RX231 Group

R11AN0015EJ0120
Rev.1.20
Aug.20.2020

RX231 HMI Solution Kit Base Demo Software (Function limited edition) RTK5RX2310P000F0ZR

Summary

This is an application note of the RX231 HMI Solution Kit Base Demo Software (Function limited edition) RTK5RX2310P000F0ZR.

Device for Operation Confirmation

RX231

Contents

1. Introduction

This application note outlines the functions of the RX231 Human Machine Interface Solution Kit R0K5RX231D000BR hardware using the RX231, the MCU manufactured by Renesas Electronics Corporation, and describes the usage of the basic demonstration software (RX231 HMI Solution Kit Base Demo Software (Function limited edition) RTK5RX2310P000F0ZR).

Red frame part in this book is a function-limited part.

1.1. Related Documents

[1] RX230 Group, RX231 Group User's Manual: Hardware (R01UH0496EJ)

[2] RX231 Human Machine Interface Solution Kit R0K5RX231D000BR Instruction Manual (R01AN2586EJ0100)

Firmware Integration Technology (FIT) used:

[3] IRQ (External Pin Interrupt Request) Module Using Firmware Integration Technology
[4] CMT Module Using Firmware Integration Technology
[5] Simple I^2C Module Using Firmware Integration Technology

Sample programs used:

[6] Capacitive touch sensor system

The documents above are available on the Renesas Electronics website.

- Renesas Electronics website
  [http://www.renesas.eu/]

Capacitive touch sensor system

[http://www.renesas.eu/applications/key_technology/human_interface/touch_sensor_top/index.jsp]

1.2. Terms and Abbreviations

The following terms and abbreviations are used in this document.

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. Operation Confirmation Conditions

Table 2-1 shows the conditions under which the operations of the basic demonstration software are confirmed.

Table 2-1 Operation Confirmation Conditions

Item	Descriptions
Hardware	Renesas Electronics RX231 Human Machine Interface Solution Kit R0K5RX231D000BR
Integrated development environment	Renesas Electronics e²studio 2020-04
Toolchain	Renesas Electronics Renesas RXC Toolchain V3.02.00
Emulator	Renesas Electronics E1 emulator
Operating frequency	Main system clock: 32 MHz (crystal oscillator: 16 MHz) Subsystem clock: 32.768 kHz (crystal oscillator)
Operating voltage	3.2 V

Notes:

If the same version of the toolchain (C compiler) specified in the original project is not in the import destination, the toolchain will not be selected and an error will occur.

Check the selected status of the toolchain on the project configuration dialog.

For the setting method, refer to FAQ 3000404.

FAQ 3000404 :Program ""make"" not found in PATH’ error when attempting to build an imported project (e² studio)"

3. Hardware

This section outlines the hardware.

For details of the hardware, refer to the RX231 HMI Solution Kit Instruction Manual, given as the related document [2].

3.1. Block Diagram

Figure 3-1 shows a block diagram of the R0K5RX231D000BR hardware.

