RX231 グループ

RX231 HMI Solution Kit Base Demo Software (機能限定版) RTK5RX2310P000F0ZR

R11AN0015JJ0120
Rev.1.20
2020.8.20

要旨

本資料は、RX231 HMI Solution Kit Base Demo Software (機能限定版) RTK5RX2310P000F0ZR のアプリケーションノートです。

動作確認デバイス

RX231

目次

1. はじめに 3
1.1. 関連ドキュメント 3
1.2. 用語と略語 3
2. 動作確認条件 4
3. ハードウェア 5
3.1. ブロック図 5
3.2. 端子配置表 6
4. 基本デモソフトウェア 8
4.1. 動作概要 8
4.2. ソフトウェア構成 8
4.3. FIT モジュール 9
4.4. 割り込み 10
4.5. 動作フロー 11
4.6. 状態遷移 14
4.7. ファイル構成 16
4.8. センサ制御 18
5. 定数一覧 20
6. 型・構造体一覧 22
7. グローバル変数一覧 25
8. API 関数一覧 26
9. API 関数詳細 28
9.1. MCU 関数 28
9.2. LCD 制御関数 36
9.3. SAIC101 制御関数 37
9.4. kxss5 (加速度センサ)制御関数 38

9.5. シリアルフラッシュ制御関数 39
9.6. 音声再生・録音制御関数 40
9.7. タッチキー制御関数 41
9.8. 歩数計制御関数 42
10. 注意事項 43
ホームページとサポート窓口 44

1. はじめに

本アプリケーションノートは、ルネサスエレクトロニクス製MCU「RX231」を使用したRX231 Human Machine Interface Solution Kit R0K5RX231D000BRのハードウェアの機能概要および基本デモソフトウェア(RX231 HMI Solution Kit Base Demo Software (機能限定版) RTK5RX2310P000F0ZR)の使用方法を説明します。
本書内の赤枠部分は機能限定箇所です。

1.1. 関連ドキュメント

[1] RX231 ユーザーズマニュアル ハードウェア編(R01UH0496JJ)

[2] RX231 Human Machine Interface Solution Kit R0K5RX231D000BR 取扱説明書(R01AN2586JJ0100)

使用している Firmware Integration Technology(FIT)：

[3] IRQ (External Pin Interrupt Request) Module Using Firmware Integration Technology

[4] CMT Module Using Firmware Integration Technology

[5] 簡易 I²C モジュール Firmware Integration Technology

使用しているサンプルプログラム：

[6] 静電容量式タッチセンサシステム

上記のドキュメントは、ルネサス エレクトロニクスホームページより入手できます。

・ ルネサス エレクトロニクスホームページ

【http://japan.renesas.com/】

・ 静電容量式タッチセンサシステム

【http://japan.renesas.com/applications/key_technology/human_interface/touch_sensor_top/index.jsp】

1.2. 用語と略語

本書で使用される用語と略語は以下の通りです。

ADC	:	Analog to Digital Converter
API	:	Application Programming Interface
CMT	:	Compare Match Timer
DAC	:	Digital to Analog Converter
FIT	:	Firmware Integration Technology
H/W	:	Hardware
HMI	:	Human Machine Interface
IRQ	:	Interrupt Request
LCD	:	Liquid Crystal Display
MCU	:	Micro Control Unit
RTC	:	Real-Time Clock
RTOS	:	Real Time OS
S/W	:	Software
USB	:	Universal Serial Bus

2. 動作確認条件

Table 2-1 に、基本デモソフトウェアの動作確認条件を示します。

Table 2-1 動作確認条件

項目	機能説明
ハードウェア	ルネサスエレクトロニクス製 RX231 Human Machine Interface Solution Kit R0K5RX231D000BR
統合開発環境	ルネサスエレクトロニクス製 e² studio 2020-04
ツールチェイン	ルネサスエレクトロニクス製 Renesas RXC Toolchain V3.02.00
エミュレータ	ルネサスエレクトロニクス製 E1 emulator
動作周波数	メイン・システム・クロック ： 32MHz(水晶発振子 16MHz) サブシステム・クロック ： 32.768kHz(水晶発振子)
動作電圧	3.2V

注1. 元のプロジェクトで指定するツールチェーン(C コンパイラ）と同一のバージョンがインポートする先にない場合は、ツールチェーンが選択されない状態になり、エラーが発生します。プロジェクトの設定画面でツールチェーンの選択状態を確認してください。

選択方法は、FAQ 3000404 を参照してください。

FAQ 3000404 :インポートしたプロジェクトをビルドすると「PATH でプログラム "make" が見つかりません」エラーになる(e² studio)

3. ハードウェア

本章では、ハードウェアについて説明します。

ハードウェアの詳細については、関連ドキュメント[2] RX231 Human Machine Interface Solution Kit R0K5RX231D000BR 取扱説明書を参照下さい。

