RA2L2 Group

USB Type-C Reference Design for RA2L2 MCUs

Introduction

This application note mainly covers the following contents.

- General overview of the USB Type-C reference design for RA2L2 MCUs
- Hardware and software specifications related to USB Type-C configuration channel (CC) detection by this reference design
- Operating specifications for demonstrations targeting small-size battery-powered applications

Devices and Hardware Used in Confirming Operation

- Board: RA2L2 USB-C Demonstration Board
- Products:
  - MCU: Renesas RA2L2 (R7FA2L2093CFM)
  - Charger IC: Renesas DA9168
  - Temperature and humidity sensor: Renesas HS4001*
  - Motion sensor: TDK ICM-42688-P

Software

- Integrated development environment: e^2 studio 2025-04.1
- C compiler: GCC Arm Embedded 13.2.1.arm-13-7
- FSP (Flexible Software Package): v6.0.0

Related Documents

- RA2L2 MCUs USB Type-C Reference Design User's Manual: R12UZ0203
- RA2L2 Group User's Manual: Hardware: R01UH1080
- Renesas Flexible Software Package (FSP) V6.0.0 User's Manual: R11UM0155EU0500
- QE for USB Usage Guide: R20AN0143

Related Links

- Web page for USB Type-C Reference Design for RA2L2 MCUs
- Web page for RA2L2 Products
- Example of USB Application: USB Data Logger
- Web page introducing QE for USB

Note. Renesas has announced the discontinuation of the HS4001."MEMS Vision #MVH4001D" is scheduled to be the replacement.
PLC250010: End-of-Life (EOL) process for selected part numbers

Contents

1. Overview of the USB Type-C Reference Design for RA2L2 MCUs

1.1 What is the USB Type-C Reference Design for RA2L2 MCUs?

The USB Type-C reference design for RA2L2 MCUs is a solution intended for small-size battery-powered applications with the USB Type-C interface, such as USB data loggers, True Wireless Stereo (TWS) charging cases, and PC peripheral devices. In this reference design, the RA2L2 32-bit MCU with the USB Type-C CC detection function is used to implement USB Type-C detection operations presupposing use of the functionality in real applications.

Running each of the demonstrations allows users to easily experience the main features of the RA2L2, such as USB 2.0 Full-Speed (FS) communications, various types of serial communications, and low-power modes, together with the USB Type-C CC detection operation.

1.2 Features of the USB Type-C Reference Design for RA2L2 MCUs

- Uses the RA2L2, which is the industry's first 32-bit MCU to support Release 2.4 of the USB Type-C standard and USB 2.0 Full-Speed (FS) communications.
- System configuration containing the USB Type-C interface on the assumption of small-size battery-powered applications
  - Appropriate for USB data loggers, True Wireless Stereo (TWS) charging cases, PC peripheral devices, etc.
  - A small-size battery-driven single board equipped with a charger IC, sensor device, and LCD
- USB Type-C CC detection operation by using the RA2L2's built-in USB Type-C interface
- Supports three types of demo use cases utilizing the main features of the RA2L2.
  - Three types of demo use cases: Stand-alone (running from a battery), connected to a charger, and connected to a PC
  - USB Type-C CC detection, USB 2.0 Full-Speed (FS) communications, various other types of connectivity, and low-power modes
- PCB design files and a variety of technical information including sample code are provided.

1.3 Configuration of the USB Type-C Reference Design for RA2L2 MCUs

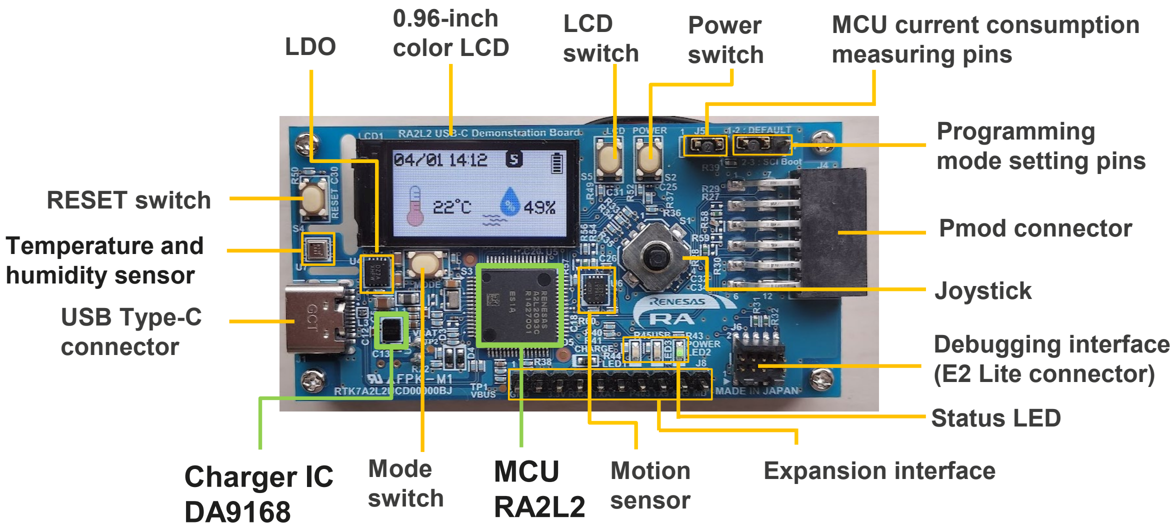
This reference design consists of the items listed in Table 1-1.

Table 1-1 Configuration of the Reference Design

No.	Item	Description
1	RA2L2 USB-C Demonstration Board (hereafter referred to as the demo board)	Single demo board with a mounted RA2L2. This board is not for sale. If you would like to use it, contact our sales team or click on the following link to the point for contact. Contact Us | Renesas
2	PCB design files	PCB design data of the demo board, including circuit diagrams, BOM, Gerber data, and artwork files. They are available on the Web page for the reference design.
3	Sample code	Sample code for the reference design. It is available on the Web page for the reference design.
4	Application note	USB Type-C Reference Design for RA2L2 MCUs Application Note (this document)

1.4 Outward Appearance and Components of the Demo Board

Figure 1-1 shows the outward appearance of the demo board and describes its components.

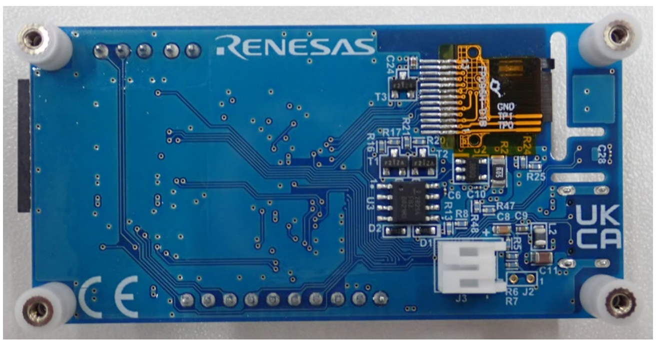


Power LED (green) ON: MCU is operating normally,
OFF: MCU is in software standby mode or operation is stopped

USB LED (yellow) ON: USB power is being supplied,
OFF: USB power is not being supplied

Charging LED (red) ON: Charging is in progress, OFF: Charging is stopped

Back of the board (after the battery is installed)



Back of the board (before the battery is installed)

Figure 1-1 Outward Appearance of the Demo Board

1.5 Outline of Hardware Specifications

Table 1-2 lists the hardware specifications of this demo board in outline.