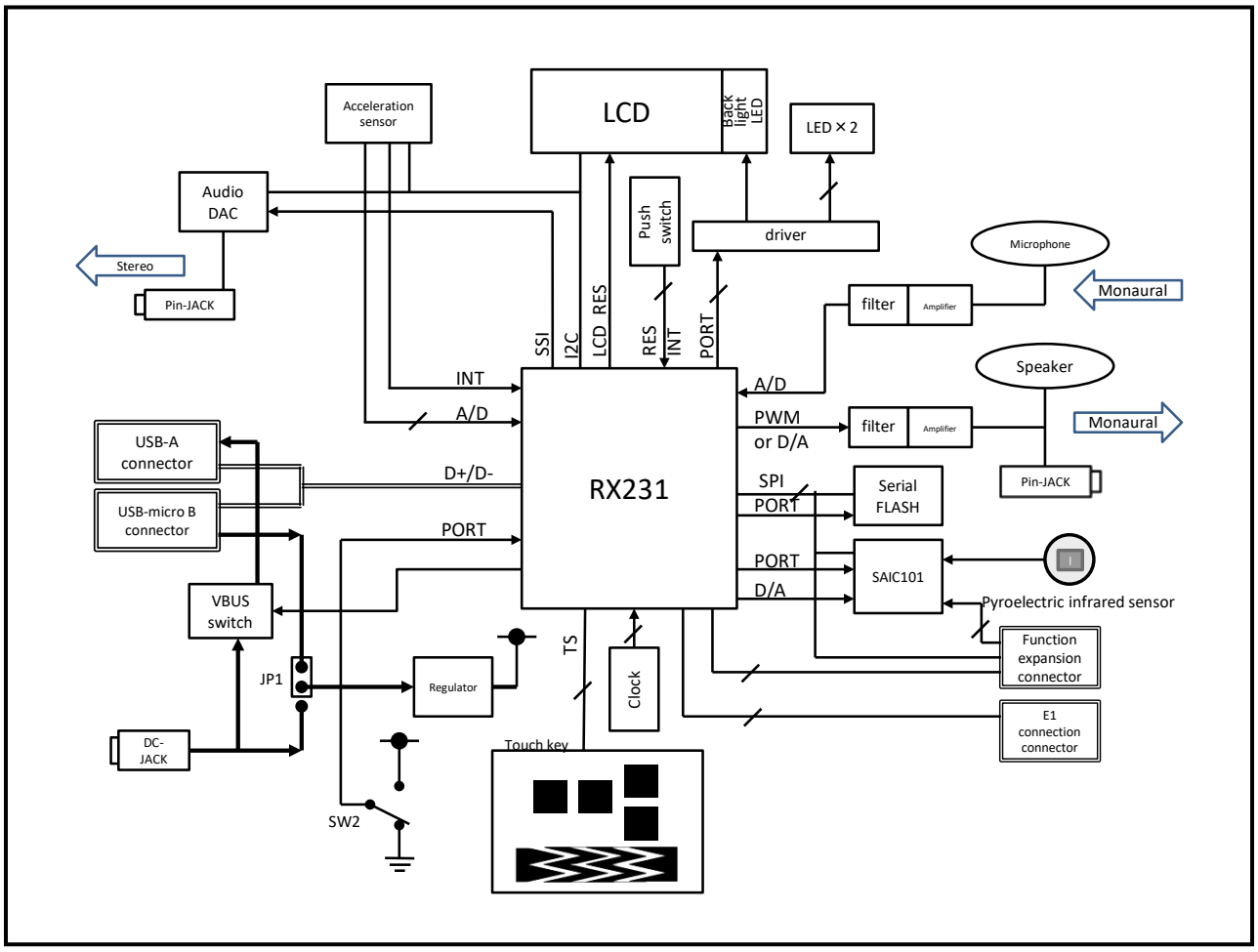
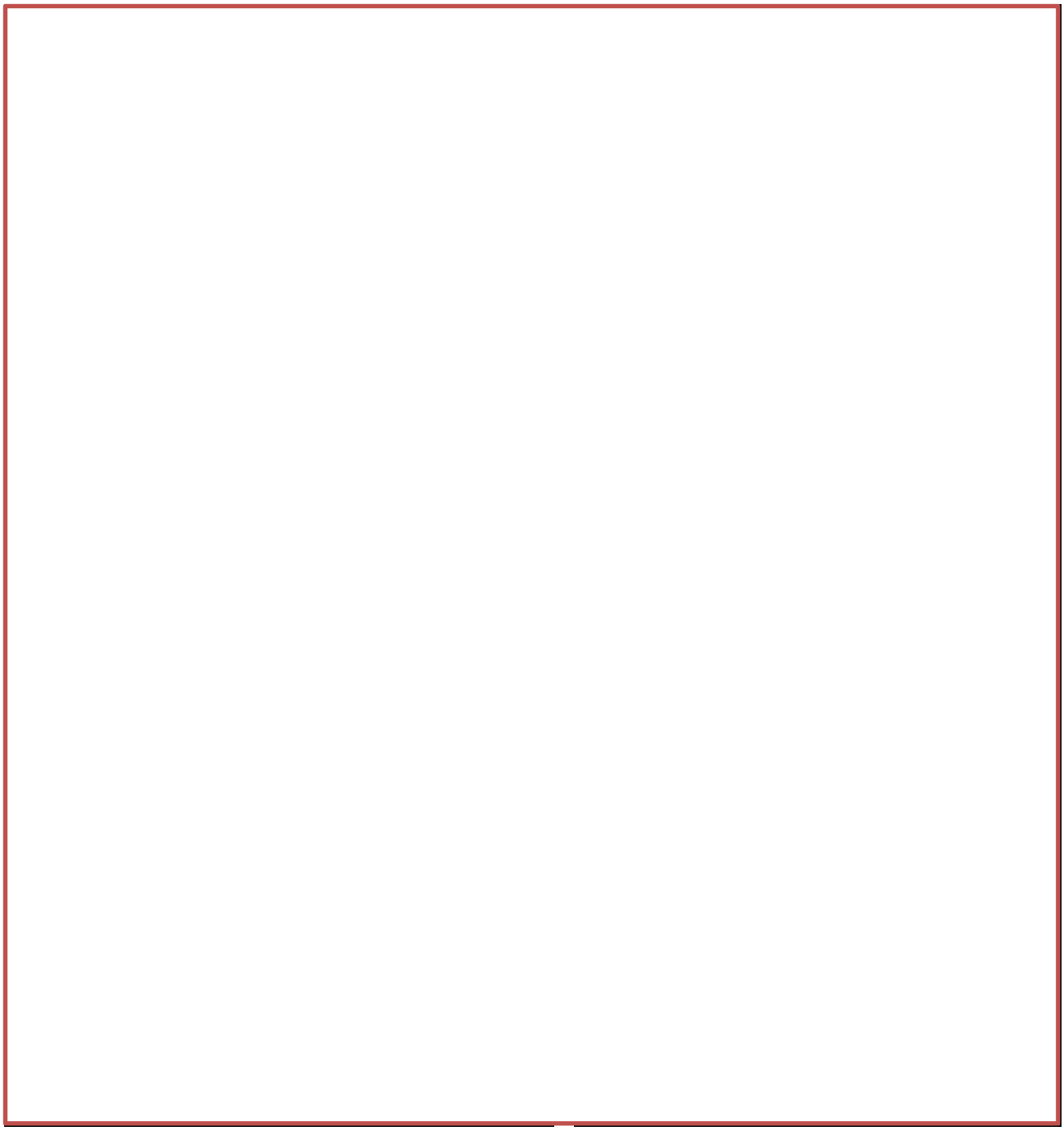


Figure 3-1 Block Diagram of R0K5RX231D000BR Hardware

3.2. Pin Assignments

Table 3-1 Pin Assignments (1/2)

Table 3-2 Pin Assignments (2/2)



4. Basic Demonstration Software

The demonstration software is based on the Firmware Integration Technology (hereinafter referred to as FIT) module offered by Renesas. It is an e² studio project not using the RTOS.

4.1. Operation Overview

After start-up, the PowerON_Reset_PC() function is called and then the main() function is called. In the PowerON_Reset_PC() function, MCU clock is set. In the main() function, the initial settings are made for the MCU peripheral functions, ports, and peripheral modules.

After the initial settings, go to mainloop.

4.2. Software Configuration

Figure 4-1 shows the software configuration of the basic demonstration software.

The API functions provided by the Device Firmware(Firmware Integration Technology) are used for the initial settings, start, and stop of the various RX231 capabilities and for control of the utility capabilities. For the application layer, the API functions provided by the Device Firmware(Firmware Integration Technology) as well as the sample programs including Capacitive Touch Key are used.