3.1. ブロック図

Figure 3-1 に、R0K5RX231D000BR のハードウェアブロック図を示します。

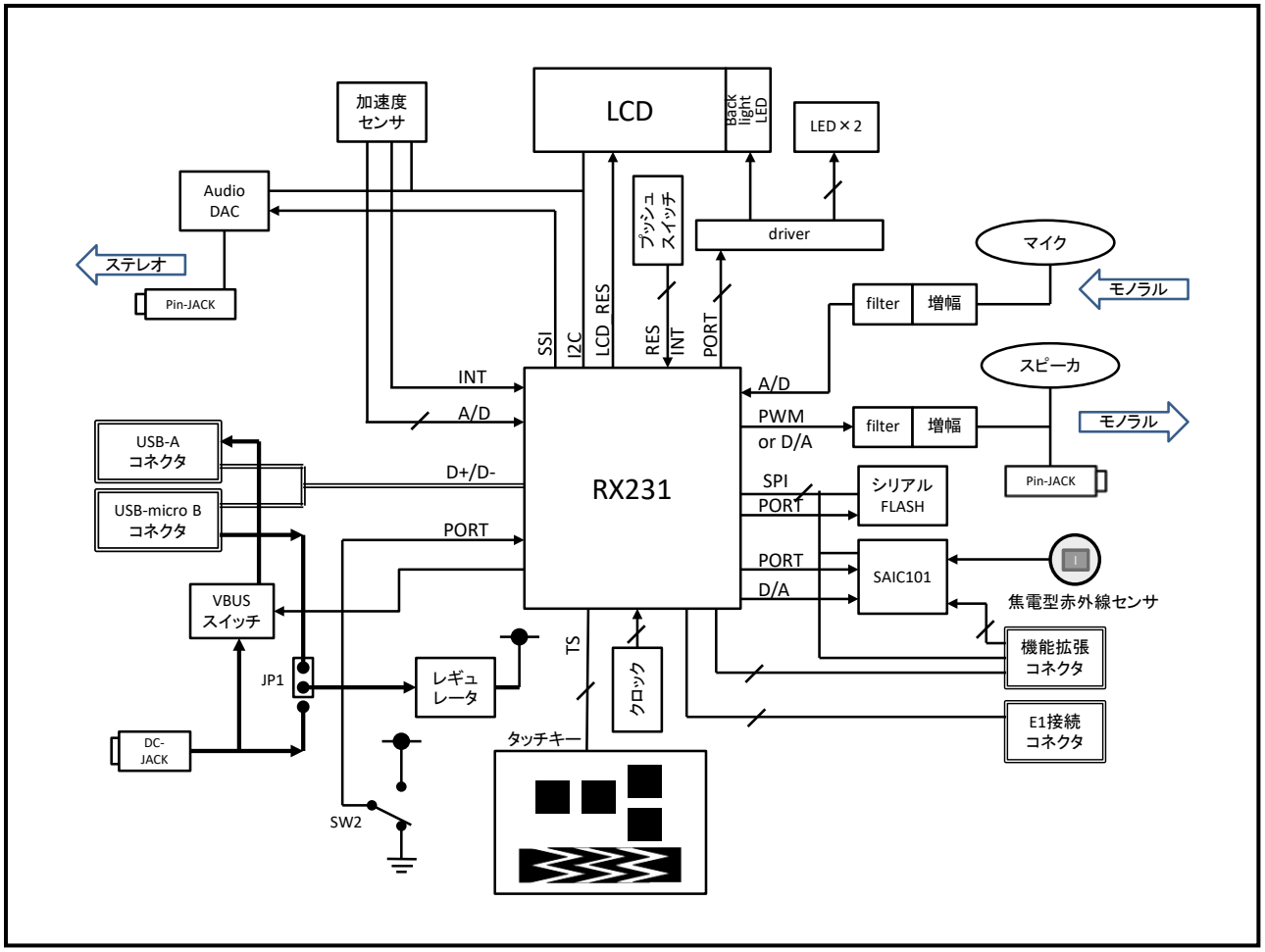


Figure 3-1 R0K5RX231D000BR ハードウェアブロック図

3.2. 端子配置表

Table 3-1 端子配置表 (1/2)

Table 3-2 端子配置表 (2/2)

4. 基本デモソフトウェア

ルネサスが策定している Firmware Integration Technology（以降 FIT と記載）モジュールを流用して作成しています。e² studio 形式でのプロジェクトになり、RTOS は使用していません。

4.1. 動作概要

スタート・アップ後、PowerON_Reset_PC() 関数がコールされ、その後に main() 関数がコールされます。PowerON_Reset_PC()関数で MCU のクロック設定を行います。main() 関数で MCU 周辺機能、ポート設定および周辺モジュールを初期設定します。

初期設定をした後、メインループが動作します。

4.2. ソフトウェア構成

Figure 4-1 に、基本デモソフトウェアのソフトウェア構成図を示します。

RX231 の各機能の初期設定、開始 ・停止および ユーティリティ機能については、Device Firmware (Firmware Integration Technology)が提供している API 関数を使用しています。Application 層は、Device Firmware が提供している API 関数及び「タッチキー」の Sample Program を使用しています。

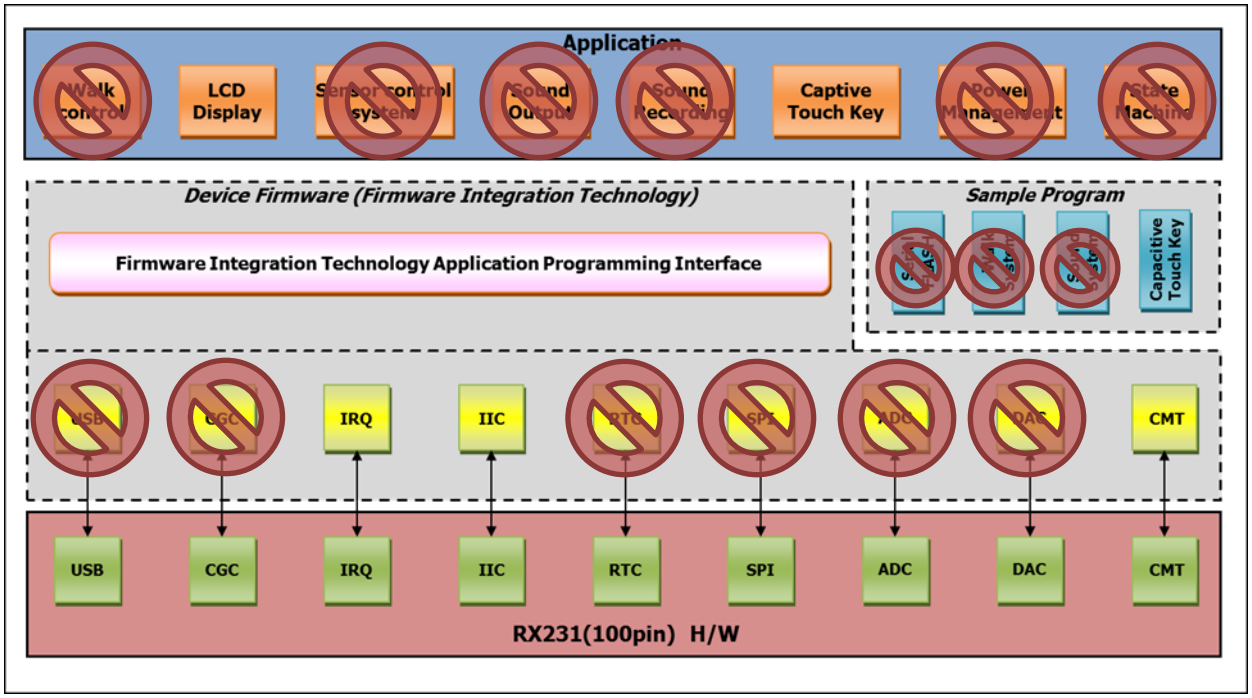


Figure 4-1 ソフトウェア構成図

4.3. FIT モジュール

Table 4-1 に基本デモソフトウェアで使用した FIT モジュールを記載します。各 FIT モジュールの詳細については、1.1 関連ドキュメントを参照下さい。

Table 4-1 FIT モジュール一覧

FIT モジュール	関数名	用途
IRQ	R_IRQ_Open()	外部割込み設定・起動 割り込みハンドラ登録 ・ハードウェアキー
CMT	R_CMT_CreatePeriodic()	CMT モジュール初期化 割り込みハンドラ登録 cmt0、cmt1 割り込み設定・起動
IIC	R_SCI_IIC_Open()	簡易 I²C モジュール初期化 割り込みハンドラ登録
	R_SCI_IIC_MasterSend()	簡易 I²C データ送信 (LCD、加速度センサ)

4.4. 割り込み

Table 4-2 に、使用している割り込みを示します。

割り込みハンドラ中に IRQ5 による処理が必要であることを考慮し、IRQ5 は多重割り込みを許可しています。

Table 4-2 割り込み一覧

名称	発生タイミング	用途	多重割り込み	優先度
IRQ5	立ち下がりエッジ	ハードキー押下検知	許可	0x03
INTCMT0	周期割り込み	タッチキー検出 焦電型赤外線センサ検出	禁止	0x03
INTCMT1	周期割り込み	加速度センサ検出	禁止	0x03
INTIIC	簡易 I²C 通信割り込み	簡易 I²C 送信完了	禁止	0x03