Table 1-2 Outline of Hardware Specifications

Item	Specifications
Board part number	RTK7A2L2UCD00000BJ
MCU	RA2L2: R7FA2L2093CFM (Code flash memory: 128 Kbytes, RAM: 16 Kbytes, Data flash memory: 4 Kbytes, PKG: 64-pin QFP)
Clocks	High-speed on-chip oscillator (48-MHz operation): System clock and peripheral module clocks 32.768-kHz crystal resonator: RTC clock
Sensors	Temperature and humidity sensor (Renesas: HS4001) × 1*1 Motion sensor (TDK: ICM-42688-P) × 1
Switches	Joystick (4-way + center press) × 1 Power switch, mode switch, LCD switch, and reset switch
Status LEDs	Power LED, charging LED, and USB LED
Display	0.96-inch color LCD
Interfaces	USB Type-C™ × 1 Pmod™ interface × 1 Debugging interface (E2 emulator Lite (E2 Lite) connector) × 1 Expansion interface × 1
Charger IC	DA9168 (1-cell battery charger IC)
Power supply	1-cell lithium-ion battery (Data Power Technology Ltd.: DTP502535 (400 mAh)) USB bus power
External dimensions	35 × 70 mm

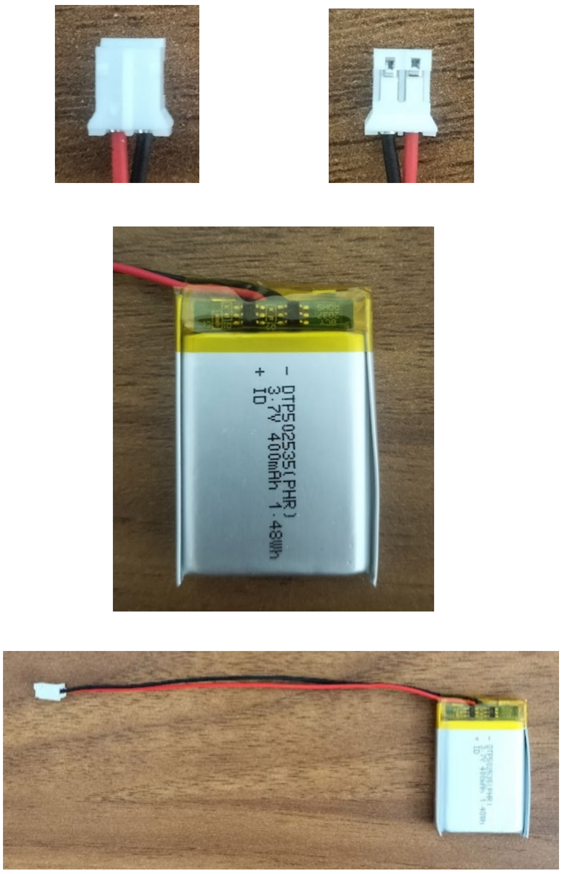
Note 1. Renesas has announced the discontinuation of the HS4001. “MEMS Vision #MVH4001D” is scheduled to be the replacement.

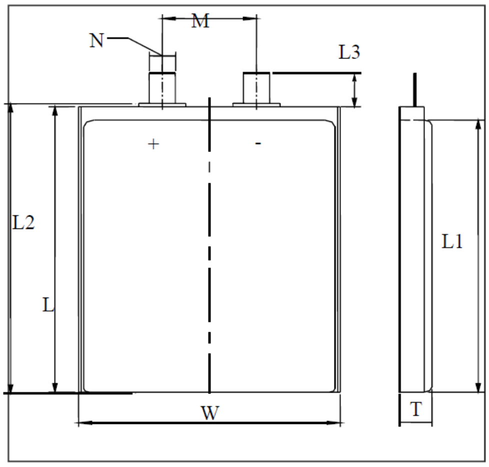
1.6 Outline of Battery Specifications

A 1-cell lithium-ion battery (part number: DTP502535) made by Data Power Technology Ltd. is used on this demo board. Table 1-3 lists the specifications of the battery in outline. Figure 1-2 shows the outward appearance of the battery.

Table 1-3 Outline of Battery Specifications

Item	Specifications
Battery type	Rechargeable lithium-ion polymer battery
Rated capacity	400 mAh
Normal voltage	3.70 V
Charge Limited voltage	4.20 V
Discharge cut-off voltage	2.80 V
Maximum continuous charge current	1C (400 mA)
Maximum continuous discharge current	1C (400 mA)
Operating temperature range	Charge: 0 to 45°C, Discharge: –20 to 60°C
Storage temperature range	–20 to 60°C
Operating and storage humidity range	65 ± 20% RH





Item	Dimension（mm）
T	Max 5.0
W	Max 25.0
L	Max 35.0
L1	Max 31.0
L2	Max 35.3
L3	6.0±2.0
M	12.0±2.0
N	2.0±0.5

Figure 1-2 Outward Appearance of the Battery

1.7 Outline of RA2L2 Specifications

Table 1-4 lists the specifications of the MCU (RA2L2) used on this demo board in outline. The part number of the MCU used on this demo board is the R7FA2L2093CFM (operating temperature: –40°C to +105°C) variant of the R7FA2L209xCFM products.

Table 1-4 Outline of RA2L2 Specifications

Parts number		R7FA2L209xCFM	R7FA2L207xCFM	R7FA2L209xCFL R7FA2L209xCNE	R7FA2L207xCFL R7FA2L207xCNE	R7FA2L209xCFJ R7FA2L209xCNH	R7FA2L207xCFJ R7FA2L207xCNH
Pin count		64		48		32	
Package		LQFP		LQFP/HWQFN		LQFP/HWQFN	
Code flash memory		128 KB	64 KB	128 KB	64 KB	128 KB	64 KB
Data flash memory		4 KB		4 KB		4 KB	
SRAM(Parity)		16 KB		16 KB		16 KB	
System	CPU clock	48 MHz		48 MHz		48 MHz	
	Sub clock oscillator	Yes		Yes		Yes	
	ICU	Yes		Yes		Yes	
	KINT	8		5		4	
Event control	ELC	Yes		Yes		Yes	
DMA	DTC	Yes		Yes		Yes	
Timers	GPT32	1 (PWM outputs: 2)		1 (PWM outputs: 2)		1 (PWM outputs: 2)	
	GPT16	6 (PWM outputs: 12)		6 (PWM outputs: 12)		6 (PWM outputs: 7)	
	AGTW	2		2		2	
	RTC	Yes		Yes		Yes	
	WDT/IWDT	Yes		Yes		Yes	
Communication	SCI	4		4		3	
	I3C	1		1		1	
	SPI	1		1		1	
	CAN	1		1		1	
	SSIE	1		1		1	
	UARTA	2		2		2	
	USBFS	1		1		1	
	USBCC	1		1		1	
Analog	ADC12	17		13		10	
	TSN	Yes		Yes		Yes	
Data processing	CRC	Yes		Yes		Yes	
	DOC	Yes		Yes		Yes	
Security		TRNG		TRNG		TRNG	
I/O ports	I/O pins	51		35		21	
	Input pins	3		3		3	
	Pull-up resistors	51		35		21	
	N-ch open-drain outputs	38		24		13	
	5-V tolerance	7		7		5	

1.8 System Block Diagram

Figure 1-3 shows a system block diagram of this demo board.