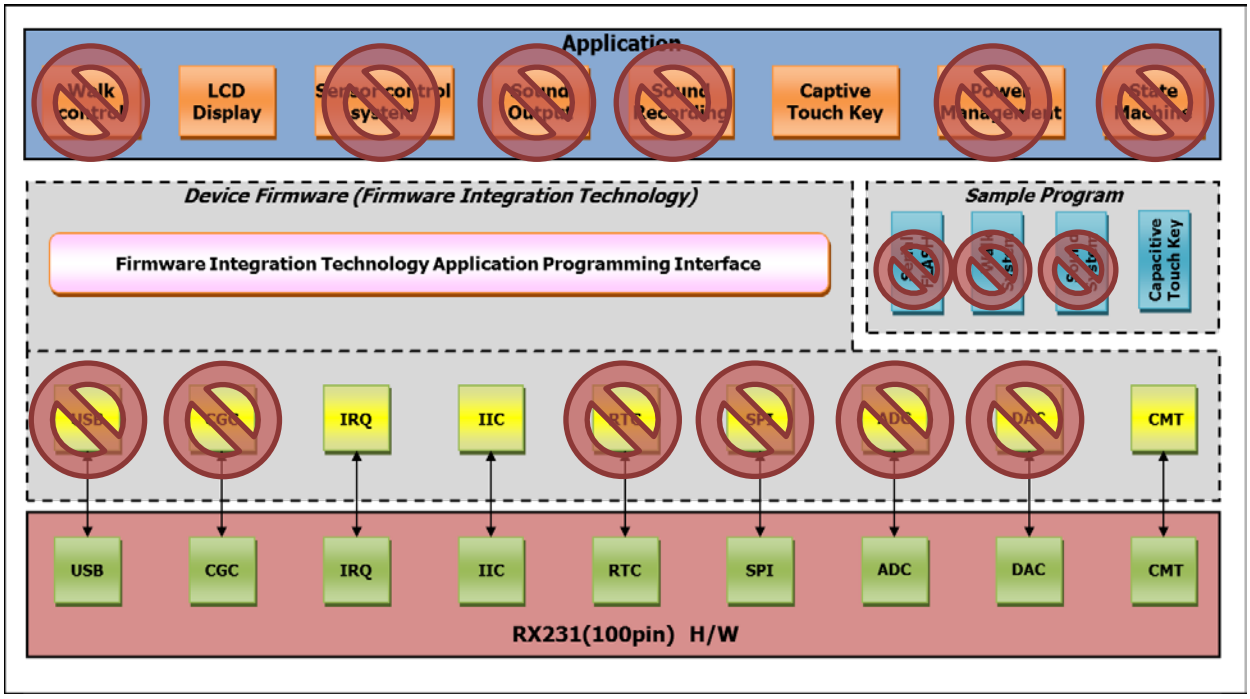


Figure 4-1 Software Configuration

4.3. FIT Modules

Table 4-1 lists the FIT modules used with the basic demonstration software. For details of each FIT module, refer to the appropriate document given in section 1.1, **Related Documents**.

Table 4-1 List of FIT Modules

FIT Module	Function Name	Use
IRQ	R_IRQ_Open()	Sets and starts external interrupts. Registers interrupt handlers. • Hardware key
CMT	R_CMT_CreatePeriodic()	Initializes the CMT module. Registers interrupt handlers. Sets and starts the cmt0 and cmt1 interrupts.
IIC	R_SCI_IIC_Open()	Initializes the simple I^2C module. Registers interrupt handler.
	R_SCI_IIC_MasterSend()	Transmits the simple I^2C data (LCD and acceleration sensor).

4.4. Interrupts

Table 4-2 lists the interrupts used.

Note that multiple interrupts are allowed for IRQ5 since IRQ5 is required for processes in the interrupt handlers.

Table 4-2 List of Interrupts

Name	Generation Timing	Use	Multiple Interrupt	Priority
IRQ5	Falling edge	Detects hardware key push-down.	Allowed	0x03
INTCMT0	Periodic interrupt	Detects the touch key. Detects the pyroelectric infrared sensor.	Prohibited	0x03
INTCMT1	Periodic interrupt	Detects the acceleration sensor.	Prohibited	0x03
INTIIC	Simple I^2C communication interrupt	Completes simple I^2C transmission.	Prohibited	0x03