4.5. 動作フロー

4.5.1. main ループ、割り込みハンドラ

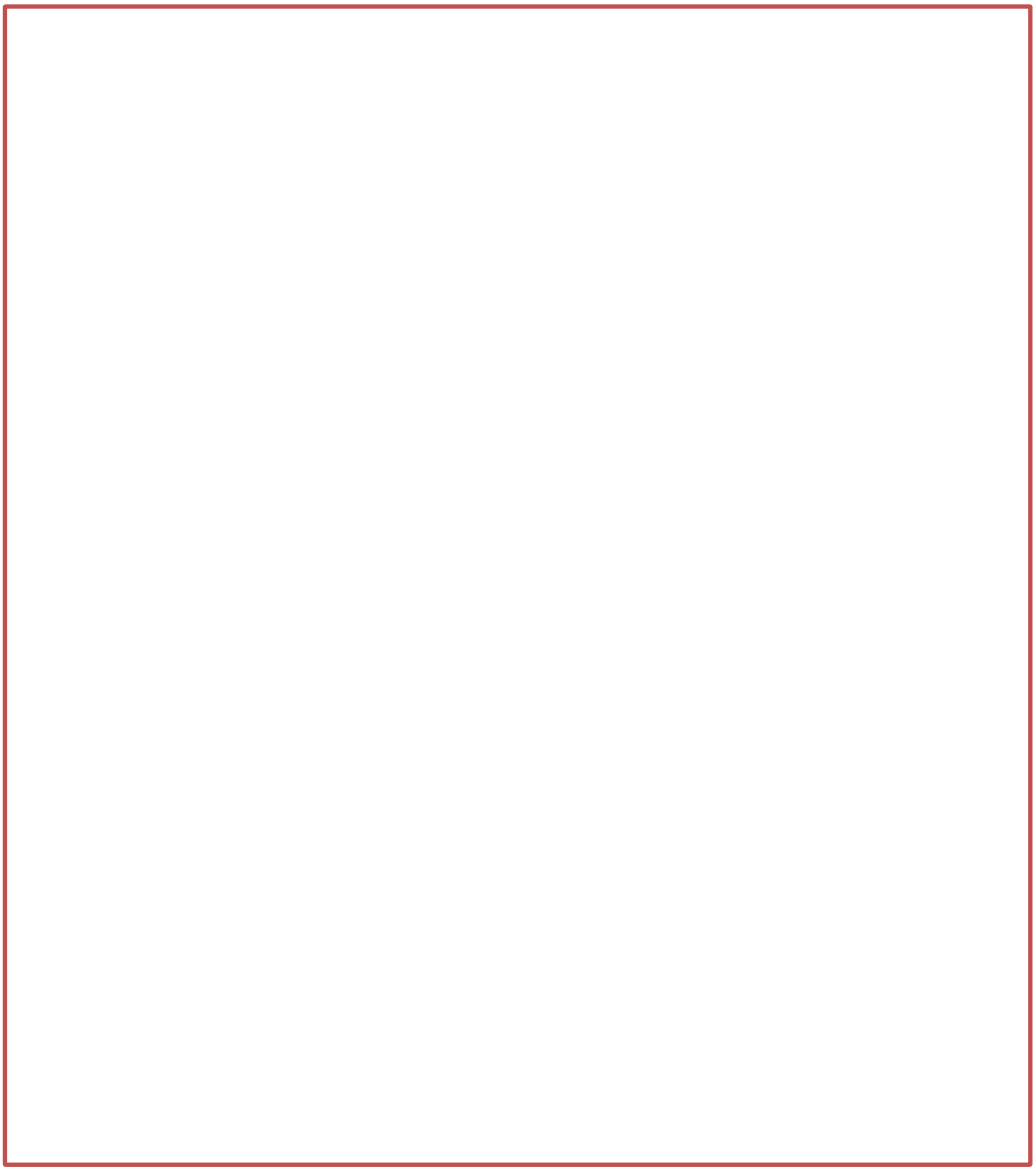


Figure 4-2 フローチャート

4.5.2. R_dapl_task 関数

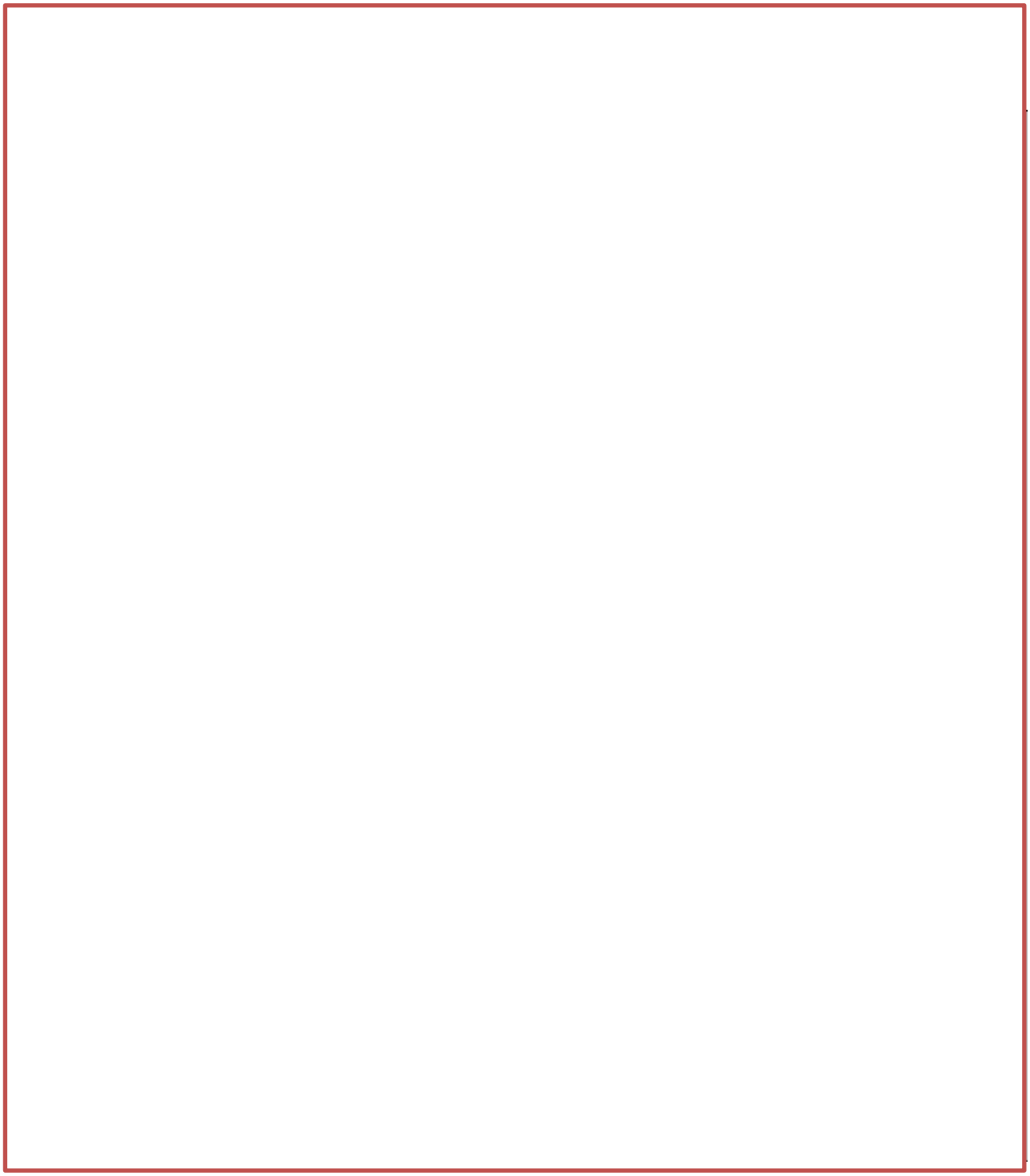


Figure 4-3 R_dapl_task()フローチャート

4.5.3. アプリケーションイベント

4.6. 状態遷移

4.6.1. 状態一覧

Table 4-4 状態一覧

4.6.2. 状態遷移図

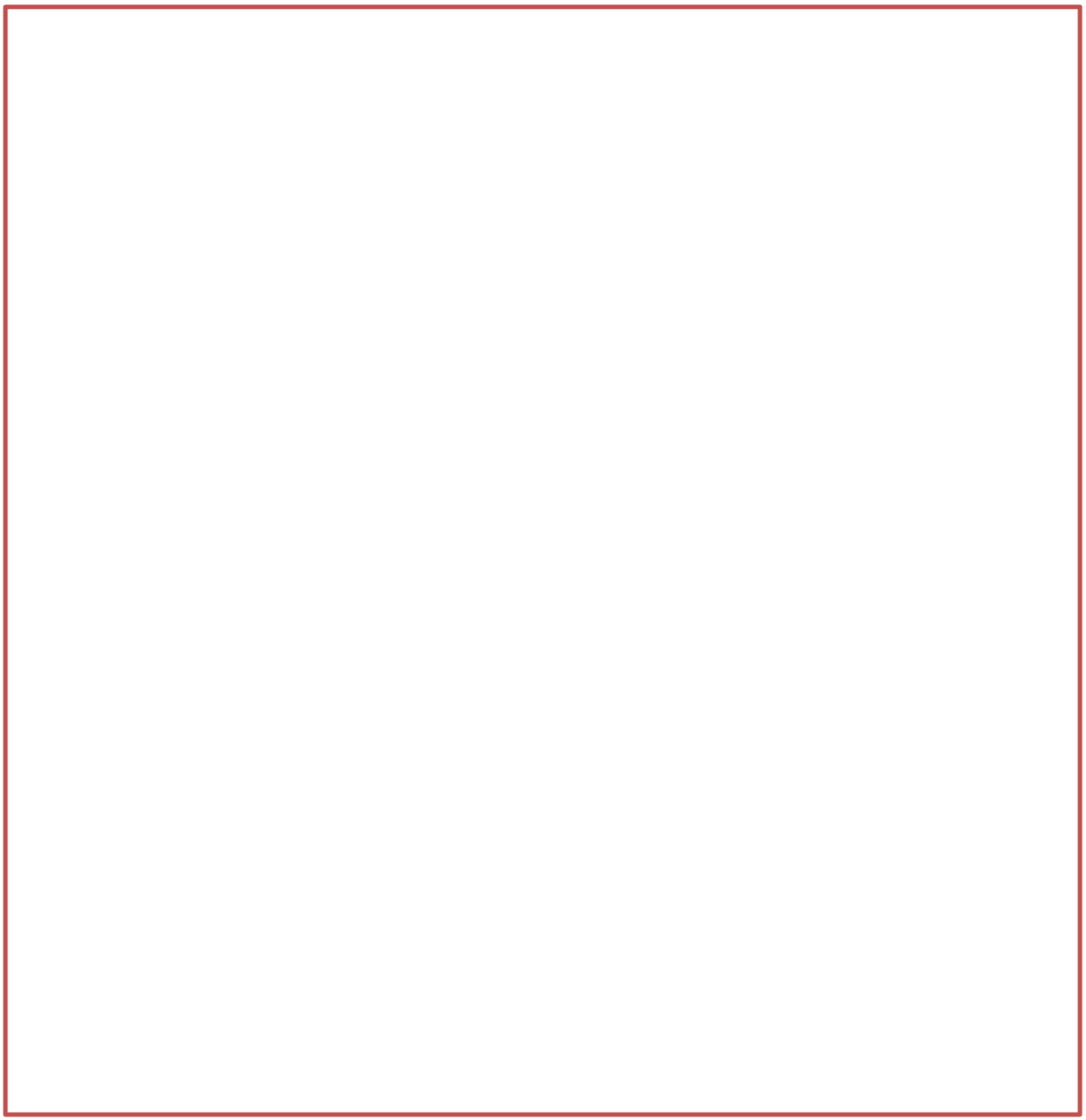


Figure 4-4 アプリケーション状態遷移図

4.7. ファイル構成

Table 4-6 に、基本デモソフトウェアのファイル構成 を示します。ビルドに必要なファイルのみ記載します。FIT モジュール内のファイル構成については 1.1 関連ドキュメントを参照下さい。

Table 4-6 ファイル構成

フォルダ/ファイル名	概要
RX231kit	プロジェクトフォルダ
.cproject	e²studio プロジェクト
.HardwareDebuglinker	e²studio プロジェクト
.info	e²studio プロジェクト
.project	e²studio プロジェクト
makefile.init	e²studio プロジェクト
.settings	e²studio 設定フォルダ
APL	アプリケーションフォルダ
demo_src	デモソースフォルダ