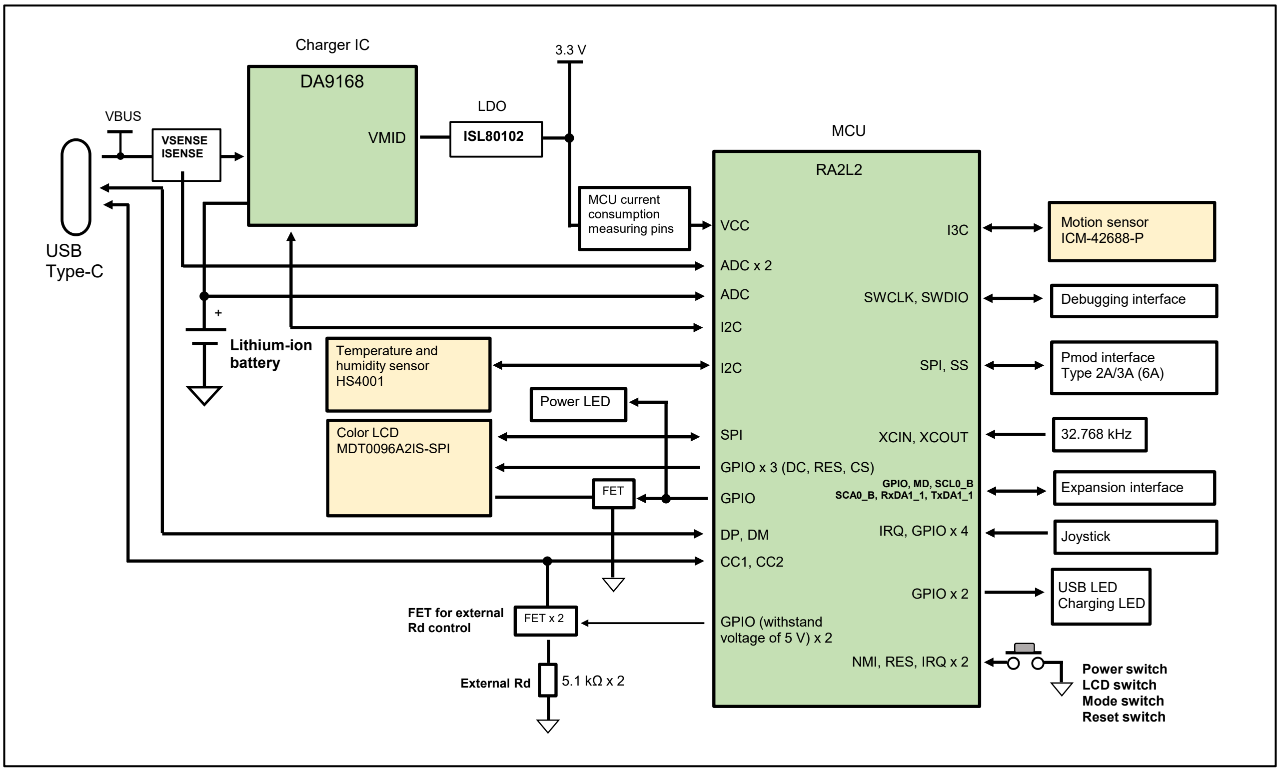


Figure 1-3 System Block Diagram

1.9 List of MCU Pin Assignments and Pin Settings

Table 1-5 lists the MCU pin assignments and pin settings of this reference design.

Table 1-5 MCU Pin Assignments and Pin Settings

Pin No.	Pin Name	Signal	Pin Setting Mode	Internal Pull-Up/Pull-Down Setting	Active Level	Function
1	P400	GPIO	Output mode (initial high output)	None	Low	External Rd control for CC1 (OD output)
2	P401	GPIO	Output mode (initial high output)	None	Low	External Rd control for CC2 (OD output)
3	P402	GPIO	Input mode	None	Low	LCD display switching button
4	P403	GPIO	Input mode	Input pull-up	—	Free (connected to the expansion interface connector)
5	VCL	—	—	—	—	4.7-uF capacitor
6	P215	XCIN	Peripheral mode	—	—	Sub-clock
7	P214	XCOUT	Peripheral mode	—	—	Sub-clock
8	VSS	—	—	—	—	VSS
9	P213	SDA1_A	Peripheral mode	None	—	Charger IC - SDA
10	P212	SCL1_A	Peripheral mode	None	—	Charger IC - SCL
11	VCC	—	—	—	—	VCC
12	P411	GPIO	Output mode (initial high output)	None	Low	Charger IC - CHG_EN
13	P410	IRQ5	IRQ mode	None	Low	Charger IC - INT
14	P409	GPIO	Output mode (initial high output)	None	High	Charger IC – EN (spare)
15	P408	IRQ7	IRQ mode	None	Low	Motion sensor INT2 (or INT1)
16	USB_DM	USB_DM	—	—	—	USB communications
17	USB_DP	USB_DP	—	—	—	USB communications
18	P913	USB_CC1	Analog mode	None	—	USB Type-C CC detection (CC1)
19	P912	USB_CC2	Analog mode	None	—	USB Type-C CC detection (CC2)
20	P407	USB_VBUS	Peripheral mode	None	—	VBUS detection
21	P207	GPIO	Input mode	Input pull-up	—	Pmod (#1) - CS/CTS/-
22	P206	MISO0_D RXD0_D SCL0_D	Input mode	Input pull-up	—	Pmod (#3) - MISO/RXD/SCL
23	P205	MOSI0_D TXD0_D SDA0_D	Input mode	Input pull-up	—	Pmod (#2) - MOSI/TXD/- Pmod (#4) - SDA
24	P204	SCK0_D	Input mode	Input pull-up	—	Pmod (#4) - SCK/RTS
25	RES	—	—	—	—	Debugger (#10) - SWD/JTAG Reset switch Charger IC - RIN_N
26	P201	MD	Peripheral mode	None	—	Debugger (#4) - SWD/JTAG (connected to the expansion interface connector [for SCI boot])
27	P200	GPIO	Input mode	None	—	—
28	P304	GPIO	Output mode (initial low output)	None	High	USB LED
29	P303	GPIO	Output mode (initial low output)	None	High	Charge status LED
30	P302	SDA2_A	Peripheral mode	None	—	Temperature and humidity sensor – SDA
31	P301	SCL2_A	Peripheral mode	None	—	Temperature and humidity sensor – SCL
32	P300	SWCLK	Peripheral mode	None	—	Debugger (#4) - SWD/JTAG
33	P108	SWDIO	Peripheral mode	None	—	Debugger (#2) - SWD/JTAG
34	P109	SCL0_B	Peripheral mode	None	—	Motion sensor - SCL (connected to the expansion interface connector [for SCI boot/TXD9_B])