4.5. Operation Flow

4.5.1. main Loops and Interrupt Handlers

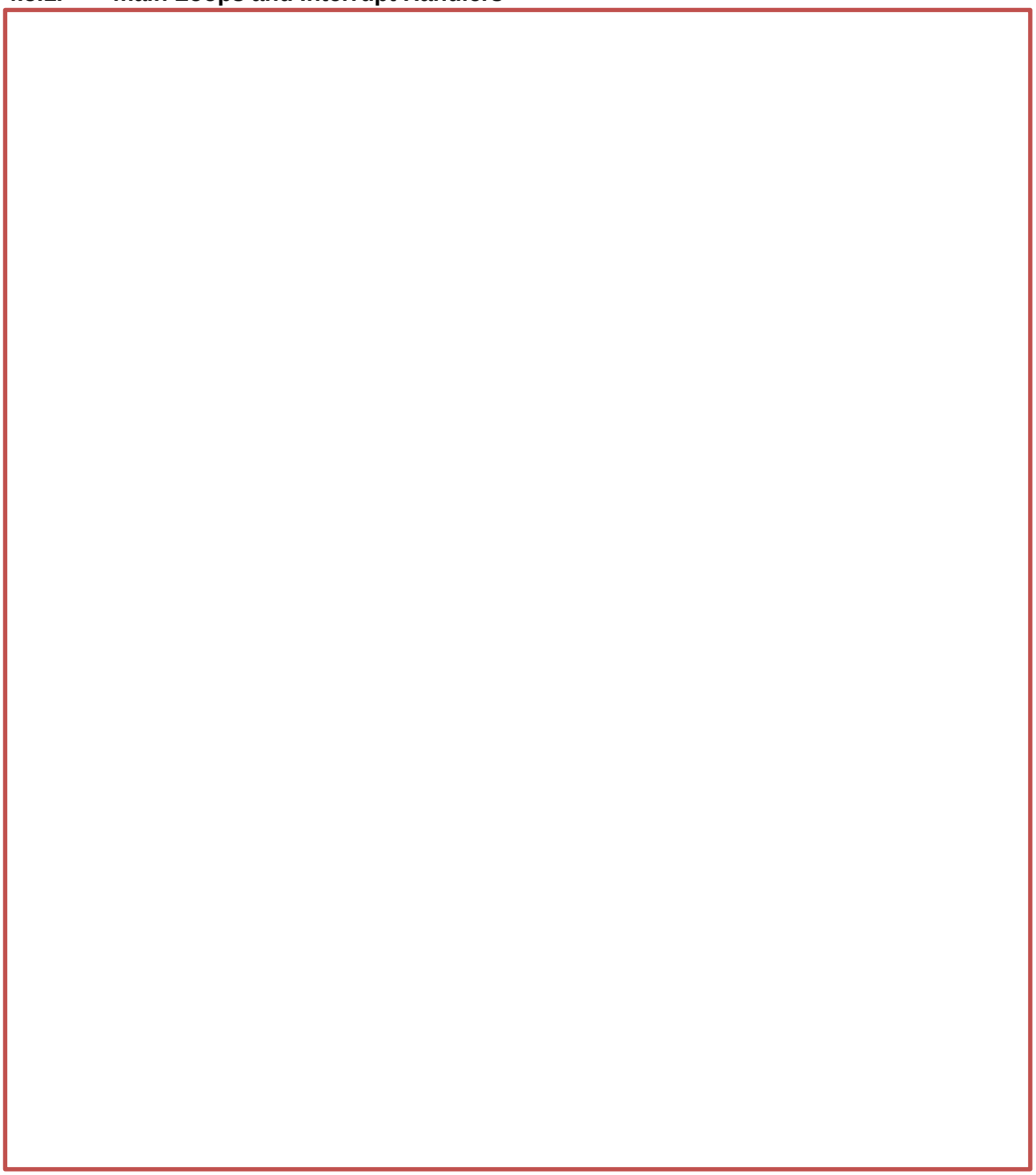


Figure 4-2 Flowcharts

4.5.2. R_dapl_task Function

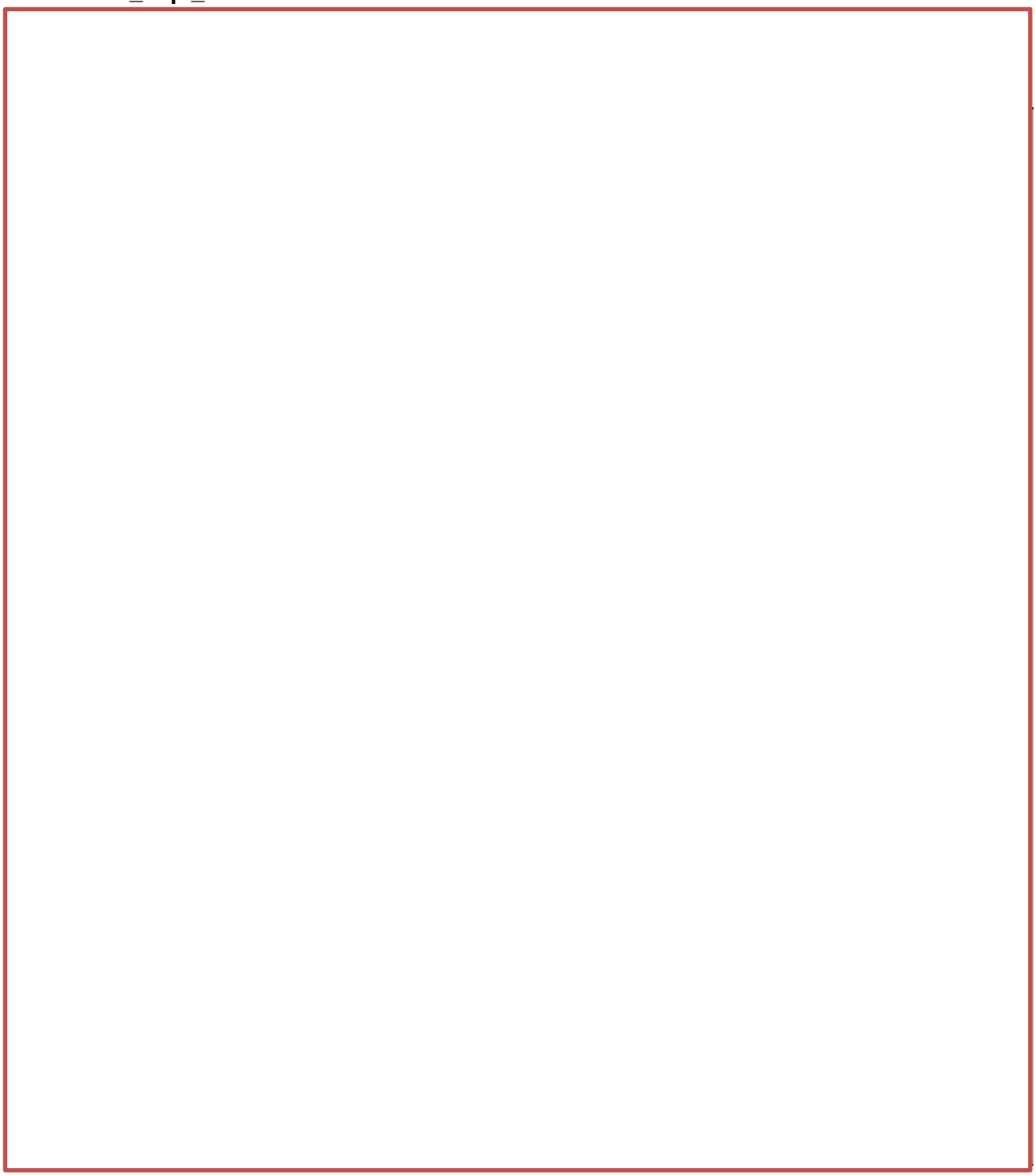


Figure 4-3 R_dapl_task() Flowchart

4.5.3. Application Events

4.6. State Transitions

4.6.1. List of States

Table 4-4 List of States

4.6.2. State Transition Diagram