demo_fw.c	デモファームウェア
leddriver.c	LED ドライバ
main.c	デモメイン関数
main_apl.c	デモアプリケーション
inc	インクルードフォルダ
demo_apl.h	デモファームウェア/アプリケーションヘッダ
leddriver.h	LED ドライバヘッダ
FIT	FIT フォルダ
r_bsp	bsp FIT モジュール一式
r_cmt_rx	cmt FIT モジュール一式
r_config	FIT config ファイル一式
r_irq_rx	irq FIT モジュール一式
r_sci_iic_rx	siic FIT モジュール一式
MW	ミドルウェアフォルダ

touch	タッチキー制御ソフトウェアフォルダ
r_touch_API.c	**タッチキー制御 API**
CTSU	タッチキー制御ソフトウェア
r_ctsu.c	
r_ctsu_physical_driver.c	
DTC	
r_dtc.c	
HwResource	
r_mpc.c	
Include	
r_cg_userdefine.h	
r_data_output.h	
r_touch_API.h	
CTSU	
r_ctsu.h	
DTC	
r_dtc.h	
HwResource	
Touch	
r_touch.h	
Workbench	
r_mcu_model_define.h	
r_serial_control.h	
Touch	
r_touch.c	
Workbench	
r_serial_control.c	

4.8. センサ制御

4.8.1. 焦電型赤外線センサ(IRA-S210ST01)