Pin No.	Pin Name	Signal	Pin Setting Mode	Internal Pull-Up/Pull-Down Setting	Active Level	Function
35	P110	SDA0_B	Peripheral mode	Input pull-up	—	Motion sensor - SDA (connected to the expansion interface connector [for SCI boot/RXD9_B])
36	P111	IRQ4	IRQ mode	None	Low	Motion sensor INT1 (or INT2)
37	P112	GPIO	Input mode	Input pull-up	—	Pmod (#10) - GPIO
38	P113	GPIO	Input mode	Input pull-up	—	Pmod (#9) - GPIO
39	VCC	—	—	—	—	VCC
40	VSS	—	—	—	—	VSS
41	P107	GPIO	Input mode	Input pull-up	—	Pmod (#8) - GPIO
42	P106	GPIO	Input mode	Input pull-up	—	Pmod (#7) - GPIO
43	P105	GPIO	Output mode (initial high output)	None	Low	Color LCD - RESET
44	P104	GPIO	Output mode (initial high output)	None	—	Color LCD - D/C
45	P103	SSLA0_A	Peripheral mode	None	—	Color LCD - CS
46	P102	RSPCKA_A	Peripheral mode	None	—	Color LCD – SCL (CLOCK)
47	P101	MOSIA_A	Peripheral mode	None	—	Color LCD – SDA (DI)
48	P100	MISOA_A	Peripheral mode	None	—	Color LCD – SDA (DO)
49	P500	GPIO	Output mode (initial low output)	None	High	Power LED Color LCD - Backlight
50	P501	GPIO	Output mode (initial low output)	None	—	Motion sensor - AD0
51	P502	GPIO	Input mode	None	Low	Joystick - Center
52	P015	RXDA1_A	Input mode	Input pull-up	—	Expansion interface
53	P014	TXDA1_A	Input mode	Input pull-up	—	Expansion interface
54	P013	GPIO	Input mode	None	Low	Joystick - A
55	P012	GPIO	Input mode	None	Low	Joystick - B
56	AVCC0	—	—	—	—	VCC
57	AVSS0	—	—	—	—	VSS
58	P011	GPIO	Input mode	None	Low	Joystick - C
59	P010	GPIO	Input mode	None	Low	Joystick - D
60	P004	GPIO	Input mode	None	Low	Mode switch
61	P003	AN003	Analog mode	None	—	Battery voltage
62	P002	AN002	Analog mode	None	—	VBUS monitoring - ISENSE
63	P001	AN001	Analog mode	None	—	VBUS monitoring - VSENSE
64	P000	IRQ6	IRQ mode	None	Low	Power switch

2. Specifications of USB Type-C Detection Operations in the Reference Design

This reference design uses the on-chip USB Type-C interface of the RA2L2 to configure hardware (the demo board) and software (sample code for the MCU) intended for a small-size battery-powered application equipped with a USB-Type C interface. This chapter describes the specifications of the RA2L2's built-in USB Type-C interface, the specifications of the operations of the USB Type-C hardware and software implemented in the reference design, and the control of charging when a USB Type-C source device is connected.

2.1 Specifications of the Standard of USB Type-C CC Detection

When devices are connected through a USB Type-C cable, the source device applies voltage to the Rp resistor in the source device and Rd resistor in the sink device and the voltage on the Rd resistor is measured to detect the default USB (0.5 A), 1.5-A source, or 3.0-A source current.

Table 2-1 Differences in the Threshold Voltage (V) on the Sink CC Pins

Source Current Detection	Power Supply	USB Type-C Cable and Connector Specifications — Threshold Voltage (V) on Sink CC Pins		RA2L2 CC Detection Circuit — Voltage Detection Threshold (V)
		Old Standard Release 2.3	New Standard Release 2.4	
Default USB	0.5 A @5 V	0.25	0.277	0.15
1.5-A source	1.5 A @5 V	0.66	0.613	0.613
3.0-A source	3.0 A @5 V	1.23	1.165	1.165

For example, if the sink device has detected the 3.0-A source current, a 15-W (3 A at 5 V) source device (such as a charger or a Type-C port of a PC) is connected to it. An example of the system operation after this detection is to send modified settings from the MCU to the charger IC or power-supply IC in the sink device through serial communications so that the charging current or receiving current is to be increased and fast charging or 15-W USB bus power operation is enabled.

2.2 Specifications of USB Type-C CC Detection Using the RA2L2

The USB Type-C interface module in the RA2L2 incorporates dedicated hardware circuits for detection that complies with the Universal Serial Bus Type-C Cable and Connector Specification Release 2.4 and supports the following functions.

- Detecting connection or disconnection of a USB port that complies with the USB Type-C Specification Release 2.4
- Providing an Rd resistor for each of the CC1 and CC2 lines and detecting the current supply capability of the source (default USB, 1.5-A source, or 3.0-A source current) by measuring the voltage on the Rd resistor

Figure 2-1 gives an overview of USB-Type-C connections with the RA2L2.

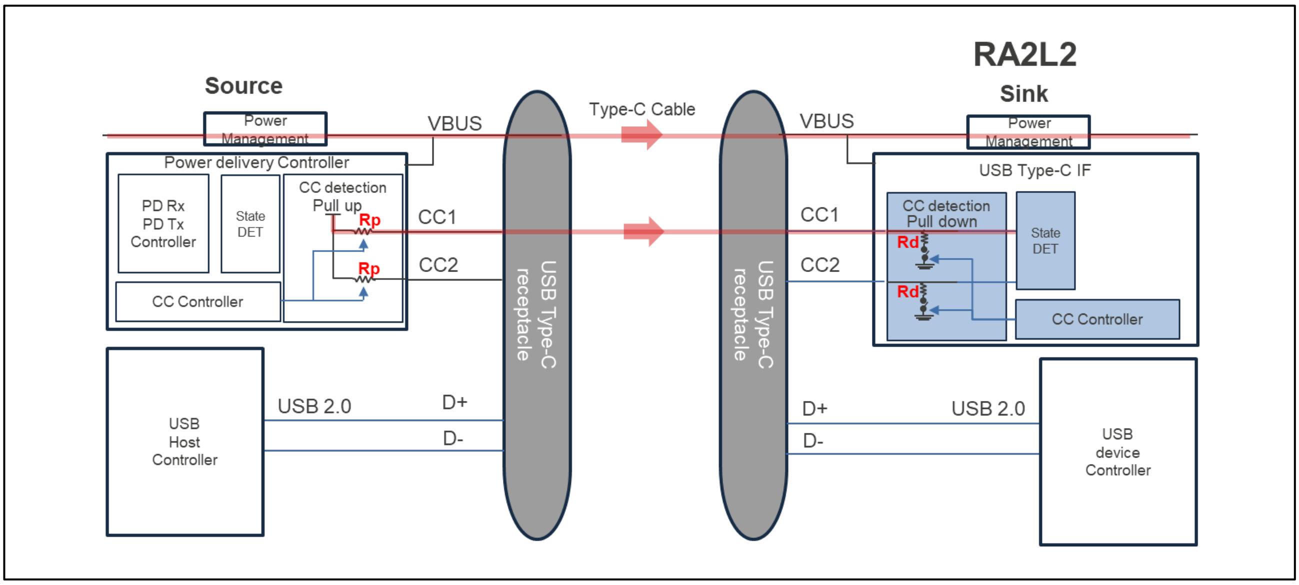


Figure 2-1 Overview of USB Type-C Connections Using the RA2L2

Figure 2-2 is a block diagram of the USB Type-C interface in the RA2L2.