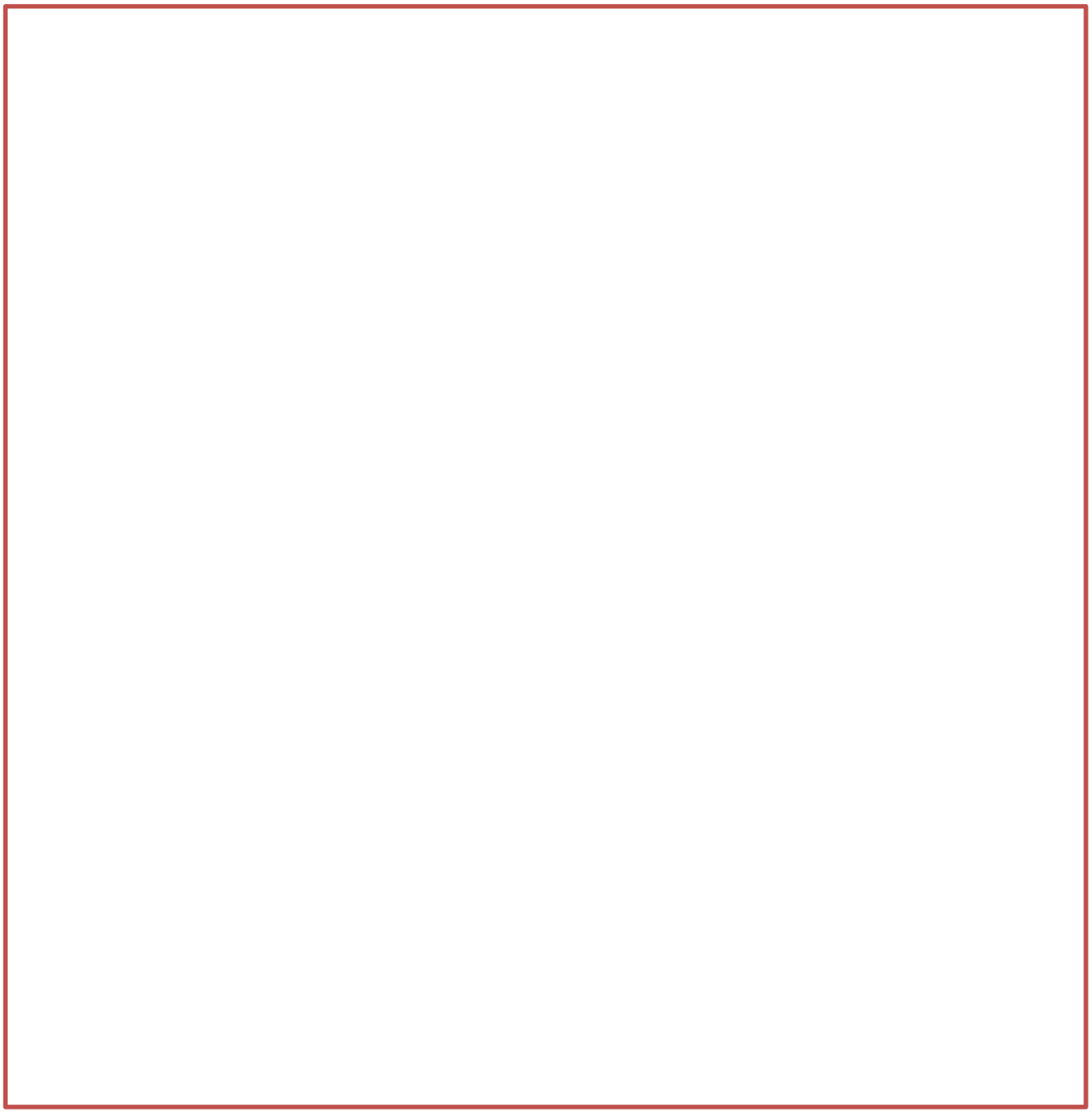


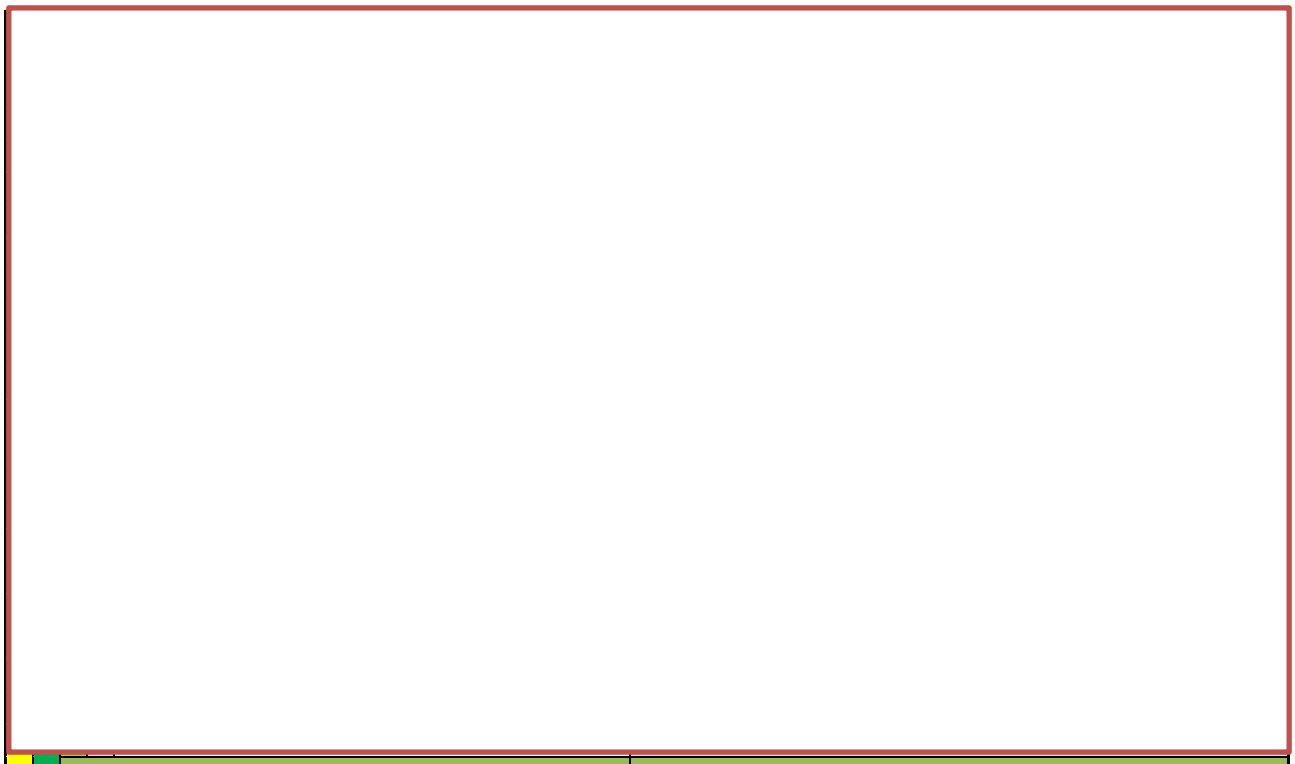
Figure 4-4 State Transition of Application

4.7. File Configuration

Table 4-6 shows the file configuration of the basic demonstration software. Described the files needed to project generation. For the file configuration of the FIT module, refer to the appropriate document given in section 1.1, Related Documents.