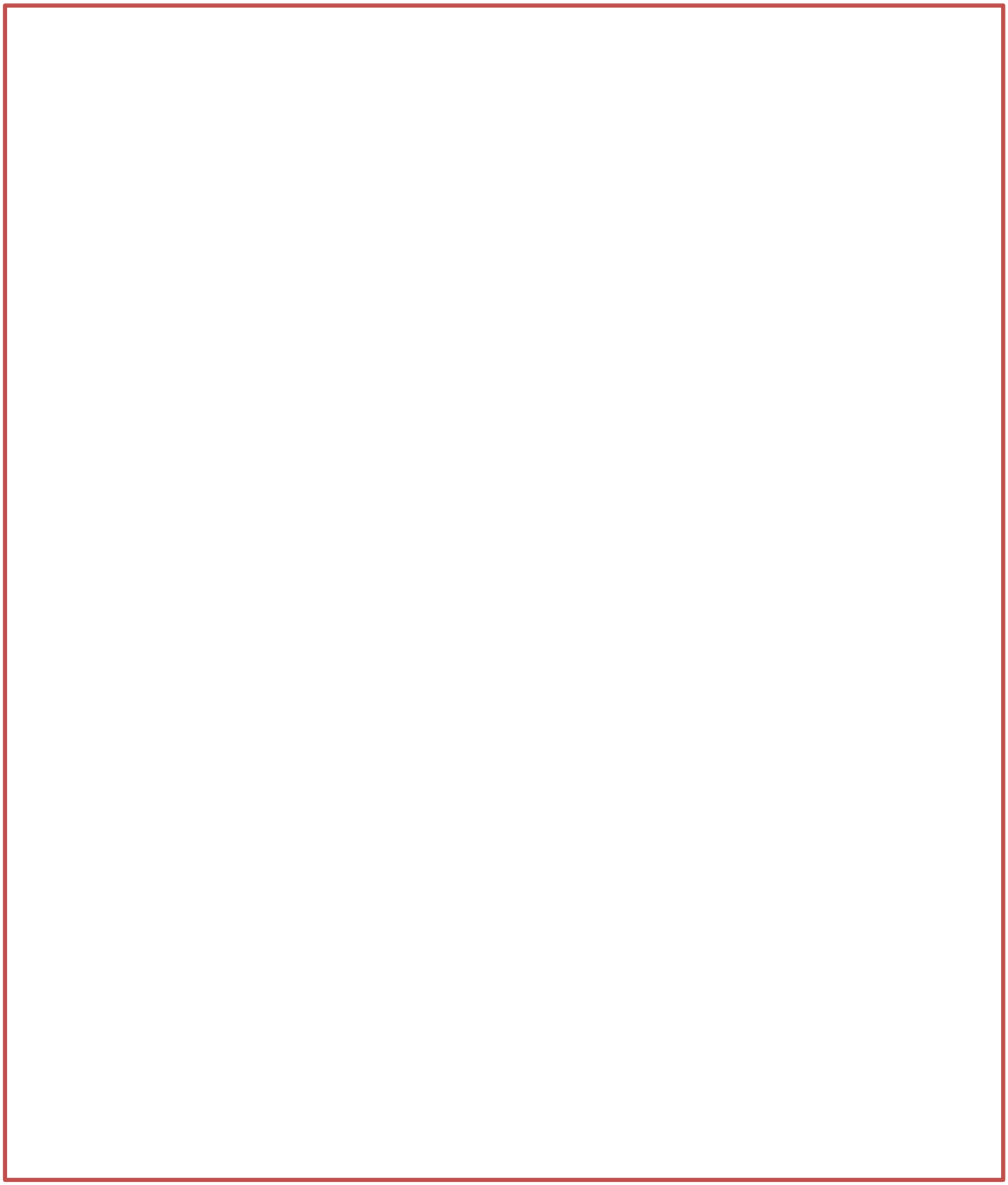


Figure 4-5 dapl_mortion_detect()フローチャート

4.8.2. 加速度センサ(KXSS5-2057)

5. 定数一覧

Table 5-1 に、ファームウェア層が定義する定数の一覧を示します。demo_apl.h で定義しています。FIT モジュール及びサンプルプログラム内の定数については 1.1 関連ドキュメントを参照下さい。

Table 5-1 ファームウェア層定数一覧

定数名	設定値	内容
INPUT_KEY_DATA	PORTD.PIDR.BIT.B5	ポート名定義
PORT_LOW	0	同上
PORT_HIGH	1	同上
PORT_OUT_MODE	1	同上
PORT_IN_MODE	0	同上
CMT0_FREQ_HZ	50	cmt 周波数設定(20ms)
CMT1_FREQ_HZ	1000	cmt 周波数設定(1ms)
SLAVE_ADDR_LCD	0x7C>>1	LCD デバイスアドレス
ACCESS_ADDR_LCD	0	-
ST7032I_RS0	0x00	LCD コマンド
ST7032I_RS1	0x40	LCD コマンド
ST7032I_C0	0x80	LCD コマンド
LCD_1LINE_START_ADDRESS	0x80	LCD 表示開始アドレス
LCD_2LINE_START_ADDRESS	0xC0	LCD 表示開始アドレス
LCD_BUFFER_SIZE	0xC0	簡易 I²C 転送バッファ
LCD_BACKRIGHT_FUNCTION	PORTB.PDR.BIT.B3	LCD バックライト
LCD_BACKRIGHT_PORT	PORTB.PODR.BIT.B3	LCD バックライト
LCD_RESET_FUNCTION	PORTB.PDR.BIT.B2	LCD リセットポート
LCD_RESET_PORT(a)	PORTB.PODR.BIT.B2 = a	LCD リセットポート
LCD_RESET_ON	0	-
LCD_RESET_OFF	1	-
LCD_RESET_WAIT	2	-

Table 5-2 に、アプリケーション層が定義する定数の一覧を示します。demo_apl.h で定義しています。

Table 5-2 アプリケーション層定数一覧

定数名	設定値	内容
SOFTWARE_VERSION	X.XX	ソフトウェアバージョン
LCD_1LINE	0	LCD 表示 line 指定
LCD_2LINE	1	LCD 表示 line 指定
LCD_SIZE	32	LCD 表示サイズ
TOUCH_DETECT	1	タッチキー検出
SENSOR_DETECT	1	センサ検出
NO_DETECT	0	未検出

6. 型・構造体一覧

変数型は以下の定義を使用しています。

```
typedef unsigned char       uint8_t;    /*  8bit */
typedef unsigned short      uint16_t;   /* 16bit */
typedef unsigned long       uint32_t;   /* 32bit */
typedef signed char         int8_t;
typedef signed short        int16_t;
typedef signed long         int32_t;
```

enum 型は以下の定義を使用しています。

```
/* タッチキーイベント */
typedef enum
{
    DAPL_TOUCH_NO_DETECT,
    DAPL_TOUCH_BACK,
    DAPL_TOUCH_ENTER,
    DAPL_TOUCH_UP,
    DAPL_TOUCH_DOWN,
    DAPL_TOUCH_RIGHT,
    DAPL_TOUCH_LEFT,
    DAPL_TOUCH_MAX,
} TOUCH_EVENT;
```

```
/* アプリケーション構造体 */
typedef struct
{
    TOUCH_EVENT     touch_action;
    uint8_t         touch_action_flag;
    uint16_t        led1_count;
    uint16_t        led2_count;
} dapl_param_t;
```

アプリケーション層の構造体 dapl_param_t を Table 6-1 に記します。demo_apl.h で定義しています。
FIT モジュール、サンプルプログラム内の構造体については 1.1 関連ドキュメントを参照下さい。

Table 6-1 アプリケーション構造体 dapl_param_t

データ型	メンバ	内容
TOUCH_EVENT	touch_action	タッチキーイベント
uint8_t	touch_action_flag	タッチキー検出フラグ
uint16_t	led1_count	LED1 点灯時間カウンタ
uint16_t	led2_count	LED2 点灯時間カウンタ

7. グローバル変数一覧

Table 7-1 にファームウェア層、Table 7-2 にアプリケーション層が定義するグローバル変数を示します。FIT モジュール、サンプルプログラム内で定義しているグローバル変数については割愛しています。

Table 7-1 ファームウェア層グローバル変数一覧

データ型	変数名	内容
sci_iic_info_t	g_iic_info_lcd	LCD 通信用構造体
uint8_t	g_slave_addr_lcd[1]	LCD デバイスアドレス格納バッファ
uint8_t	g_access_addr_lcd[1]	LCD データバッファ
uint8_t	g_lcd_init[10]	LCD 初期設定

Table 7-2 アプリケーション層グローバル変数一覧

データ型	変数名	内容
dapl_param_t	g_dapl_param	アプリケーション管理用構造体変数

8. API 関数一覧

Table 8-1、Table 8-2 に、API 関数の一覧を示します。

Table 8-1 API 関数一覧 (1/2)

関数	機能
MCU (RX231)	
void R_dfw_wait_us(uint16_t time)	ソフトウェアウェイト [us]
void R_dfw_wait_ms(uint16_t time)	ソフトウェアウェイト [ms]
void R_LED_Open(void)	LED 初期化
void R_dfw_led1_on(uint16_t cnt)	LED1 点灯
void R_dfw_led1_off(void)	LED1 消灯
void R_dfw_led2_on(uint16_t cnt)	LED2 点灯
void R_dfw_led2_off(void)	LED2 消灯
uint8_t R_dfw_get_key_status(void)	ハードキー状態取得
void R_dfw_port_init(void)	ポート初期化
void R_dfw_irq_init(void)	IRQ 初期化
void R_dfw_cmt_init(void)	CMT 初期化
void R_dfw_siic6_init(void)	簡易 I²C 初期化
void R_dfw_apl_init(void)	周辺機能初期化
LCD 制御 (NHD-C0216CiZ-FSW-FBW-3V3)	
void R_dfw_lcd_init(void)	LCD 初期化
void R_dfw_lcd_on(void)	LCD バックライト点灯
void R_dfw_lcd_off(void)	LCD バックライト消灯
void R_dfw_lcd_write_str(uint8_tl uint8_t c, uint8_t len, uint8_t *str)	LCD 文字列表示
SAIC101 制御 (RAA7301013CBG)	
加速度センサ制御 (KXSS5-2057)	