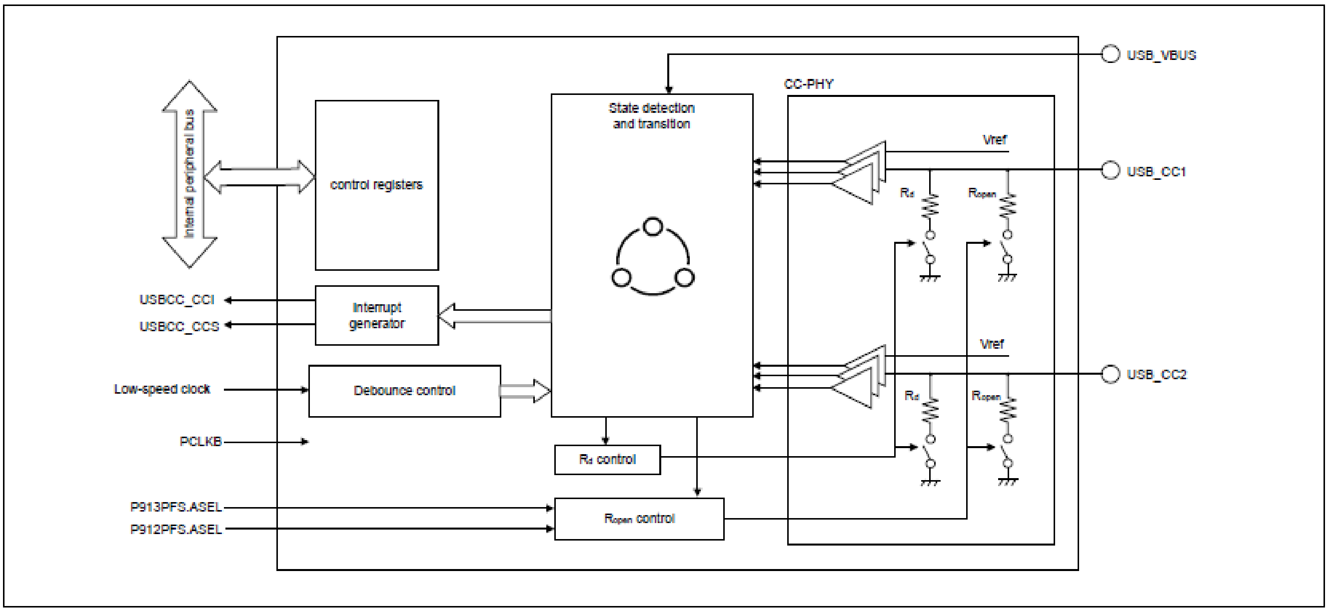


Figure 2-2 Block Diagram of the USB Type-C Interface in the RA2L2

The USB Type-C interface in the RA2L2 has a Type-CC connection state and status register (TCS register). The results of a USB Type-C detection operation can be read from the bits of this register. Table 2-2 lists the specifications of the TCS register. For details, refer to section 26.2.5, TCS: Type-CC Connection State and Status Register, in the RA2L2 Group User's Manual: Hardware (R01UH1080).

Table 2-2 Specifications of the TCS Register

Bit	Symbol	Function	R/W
0	SRCD	Detection of Source Connection 0: Source is not connected (connection status not in Attached.SNK) 1: Source is connected (connection status in Attached.SNK)	R
1	VRD15D	Detection of Power1.5 Source Connection 0: Power1.5 source is not connected 1: Power1.5 source is connected	R
2	VRD30D	Detection of Power3.0 Source Connection 0: Power3.0 source is not connected 1: Power3.0 Source is connected	R
3	PLUG	Connection of Plug Orientation 0: CC1 is connected 1: CC2 is connected	R*1
7:4	CNS[3:0]	Status of Connection State 0000: Disabled 0001: Unattached.SNK 0010: AttachWait.SNK 0100: Attached.SNK (PowerDefault.SNK) 0101: Attached.SNK (Power1.5.SNK) 0110: Attached.SNK (Power3.0.SNK) Others: Undefined	R
9:8	CC1S[1:0]	Status of CC1 00: Open (below maximum vRa) 01: Default USB (above minimum vRd-Connect) 10: Power1.5 (voltage above minimum vRd-Connect, and within the range of vRd-1.5) 11: Power3.0 (voltage above minimum vRd-Connect, and within the range of vRd-3.0)	R
11:10	CC2S[1:0]	Status of CC2 00: Open (below maximum vRa) 01: Default USB (above minimum vRd-Connect) 10: Power1.5 (voltage above minimum vRd-Connect, and within the range of vRd-1.5) 11: Power3.0 (voltage above minimum vRd-Connect, and within the range of vRd-3.0)	R
12	VBUSS	Status of VBUS 0: VBUS is off 1: VBUS is on	R
30:13	—	These bits are read as 0.	R

Note 1. This is valid only when the connection state is Attached.SNK in Sink-only mode (MEC.MODE=0).

2.3 Specifications of the USB Type-C Hardware in the Reference Design

Figure 2-3 shows the configuration of the circuits in the USB block, which includes the USB Type-C interface, in the reference design.

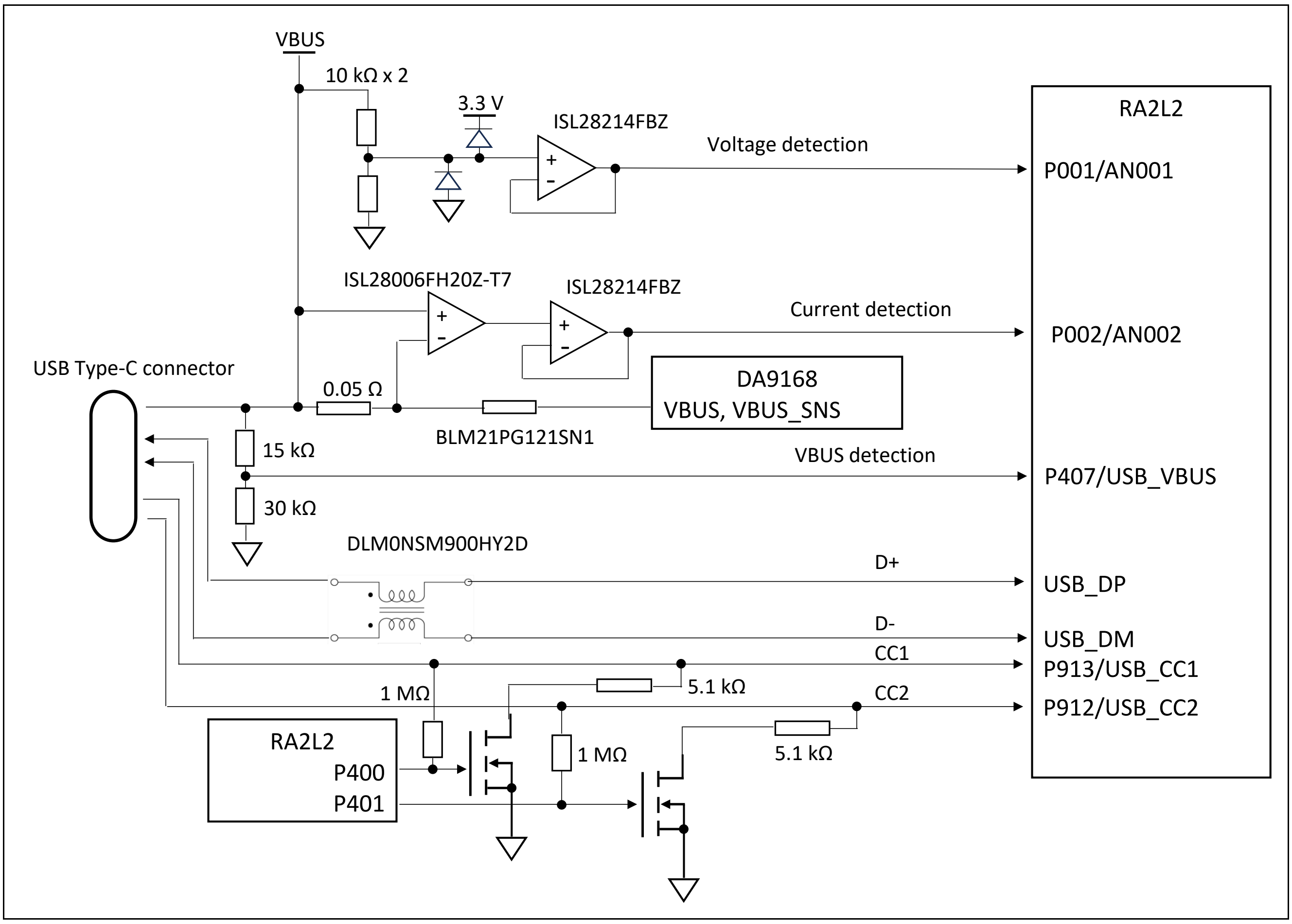


Figure 2-3 Configuration of the Circuits in the USB Block

2.4 Specifications of the Operations of the USB Type-C and USBFS Software in the Reference Design

The following describes the specifications of the operations of the USB Type-C and USBFS software in the reference design.