Table 4-6 File Configuration

Folder/File name	Description
RX231kit	Project folder
.cproject	**e² studio project**
.HardwareDebuglinker	**e² studio project**
.info	**e² studio project**
.project	**e² studio project**
makefile.init	**e² studio project**
.settings	e² studio setting folder
APL	Application folder
demo_src	Demonstration source folder
demo_fw.c	**Demonstration firmware**
leddriver.c	**LED driver**
main.c	**Demonstration main function**
main_apl.c	**Demonstration application**
inc	Include folder
demo_apl.h	**Demonstration firmware / application header**
leddriver.h	**LED driver header**
FIT	FIT folder
r_bsp	bsp FIT module set
r_cmt_rx	cmt FIT module set
r_config	FIT config file set
r_irq_rx	irq FIT module set
r_sci_iic	siic FIT module set
MW	Middleware folder



touch	Touch key control software folder
r_touch_API.c	**Touch key control API**
CTSU	Touch key control software
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	
walk	Walk system control software folder
App_PDM_Main.c	**Source**
App_PDM_Main.h	**Header**

4.8. Sensor Control

4.8.1. Pyroelectric Infrared Sensor (IRA-S210ST01)

Figure 4-5 dapl_mortion_detect() Flowchart

4.8.2. Acceleration Sensor (KXSS5-2057)

5. Lists of Constants

Table 5-1 lists the constants defined by the firmware layer. These constants are defined with demo_apl.h. See the appropriate documents given in section 1.1, Related Documents, for the constants in the FIT modules and sample programs.

Table 5-1 Firmware Layer Constants

Constant Name	Setting	Description
INPUT_KEY_DATA	PORTD.PIDR.BIT.B5	Port name definition
PORT_LOW	0	Port name definition
PORT_HIGH	1	Port name definition
PORT_OUT_MODE	1	Port name definition
PORT_IN_MODE	0	Port name definition
CMT0_FREQ_HZ	50	cmt frequency setting (20 ms)
CMT1_FREQ_HZ	1000	cmt frequency setting (1 ms)
SLAVE_ADDR_LCD	0x7C>>1	LCD device address
ACCESS_ADDR_LCD	0	-
ST7032I_RS0	0x00	LCD command
ST7032I_RS1	0x40	LCD command
ST7032I_CO	0x80	LCD command
LCD_1LINE_START_ADDRESS	0x80	LCD display start address
LCD_2LINE_START_ADDRESS	0xC0	LCD display start address
LCD_BUFFER_SIZE	0xC0	Simple I^2C transfer buffer
LCD_BACKRIGHT_FUNCTION	PORTB.PDR.BIT.B3	LCD backlight
LCD_BACKRIGHT_PORT	PORTB.PODR.BIT.B3	LCD backlight
LCD_RESET_FUNCTION	PORTB.PDR.BIT.B2	LCD reset port
LCD_RESET_PORT(a)	PORTB.PODR.BIT.B2 = a	LCD reset port
LCD_RESET_ON	0	-
LCD_RESET_OFF	1	-
LCD_RESET_WAIT	2	-

Table 5-2 lists the constants defined by the application layer. These constants are defined with demo_apl.h.

Table 5-2 Application Layer Constants

Constant Name	Setting	Description
SOFTWARE_VERSION	X.XX	Software version
LCD_1LINE	0	LCD display line specification
LCD_2LINE	1	LCD display line specification
LCD_SIZE	32	LCD display size
TOUCH_DETECT	1	Touch key detection
SENSOR_DETECT	1	Sensor detection
NO_DETECT	0	Not detected

6. Types and Structures

The variable type is defined as follows:

```
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;
```

The enum type is defined as follows:

```
/* Touch key event */
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;
```

```
/* Application structure */
typedef struct
{
    TOUCH_EVENT     touch_action;
    uint8_t         touch_action_flag;
    uint16_t        led1_count;
    uint16_t        led2_count;
} dapl_param_t;
```

Table 6-1 lists the application layer structure dapl_param_t, which is defined with demo_apl.h.
See the appropriate documents given in section 1.1, Related Documents, for the structures in the FIT modules and sample programs.

Table 6-1 Application Layer Structure dapl_param_t

Data Type	Member	Description
TOUCH_EVENT	touch_action	Touch key event
uint8_t	touch_action_flag	Touch key detection flag
uint16_t	led1_count	LED1 lighting time counter
uint16_t	led2_count	LED2 lighting time counter

7. Lists of Global Variables

Table 7-1 and Table 7-2 list the global variables defined by the firmware and application layers, respectively. The global variables defined in the FIT modules and sample programs are not shown.

Table 7-1 Firmware Layer Global Variables

Data Type	Variable Name	Description
sci_iic_info_t	g_iic_info_lcd	LCD communication structure
uint8_t	g_slave_addr_lcd[1]	LCD device address storage buffer
uint8_t	g_access_addr_lcd[1]	LCD data buffer
uint8_t	g_lcd_init[10]	LCD initial settings

Table 7-2 Application Layer Global Variables

Data Type	Variable Name	Description
dapl_param_t	g_dapl_param	Structure variable for application management

8. List of API Functions

Table 8-1 and Table 8-2 list the API functions.

Table 8-1 API Functions (1/2)

Function	Description
MCU (RX231)	
void R_dfw_wait_us(uint16_t time)	Software wait [us]
void R_dfw_wait_ms(uint16_t time)	Software wait [ms]
void R_LED_Open(void)	Initializes LED
void R_dfw_led1_on(uint16_t cnt)	Turns on LED1
void R_dfw_led1_off(void)	Turns off LED1
void R_dfw_led2_on(uint16_t cnt)	Turns on LED2
void R_dfw_led2_off(void)	Turns off LED2
uint8_t R_dfw_get_key_status(void)	Acquires hardware key state
void R_dfw_port_init(void)	Initializes ports
void R_dfw_irq_init(void)	Initializes IRQ
void R_dfw_cmt_init(void)	Initializes CMT
void R_dfw_siic6_init(void)	Initializes simple I^2C
void R_dfw_apl_init(void)	Initializes peripheral functions
LCD control (NHD-C0216CiZ-FSW-FBW-3V3)	
void R_dfw_lcd_init(void)	Initializes LCD
void R_dfw_lcd_on(void)	Turns on LCD backlight
void R_dfw_lcd_off(void)	Turns off LCD backlight
void R_dfw_lcd_write_str(uint8_tl uint8_t c, uint8_t len, uint8_t *str)	Displays LCD character strings
SAIC101 control (RAA7301013CBG)	
Acceleration sensor control (KXSS5-2057)	

Table 8-2 API Functions (2/2)

Function	Description
Serial Flash (MX25L6433FM2I-08G)	
SOUND	
Touch	
void R_dfw_touch_init(void)	Initializes touch key
uint8_t dapl_touch_detect(void)	Detects touch key
Walk	

9. Descriptions for API Functions

This section describes in detail the API functions shown in section 8.