Table 8-2 API 関数一覧 (2/2)

関数	機能
Serial Flash(MX25L6433FM2I-08G)	
SOUND	
Touch	
void R_dfw_touch_init(void)	タッチキー初期化
uint8_t dapl_touch_detect(void)	タッチキー検出
Walk	

9. API 関数詳細

本章では、8 章で示した API 関数の詳細について説明します。

9.1. MCU 関数

9.1.1. R_dfw_wait_us

形式	void R_dfw_wait_us(uint16_t time)		
引数	uint16_t	Time	Wait time(*1us)
戻り値	void	ー	
仕様	引数の数値に応じてソフトウェアウェイトします。 例えば 500 の場合は、500[us]ウェイトします。		
注意事項	Renesas RXC Toolchain V2.04.01 かつメインクロックが 32MHz の場合に合わせて作成しています。開発環境およびメインクロックが変更となる場合は、NOP の数を変更するなどして命令数を調整してください。		

9.1.2. R_dfw_wait_ms

形式	void R_dfw_wait_ms(uint16_t time)		
引数	uint16_t	Time	Wait time(*1ms)
戻り値	void	ー	
仕様	引数の数値に応じてソフトウェアウェイトします。 例えば 500 の場合は、500[ms]ウェイトします。		
注意事項	Renesas RXC Toolchain V2.04.01 かつメインクロックが 32MHz の場合に合わせて作成しています。開発環境およびメインクロックが変更となる場合は、NOP の数を変更するなどして命令数を調整してください。		

9.1.3. R_LED_Open

形式	void R_LED_Open(void)		
引数	void	ー	ー
戻り値	void	ー	
仕様	LED1,LED2 のポート初期設定を行います。		

9.1.4. R_dfw_led1_on

形式	void R_dfw_led1_on(uint16_t cnt)		
引数	uint16_t	cnt	0>:点灯時間[ms] 0 :連続点灯
戻り値	void	ー	
仕様	LED1(橙)を点灯します。 引数 cnt により点灯する時間を[ms]単位で指定できます。 引数が 0 の場合は連続点灯します。 ※引数 cnt には「指定したい時間 + 1」を設定して下さい。 ※最大+1ms の誤差が発生します		

9.1.5. R_dfw_led1_off

形式	void R_dfw_led1_off(void)		
引数	void	ー	ー
戻り値	void	ー	
仕様	LED1(橙)を消灯します。		

9.1.6. R_dfw_led2_on

形式	void R_dfw_led2_on(uint16_t cnt)		
引数	uint16_t	cnt	0>:点灯時間[ms] 0 :連続点灯
戻り値	void	ー	
仕様	LED2(緑)を点灯します。 引数 cnt により点灯する時間を[ms]単位で指定できます。 引数が 0 の場合は連続点灯します。 ※引数 cnt には「指定したい時間 + 1」を設定して下さい。 ※最大+1ms の誤差が発生します		

9.1.7. R_dfw_led2_off

形式	void R_dfw_led2_off(void)		
引数	void	ー	ー
戻り値	void	ー	
仕様	LED2(緑)を消灯します。		

9.1.8. R_dfw_get_key_status

形式	uint8_t R_dfw_get_key_status(void)		
引数	void	ー	ー
戻り値	uint8_t	0 : 押下状態 1 : 非押下状態	
仕様	ハードキーの状態を返却します。		

9.1.9. R_dfw_port_init

形式	void R_dfw_port_init(void)		
引数	void	ー	ー
戻り値	void	ー	
仕様	ポートの初期設定を行います。 使用しているポートについては、各 FIT モジュール内で初期設定をしています。		

```
    /* unused port */
    PORTJ.PDR.BIT.B3 = 1;
    PORT3.PDR.BYTE   = 0x1b;
    PORT2.PDR.BYTE   = 0xdf;
    PORT1.PDR.BYTE   = 0xa4;
    PORTA.PDR.BIT.B0 = 1;
    PORTE.PDR.BYTE   = 0xff;
    PORTD.PDR.BYTE   = 0xd8;
    PORT4.PDR.BYTE   = 0x06;
    PORT0.PDR.BIT.B7 = 1;

    SYSTEM.VBATTCR.BIT.VBATTDIS = 1;
```

9.1.10. R_dfw_usb_init

形式	void R_dfw_usb_init(void)		
引数	void	ー	ー
戻り値	void	ー	
仕様	USB モジュールの初期設定を行います。 USB FIT の R_USB_Open()で初期設定を行います。		

9.1.11. R_dfw_irq_init

形式	void R_dfw_irq_init(void)		
引数	void	ー	ー
戻り値	void	ー	
仕様	IRQ モジュールの初期設定を行います。 IRQ FIT の R_IRQ_Open()で以下の初期設定を行います。		

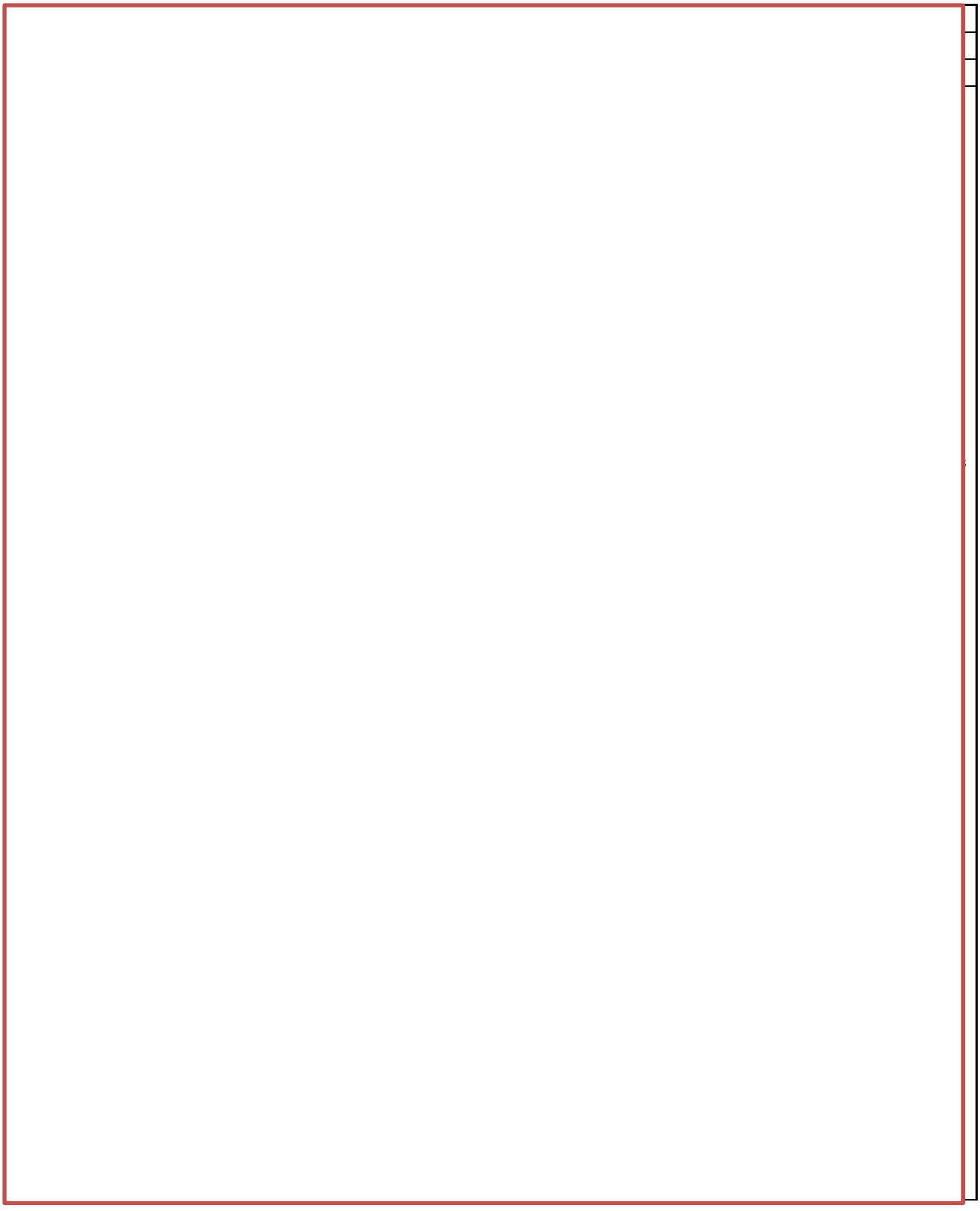
IRQ5:立ち下がりエッジ、優先度 3、割り込みハンドラ dapl_intirq5_handler 登録

```
irq_handle_t irq_handle_dummy;

/* IRQ port setting */
MPC.PWPR.BIT.B0WI   = 0;            /* PFSWE write enable */
MPC.PWPR.BIT.PFSWE  = 1;            /* PFS register write enable */
```

```
MPC.PD5PFS.BIT.ISEL = 1;            /* IRQ5 MPC setting */

MPC.PWPR.BIT.PFSWE  = 0;            /* PFS register write disable */
MPC.PWPR.BIT.B0WI   = 1;            /* PFSWE write disable */
```

```
R_IRQ_Open(IRQ_NUM_5, IRQ_TRIG_FALLING, IRQ_PRI_3, &irq_handle_dummy,
           &dapl_intirq5_handler);
```

