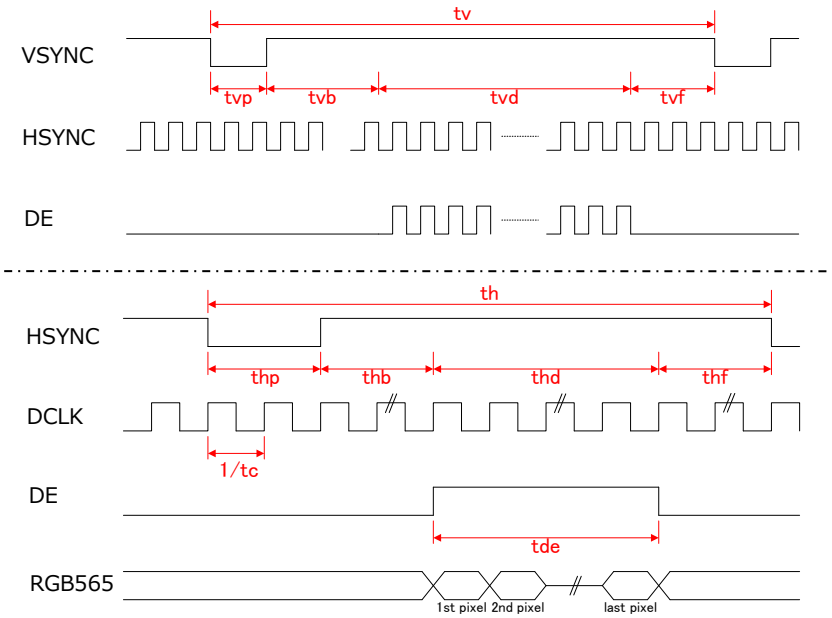
3.4　サンプルプログラムのダウンロード

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。

©2026 Alpha Project Co., Ltd.

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およびタイミングは、以下のように設定されています。



・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 <lcd_ctrl.h>内）

29 行	#define LCD_CLKNUM **10**	1/tc	PΦ / **10** = 120 MHz / 10 = **12** MHz
30 行	#define LCD_DE_START **43**	thp + thb	**43** clk
31 行	#define LCD_DE_LENGTH **480**	tde	**480** clk
33 行	#define LCD_HSYNC_CYCLE **575**	th	**575** clk
34 行	#define LCD_HSYNC_LENGTH **38**	thp	**38** clk
36 行	#define LCD_VSYNC_LENGTH **10**	tvp	**10** HSYNC
37 行	#define LCD_VSYNC_BP (**10** + **2**)	tvp + tvb	**10** + **2** = **12 HSYNC**
38 行	#define LCD_VSYNC_DP (**10** + **2**+ **272**)	tvp + tvb + tvd	**10** + **2** + **272** = **284 HSYNC**
39 行	#define LCD_VSYNC_FP (**10** + **2** + **272** + **37**)	tvp + tvb + tvd + tvf	**10** + **2** + **272** + **37** = **321 HSYNC**
40 行	#define LCD_VSYNC_CYCLE **321**	tv	**321** HSYNC

©2026 Alpha Project Co., Ltd.

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には、下記に注意が必要です。

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 のデータ化けが生じることがあります。
これは、LCD への DMA 転送を行っている期間は、リフレッシュを行うと転送のタイミングにズレが生じて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ことにより発生します。
ブレークをかけず通常動作している状態であれば、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、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する場合は、下記に注意が必要です。

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 のタイミングがずれてしまい、
画面が乱れることがあります。
DMA への転送を行っている vtd の期間を避けて、DMA 転送をしていない期間に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。

※　本サンプルではルネサス エレクトロニクス株式会社提供のドライバ等を使用していますが、上記の対策として、
BSP モジュールおよび CMT モジュールに多重割り込み許可の処理を追加しています。
Smart Configurator を使用するなどしてプログラムを作成する場合はご注意ください。

	BSP モジュール（r_bsp）	CMT モジュール（r_cmt_rx）
該当ファイル	mcu_interrupts.c （¥src¥smc_gen¥r_bsp¥mcu¥rx71m）	r_cmt_rx.c （¥src¥smc_gen¥r_cmt_rx¥src）
追加処理	**set_psw(get_psw() | 0x00010000);**	
追加箇所	・void group_bl0_handler_isr (void)内 ・void group_bl1_handler_isr (void)内 ・void group_al0_handler_isr (void)内 ・void group_al1_handler_isr (void)内	・static void cmt0_isr (void)内 ・static void cmt0_isr (void)内 ・static void cmt0_isr (void)内 ・static void cmt0_isr (void)内

©2026 Alpha Project Co., Ltd.