2.4.1 Processing Structure

Figure 2-4 shows the processing structure for controlling USB Type-C and USBFS connections.

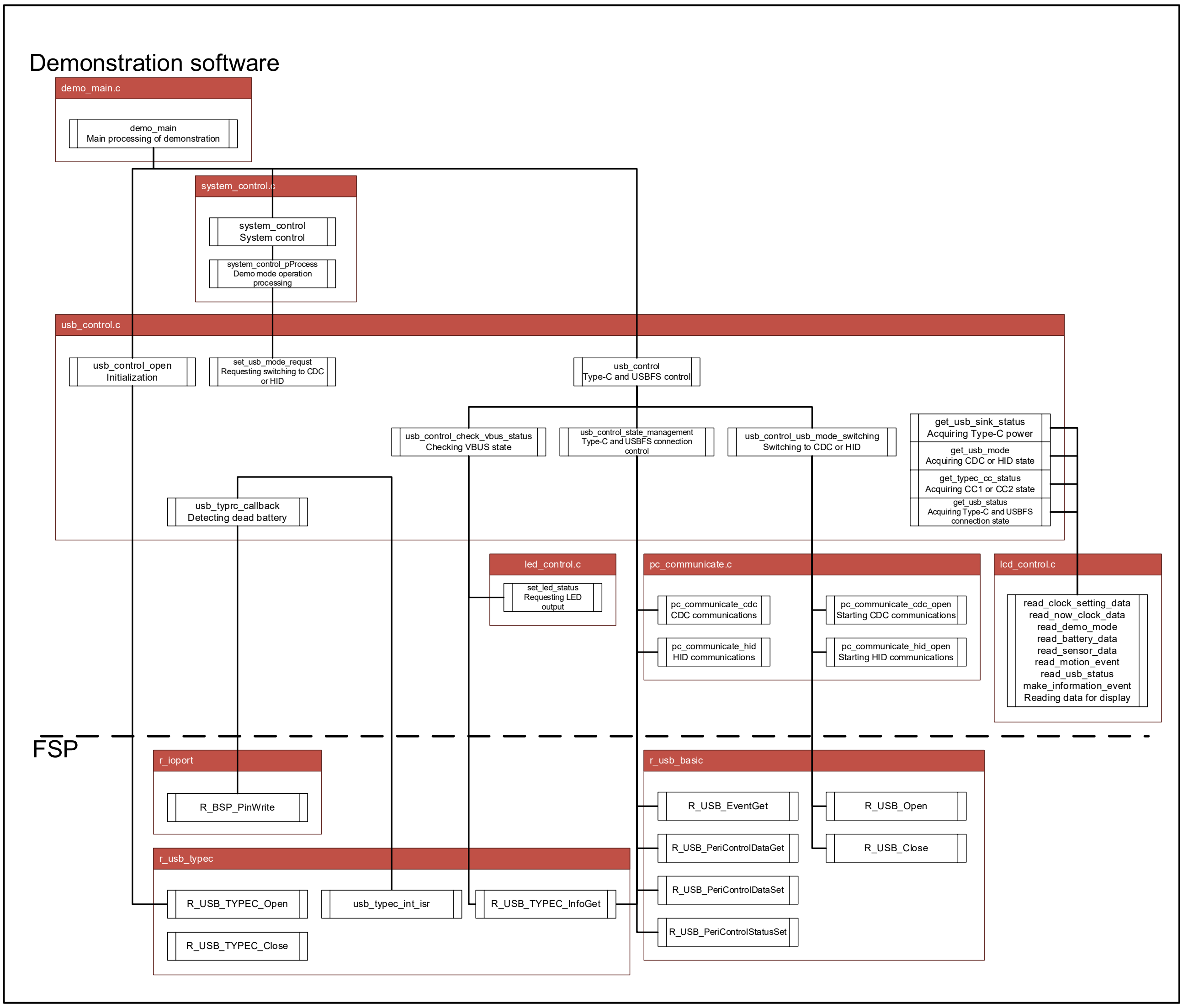


Figure 2-4 Processing Structure for Controlling USB Type-C and USBFS Connections

2.4.2 Connection State Control

Figure 2-5 is a state transition diagram of the control of the USB Type-C and USBFS connections.

The USB Type-C state is controlled by using the USB Type-C connection state and CDC or HID connection events. Switching to a CDC connection or an HID connection is done upon request from the system control process. This switching is independent of the USB Type-C connection state and the result of switching executed before the USB Type-C connection is made is reflected after that. The initial state in terms of CDC or HID connection is set for a CDC connection.

The USB-TYPEC (r_usb_typec) module of the FSP is used to detect a USB Type-C CC connection. For details of the USB-TYPEC (r_usb_typec) module, refer to section 7.2.6.45, USB TYPEC (r_usb_typec), in the Flexible Software Package (FSP) V6.0.0 Manual.

In addition, the USB (r_usb_basic) module of the FSP is used to handle the USBFS connection operation (enumeration). For details of the USB (r_usb_basic) module, refer to section 7.2.6.33, USB (r_usb_basic), in the FSP Manual.

- GitHub FSP Manual
  - FSP top page
  - USB-TYPEC (r_usb_typec) module
  - USB (r_usb_basic) module