9.1. MCU Functions

9.1.1. R_dfw_wait_us

Format	void R_dfw_wait_us(uint16_t time)		
Parameter	uint16_t	Time	Wait time(*1us)
Return value	void	—	
Specification	Generates a software wait according to the set parameter. For example, when 500 is set, 500 μs of wait is inserted.		
Note	This function is prepared for the Renesas RXC Toolchain V2.04.01 with 32-MHz main clock. When the development environment or main clock is changed, adjust the number of instructions by modifying the NOP count, etc.		

9.1.2. R_dfw_wait_ms

Format	void R_dfw_wait_ms(uint16_t time)		
Parameter	uint16_t	Time	Wait time(*1ms)
Return value	void	—	
Specification	Generates a software wait according to the set parameter. For example, when 500 is set, 500 ms of wait is inserted.		
Note	This function is prepared for the Renesas RXC Toolchain V2.04.01 with 32-MHz main clock. When the development environment or main clock is changed, adjust the number of instructions by modifying the NOP count, etc.		

9.1.3. R_LED_Open

Format	void R_LED_Open(void)		
Parameter	void	—	—
Return value	void	—	
Specification	Makes the initial setting of ports LED1 and LED2.		

9.1.4. R_dfw_led1_on

Format	void R_dfw_led1_on(uint16_t cnt)		
Parameter	uint16_t	cnt	0>: Lighting time [ms] 0 : Continuous lighting
Return value	void	—	
Specification	Turns on LED1 (orange). The lighting time can be specified with the parameter cnt in the ms units. When cnt is 0, LED1 is lit continuously. *Set “time to be specified + 1” for the parameter cnt. *At most +1 ms of error may occur.		

9.1.5. R_dfw_led1_off

Format	void R_dfw_led1_off(void)		
Parameter	void	—	—
Return value	void	—	
Specification	Turns off LED1 (orange).		

9.1.6. R_dfw_led2_on

Format	void R_dfw_led2_on(uint16_t cnt)		
Parameter	uint16_t	cnt	0>: Lighting time [ms] 0 : Continuous lighting
Return value	void	—	
Specification	Turns on LED2 (green). The lighting time can be specified with the parameter cnt in the ms units. When cnt is 0, LED2 is lit continuously. *Set “time to be specified + 1” for the parameter cnt. *At most +1 ms of error may occur.		

9.1.7. R_dfw_led2_off

Format	void R_dfw_led2_off(void)		
Parameter	void	—	—
Return value	void	—	
Specification	Turns off LED2 (green).		

9.1.8. R_dfw_get_key_status

Format	uint8_t R_dfw_get_key_status(void)		
Parameter	void	—	—
Return value	uint8_t	0: Pressed 1: Not pressed	
Specification	Returns the hardware key status.		

9.1.9. R_dfw_port_init

Format	void R_dfw_port_init(void)		
Parameter	void	—	—
Return value	void	—	
Specification	Makes the initial setting of the ports. For the used ports, the initial values are set in the pertinent FIT module. /* 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