4.　開発環境使用時の各設定値

開発環境を使用する際の、AP-RX71M-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	Sample¥ap_rx71m_0a_lcdkit ¥DefaultBuild
モトローラファイル名	2-3	Sample¥ap_rx71m_0a_lcdkit ¥DefaultBuild¥ap_rx71m_0a_lcdkit.mot
アブソリュートファイル名	2-4	Sample¥ap_rx71m_0a_lcdkit ¥DefaultBuild ¥ap_rx71m_0a_lcdkit.abs
マップファイル	2-5	Sample¥ap_rx71m_0a_lcdkit ¥DefaultBuild ¥ap_rx71m_0a_lcdkit.map

Renesas Flash Programmer を使用した Flash 書き込み方法（シリアルポート(SCI)を使用する方法）		
項目名	項目番号	設定値
ボード設定（Flash 書き込み）	3-1	ボード：Fig 4-1 を参照　　ケーブル接続：CN11
Flash に書き込むファイル	3-3	Sample¥ap_rx71m_0a_lcdkit ¥DefaultBuild ¥ap_rx71m_0a_lcdkit.mot
ボード設定（動作）	3-4	Fig 4-3 を参照

Renesas Flash Programmer を使用した Flash 書き込み方法（USB ブートモードを使用する方法）		
項目名	項目番号	設定値
ボード設定（Flash 書き込み）	3-5	ボード：Fig 4-2 を参照　　ケーブル接続：CN6(USB microB)
ツール選択	3-6	[USB Direct]
Flash に書き込むファイル	3-7	Sample¥ap_rx71m_0a_lcdkit ¥DefaultBuild ¥ap_rx71m_0a_lcdkit.mot
ボード設定（動作）	3-8	Fig 4-3 を参照

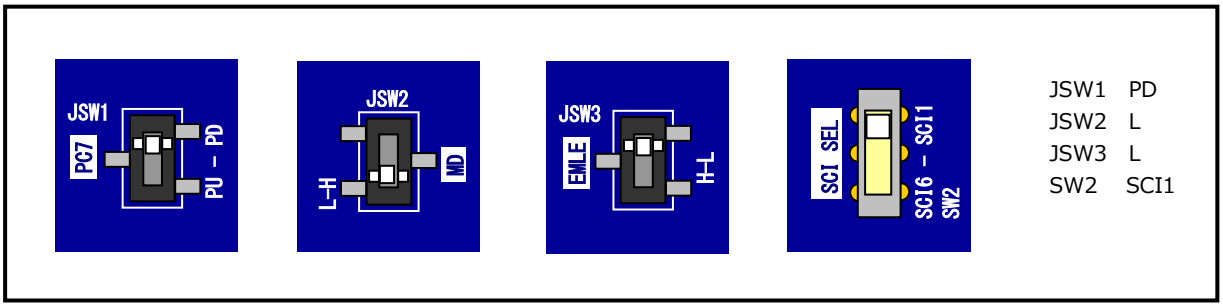


Fig 4-1　Flash 書き込み(シリアルポート使用)時のボード設定

©2026 Alpha Project Co., Ltd.