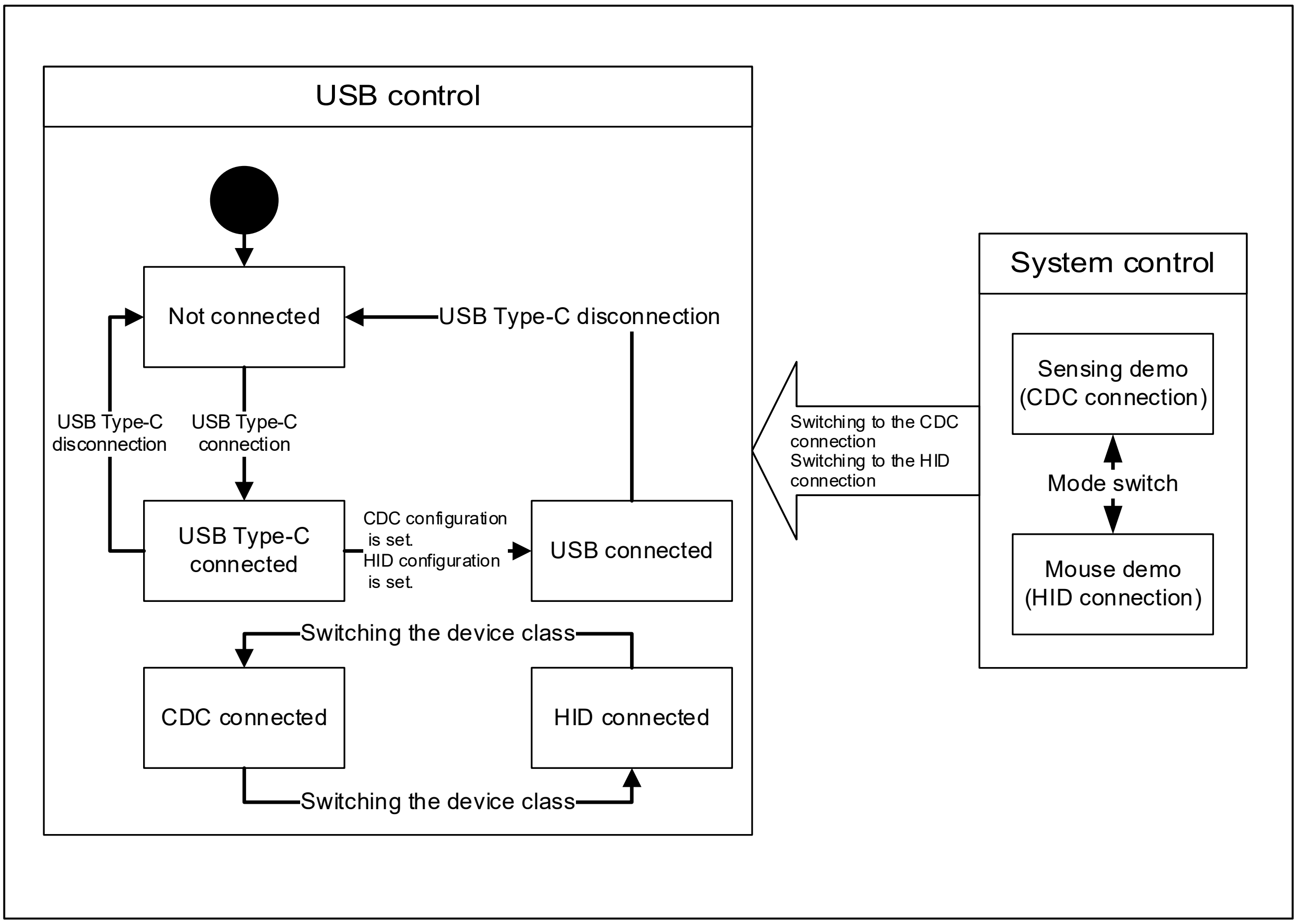


Figure 2-5 State Transitions for Controlling USB Type-C and USBFS Connections

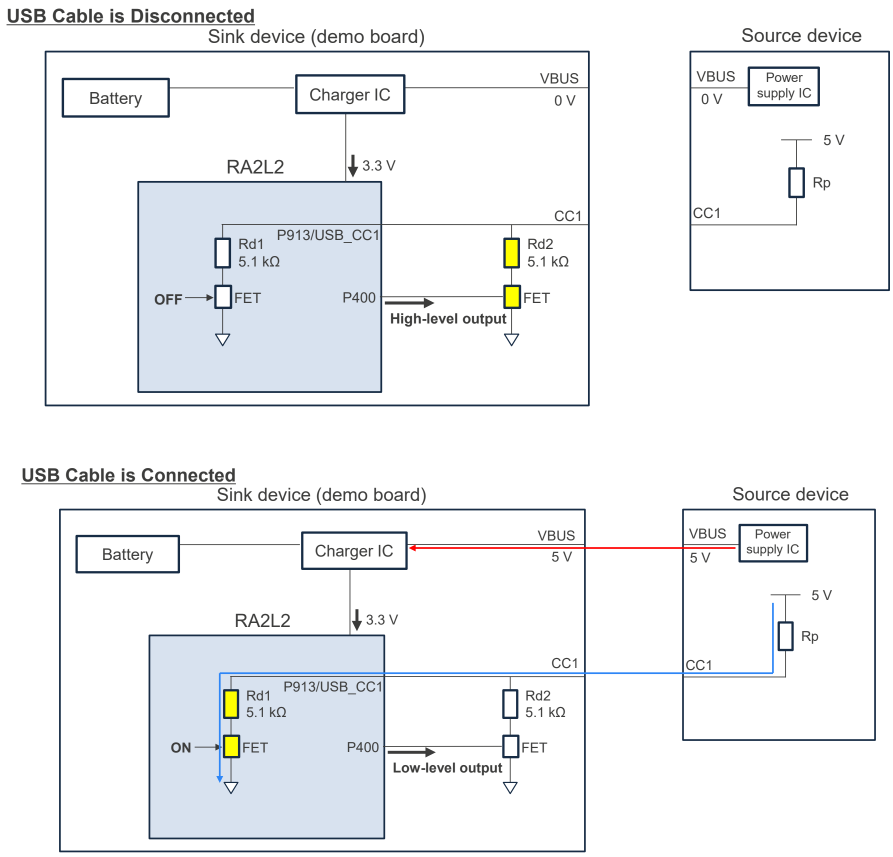
2.4.3 USB Type-C CC Detection

The USB Type-C standard prescribes that a USB Type-C source device such as a PC or a charger shall always detect the pull-down resistor, Rd, on each of the CC lines in a sink device (the demo board in this reference design) when the sink device is connected to the source device through a USB Type-C port. After the source device has detected the Rd pull-down resistor, the device starts VBUS output to the sink device.

The RA2L2 has the Rd1 pull-down resistor in the USB Type-C Interface but the reference design incorporates the external Rd2 pull-down resistor on the demo board so that the source device can detect the Rd pull-down resistor regardless of whether the RA2L2 has been activated. Therefore, the sample code of the reference design switches between the on-chip Rd1 pull-down resistor and external Rd2 pull-down resistor depending on whether a USB cable is connected as described below.

- When a USB Type-C cable is connected, the source detects the external Rd2 pull-down resistor, which is enabled. The source then turns VBUS on and the RA2L2 begins the USB sequence. After that, the on-chip Rd1 pull-down resistor is enabled and the low level is output through P400 and P401 to disable the external Rd2 pull-down resistor (by turning off the external FET immediately below Rd2) so that the USB power supply capability of the source can be detected.
- When a USB Type-C cable is disconnected, VBUS is turned off or the CC lines for detection are disconnected. Therefore, the on-chip Rd1 pull-down resistor is enabled and the high level is output through P400 and P401 to enable the external Rd2 pull-down resister (by turning on the external FET immediately below Rd2).
  In software standby mode, the USB Type-C module in the RA2L2 is shut down to reduce the current consumption. In this case, the on-chip Rd1 pull-down resistor cannot be used, so the Rd1 resistor is disabled and the external Rd2 pull-down resistor is enabled in the same way as when the cable is disconnected.

Figure 2-6 gives an overview of this switching between the on-chip Rd1 pull-down resistor and external Rd2 pull-down resistor.



Note: This figure only shows the circuit for CC1 but the circuit for CC2 has a similar configuration.

Figure 2-6 Overview of Switching between the On-chip Rd2 Pull-down Resistor and External Rd2 Pull-down Resistor

2.5 Charging Circuits in the Reference Design

The demo board has two sources of power: the supply from the battery and USB bus power. The battery is charged via a charger IC (DA9168) made by Renesas. Figure 2-7 is a diagram of the charging block in the DA9168. The red line indicates the route used in charging the battery.