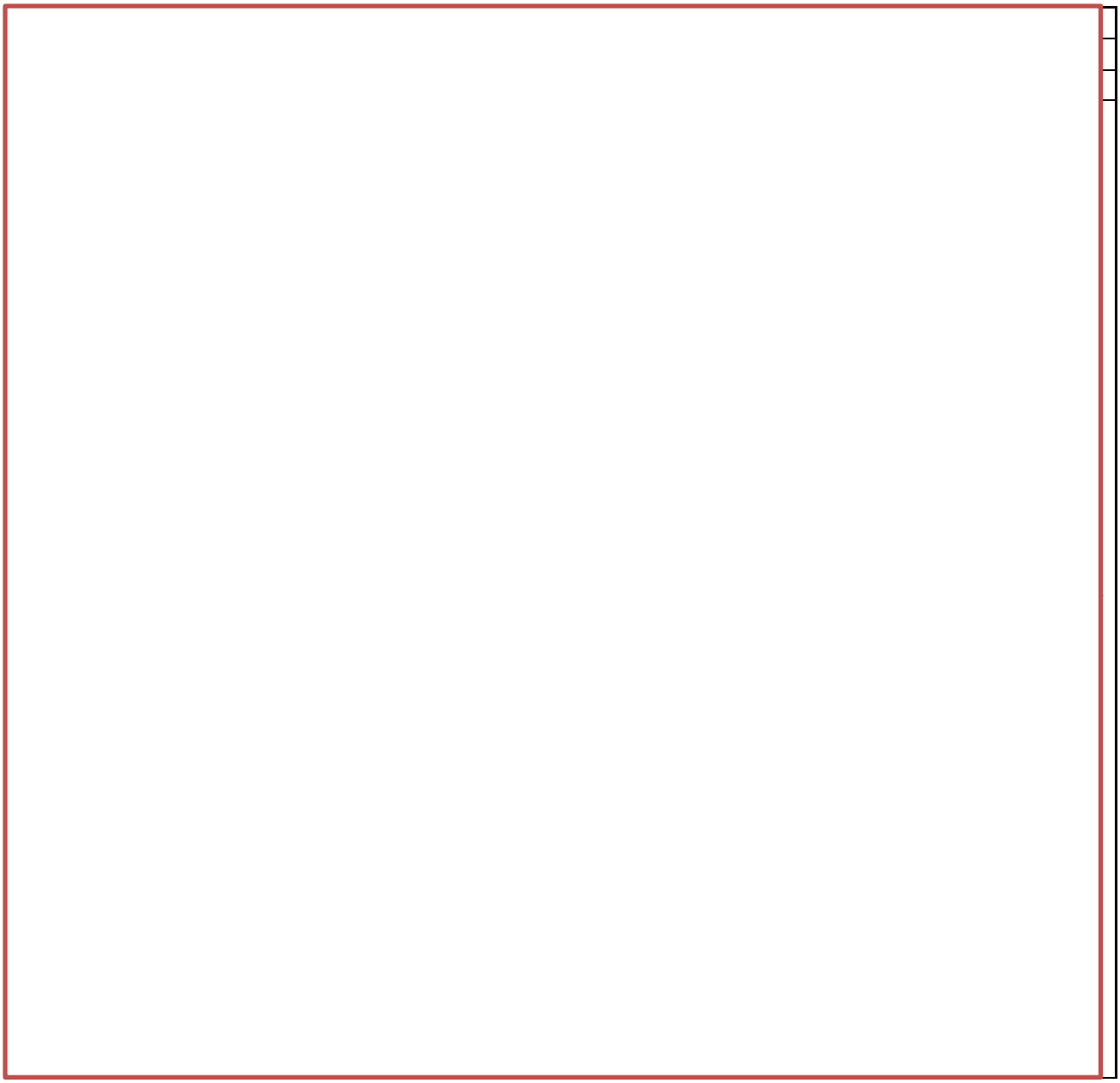
9.1.12. R_dfw_rtc_init



9.1.13. R_dfw_cmt_init

形式	void R_dfw_cmt_init(void)		
引数	void	ー	ー
戻り値	void	ー	
仕様	CMT モジュールの初期設定を行います。 CMT FIT の R_CMT_CreatePeriodic()で cmt0,cmt1 の初期設定を行います。 cmt0： ・割り込み周期 20ms ・割り込みハンドラ dapl_intcmt0_handler 登録 cmt1： ・割り込み周期 1ms ・割り込みハンドラ dapl_intcmt1_handler 登録 /* Start open module */ /* system cmt */ R_CMT_CreatePeriodic(CMT0_FREQ_HZ, dapl_intcmt0_handler, &cmt_ch); /* walk apl cmt */ R_CMT_CreatePeriodic(CMT1_FREQ_HZ, dapl_intcmt1_handler, &cmt_ch);		

9.1.14. R_dfw_adc_init



9.1.15. R_dfw_dac_init

9.1.16. R_dfw_siic6_init

形式	void R_dfw_siic6_init(void)		
引数	void	ー	ー
戻り値	void	ー	

仕様	簡易 I²C モジュールの初期設定を行います。
	SCI IIC FIT の R_SCI_IIC_Open()で初期設定を行います。
	・簡易 I²C6 チャンネル設定
	・割り込みハンドラ dapl_intiic_handler 登録
	
	R_BSP_RegisterProtectDisable(BSP_REG_PROTECT_MPC);
	PORTB.PMR.BIT.B0 = 1;
	PORTB.PMR.BIT.B1 = 1;
	R_BSP_RegisterProtectEnable(BSP_REG_PROTECT_MPC);
	
	siic_info.dev_sts = 0;
	siic_info.ch_no = 6;
	siic_info.callbackfunc = &dapl_intiic_handler;
	
	ret = R_SCI_IIC_Open(&siic_info);		