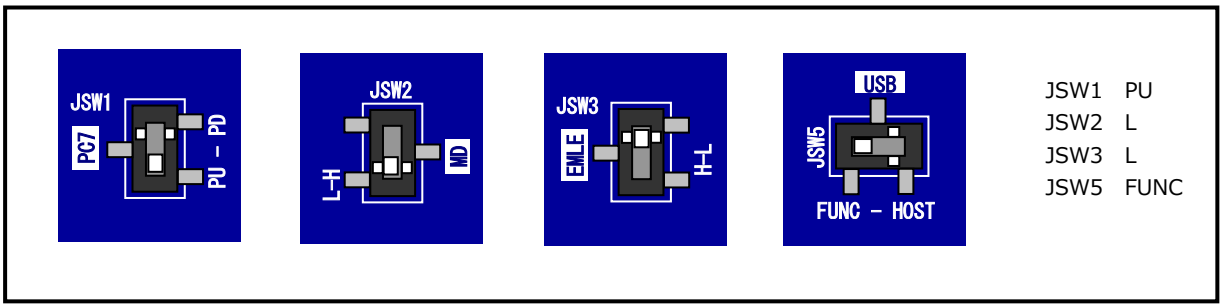


Fig 4-2　Flash 書き込み(USB ブートモード)時のボード設定

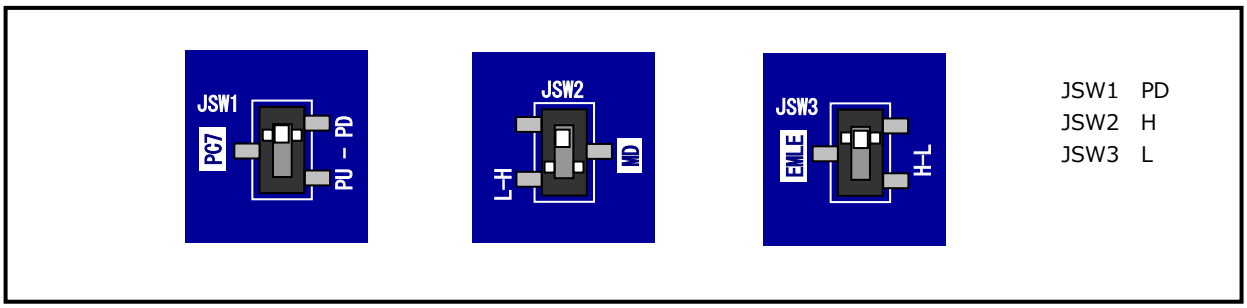


Fig 4-3　サンプルプログラム動作時のボード設定

E1 エミュレータ/E2 エミュレータ Lite 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 4-4 を参照
JTAG クロック	4-10	E1 エミュレータを使用する場合：16.5(MHz) E2 エミュレータ Lite を使用する場合：6.00(MHz)
EXTAL クロック	4-11	24(MHz)

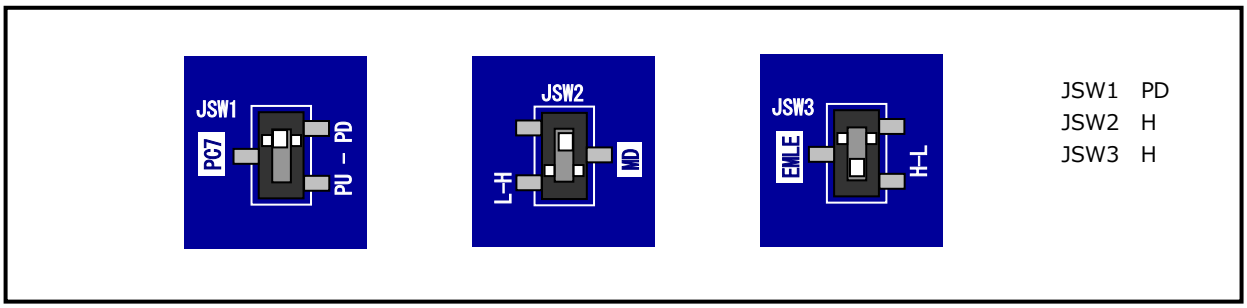


Fig 4-4　E1 エミュレータ/E2 エミュレータ Lite デバッグ時のボード設定

©2026 Alpha Project Co., Ltd.

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およ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ください。
・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が、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡ください。
・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。
・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。
・E1 エミュレータはルネサス 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。
・E2 エミュレータ Lite 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。
・Renesas Flash Programmer 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。

・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。

ALPHA PROJECT

株式会社アルファプロジェクト

〒431-3114
静岡県浜松市中央区積志町８３４
https://www.apnet.co.jp
E-Mail: query@apnet.co.jp

©2026 Alpha Project Co., Ltd.