9.1.11. R_dfw_irq_init

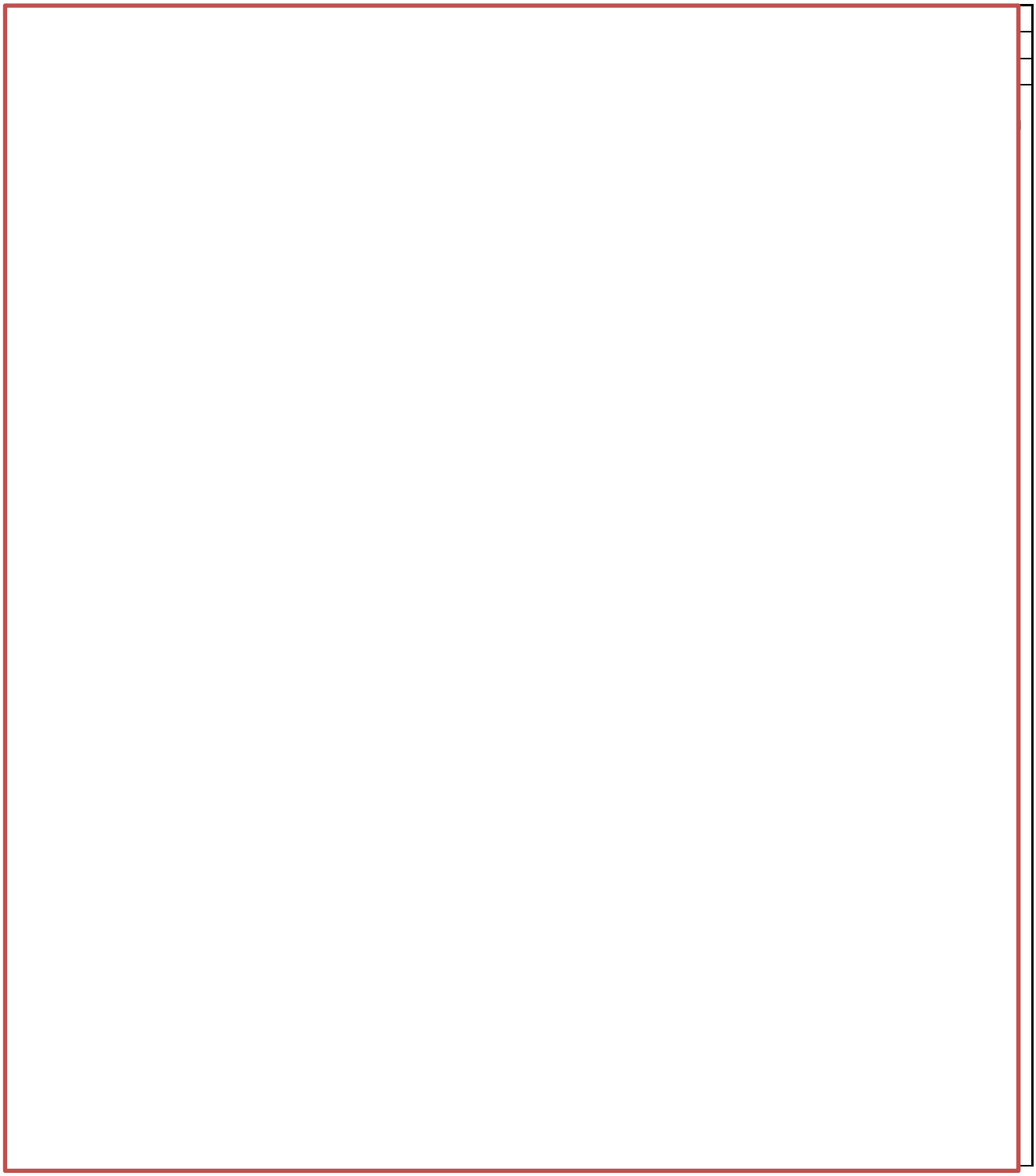
Format	void R_dfw_irq_init(void)		
Parameter	void	—	—
Return value	void	—	
Specification	Makes the initial setting of the IRQ module. The initial values are set with R_IRQ_Open() of the IRQ FIT. IRQ0: Rising edge, priority level 2, interrupt handler dapl_intirq0_handler registered		

```
irq_handle_t irq_handle_dummy;

/* IRQ port setteing */
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

Format	void R_dfw_cmt_init(void)		
Parameter	void	—	—
Return value	void	—	
Specification	Makes the initial setting of the CMT module. The initial values of cmt0 and cmt1 are set with R_CMT_CreatePeriodic() of CMT FIT. cmt0: • Interrupt cycle period: 20ms • Interrupt handler dapl_intcmt0_handler registered cmt1: • Interrupt cycle period: 1ms • Interrupt handler dapl_intcmt1_handler registered /* 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

Format	void R_dfw_siic6_init(void)		
Parameter	void	—	—
Return value	void	—	

Specification	Makes the initial setting of the simple I²C module.
	The initial values are set with R_SCI_IIC_Open() of SCI IIC FIT.
	
	• Simple I²C6 channel setting
	• Interrupt handler dapl_intiic_handler registered
	
	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

Format	void R_dfw_apl_init(void)		
Parameter	void	—	—
Return value	Void	—	
Specification	Makes the initial setting of the peripheral modules (LDC, SAIC101, KXSS5 (acceleration sensor)) • R_lcd_init() call • R_dfw_saic_init() call • R_dfw_kxss5_init() call Sets the initial values for demonstration application. • Walk data check • Recorded data check		
Note	A 500-ms software wait is included.		

9.2. LCD Control Functions

9.2.1. R_dfw_lcd_init

Format	void R_dfw_lcd_init(void)		
Parameter	void	—	—
Return value	void	—	
Specification	Makes the initial setting of the LCD. • LCD reset port setting • LCD initial setting • 8-bit mode, 2-line display mode • OSC frequency 347kHz • Contrast 0x70 set • Display turned on		
Note	Call this function before using the functions described in sections 9.2.2 to 9.2.4. A 40-ms software wait is included.		

9.2.2. R_dfw_lcd_on

Format	void R_dfw_lcd_on(void)		
Parameter	void	—	—
Return value	void	—	
Specification	Turns on the LCD backlight.		

9.2.3. R_dfw_lcd_off

Format	void R_dfw_lcd_off(void)		
Parameter	void	—	—
Return value	void	—	
Specification	Turns off the LCD backlight.		

9.2.4. R_dfw_lcd_write_str

Format	void R_dfw_lcd_write_str(uint8_t l, uint8_t c, uint8_t len, uint8_t * str)		
Parameter	uint8_t	l	LCD display line
	uint8_t	c	Display start column
	uint8_t	len	Number of characters displayed
	uint8_t *	str	Head pointer of the displayed character string
Return value	void	—	
Specification	Displays a character string specified by the parameter str on the LCD. The display start line, start column, and number of characters displayed are set with the parameters l, c, and len, respectively.		
Note	Set the parameters c and len so that the total of them does not exceed 16 (LCD display size).		

9.3. SAIC101 Control Functions

9.4. kxss5 (Acceleration Sensor) Control Functions

9.5. Serial Flash Control Functions

9.6. Sound Playback/Recording Control Functions

9.7. Touch Key Control Functions

9.7.1. R_dfw_touch_init

Format	void R_dfw_touch_init(void)		
Parameter	void	—	—
Return value	void	—	
Specification	Makes the initial setting of the touch key.		
Note	Call this function before using the function described in section 9.7.2.		

9.7.2. dapl_touch_detect

Format	uint8_t dapl_touch_detect(void)		
Parameter	void	—	—
Return value	uint8_t	0: Not detected 1: Detected	
Specification	Checks that the touch key has been detected or not. Call this function in a specific time interval. Touch key detection information is stored in g_touch_all_result. Capacitive touch key is detected by the self-capacitance method. For details, see the Capacitive touch sensor system, given as the related document.		

9.8. Walk System Control Functions

10. Notes

The USB IDs (vendor ID, product ID) installed in the basic demonstration software cannot be used on the products for the customers.

Please acquire a vendor ID from the USB-IF.

http://www.usb.org/home

Website and Support

Renesas Electronics Website

http://www.renesas.com/

Inquiries

http://www.renesas.com/contact/

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2015.10.19	—	First Edition issued
1.10	Jun 24, 2016	4 16 28	Modified ‘Table 2-1 Operation Confirmation Conditions’. Modified ‘Table 4-6 File Configuration’. Modified ‘9.1.1 R_dfw_wait_us’. Modified ‘9.1.2 R_dfw_wait_ms’.
1.20	2020.8.20	—	Update the toolchain version

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.