9.1.17. R_dfw_sspi9_init

9.1.18. R_dfw_apl_init

形式	void R_dfw_apl_init(void)		
引数	void	ー	ー
戻り値	Void	ー	
仕様	周辺機器(LCD)の初期設定を行います。 ・R_lcd_init()コール		
注意事項	500[ms]のソフトウェアウェイトを含みます。		

9.2. LCD 制御関数

9.2.1. R_dfw_lcd_init

形式	void R_dfw_lcd_init(void)		
引数	void	ー	ー
戻り値	void	ー	
仕様	LCD を初期化します。 ・LCD リセットポート設定 ・LCD 初期設定 ・8-bit mode,2-line display mode ・OSC frequency 347kHz ・Contrast 0x70 set ・Display 点灯		
注意事項	9.2.2～9.2.4 の関数を使用する前にコールしてください。 40[ms]のソフトウェアウェイトを含みます。		

9.2.2. R_dfw_lcd_on

形式	void R_dfw_lcd_on(void)		
引数	void	ー	ー
戻り値	void	ー	
仕様	LCD バックライトを点灯します。		

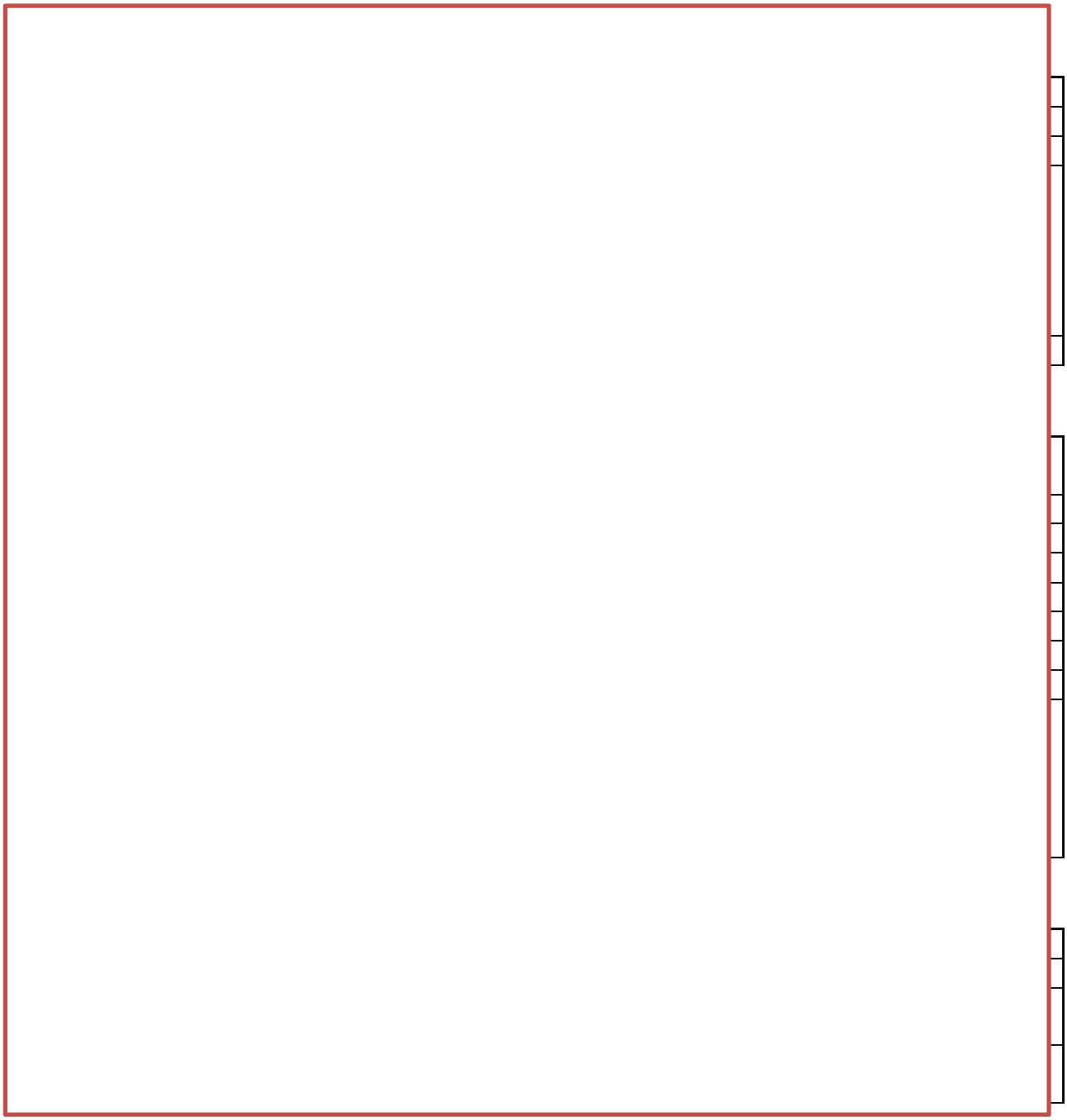
9.2.3. R_dfw_lcd_off

形式	void R_dfw_lcd_off(void)		
引数	void	ー	ー
戻り値	void	ー	
仕様	LCD バックライトを消灯します。		

9.2.4. R_dfw_lcd_write_str

形式	void R_dfw_lcd_write_str(uint8_t l, uint8_t c, uint8_t len, uint8_t * str)		
引数	uint8_t	l	LCD 表示 line
	uint8_t	c	表示開始 Column
	uint8_t	len	表示文字数
	uint8_t *	str	表示する文字列の先頭ポインタ
戻り値	void	ー	
仕様	LCD に引数 str の文字列を表示します。 表示開始列を引数 l、開始 Column を引数 c、表示文字数を引数 len で指定します。		
注意事項	引数 c と引数 len の合計値が 16(LCD 表示サイズ)を超えないようにしてください。		

9.3. SAIC101 制御関数



9.4. kxss5 (加速度センサ)制御関数

9.5. シリアルフラッシュ制御関数

9.6. 音声再生・録音制御関数

9.7. タッチキー制御関数

9.7.1. R_dfw_touch_init

形式	void R_dfw_touch_init(void)		
引数	void	ー	ー
戻り値	void	ー	
仕様	タッチキーの初期設定を行います。		
注意事項	9.7.2の関数を使用する前にコールしてください。		

9.7.2. dapl_touch_detect

形式	uint8_t dapl_touch_detect(void)		
引数	void	-	-
戻り値	uint8_t	0:非検出 1:検出	
仕様	タッチキーの検出/非検出を検出します。 一定間隔でコールしてください。 タッチキー検出情報は g_touch_all_result に格納されています。 静電容量タッチキーの検出方式は自己容量方式で検出しております。 詳細は静電容量式タッチセンサシステムを参照下さい。		

9.8. 歩数計制御関数

10. 注意事項

基本デモソフトウェアに実装されている USB ID(ベンダーID、プロダクト ID)はお客様の製品にはご使用できません。

USB-IF からベンダーID を必ず取得して下さい。

http://www.usb.org/home

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ
http://japan.renesas.com/

お問合せ先
http://japan.renesas.com/contact/

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2015.10.19	—	初版発行
1.10	2016.6.24	4 16 28	「Table 2-1 動作確認条件」を変更 「Table 4-6 ファイル構成」を変更 「9.1.1 R_dfw_wait_us」の注意事項を変更 「9.1.2 R_dfw_wait_ms」の注意事項を変更
1.20	2020.8.20	—	toolchain　version の更新

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。
3. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。
5. クロックについて
   リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1.本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2.本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1　2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。