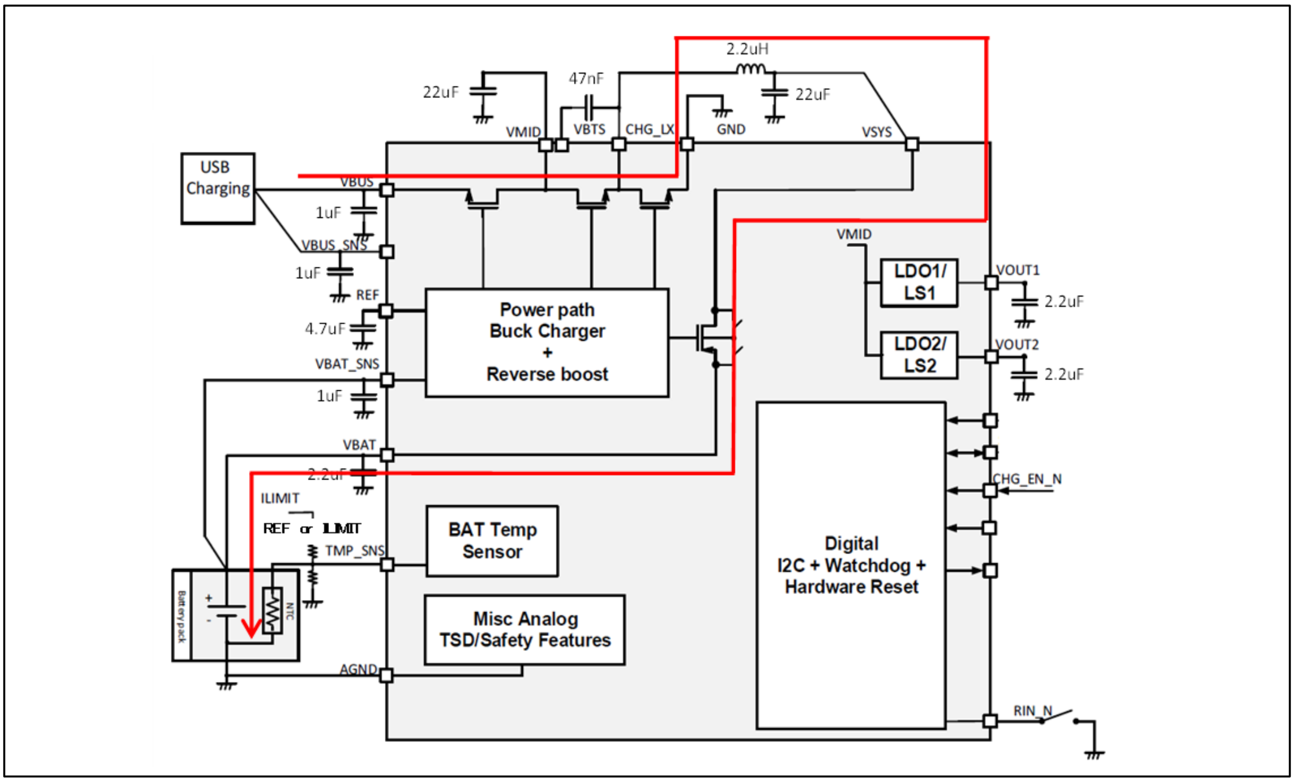


Figure 2-7 Diagram of the Charging Block

The charging voltage is 3.4 V to 4.2 V. The charging current is specified by the RA2L2 through I²C. The current is set to 200 mA in the sample code.

The demo board includes a negative temperature coefficient (NTC) thermistor circuit for measuring the battery temperature but the thermistor is not mounted on the board as shipped. Figure 2-8 shows the peripheral circuit for the thermistor pins on the demo board.

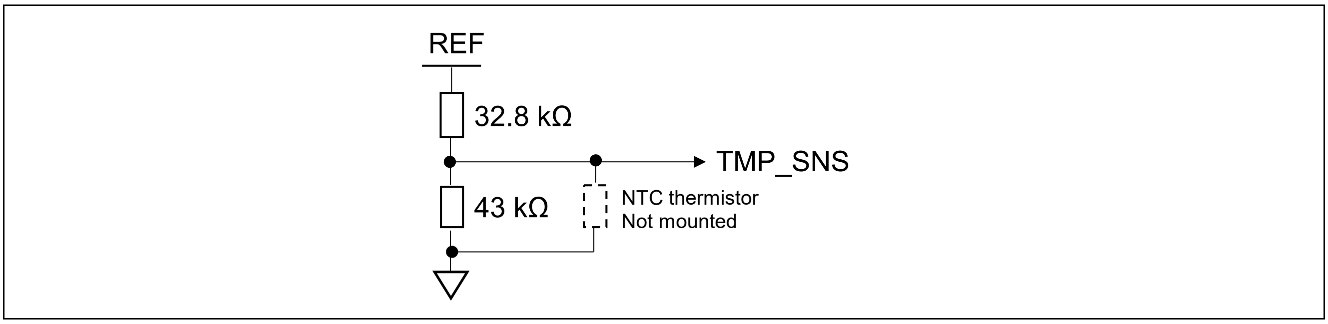


Figure 2-8 Peripheral Circuit for the Thermistor Pin

Figure 2-9 shows the circuit between the RA2L2 and DA9168.

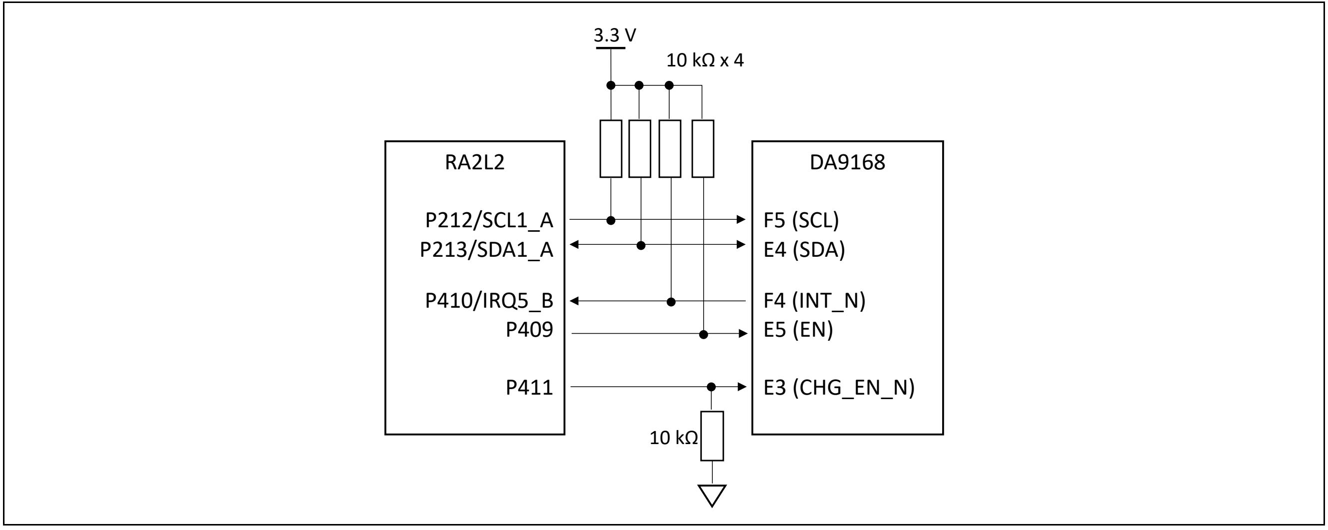


Figure 2-9 Circuit between the RA2L2 and DA9168

2.6 3.3V System Power Supply

The 3.3V system power supply for the demo board is generated by stepping down the VMID output voltage from the DA9168 through an external low-dropout linear regulator (LDO). The RA2L2 uses the 3.3V system power supply. Figure 2-10 shows the circuit of the system power supply

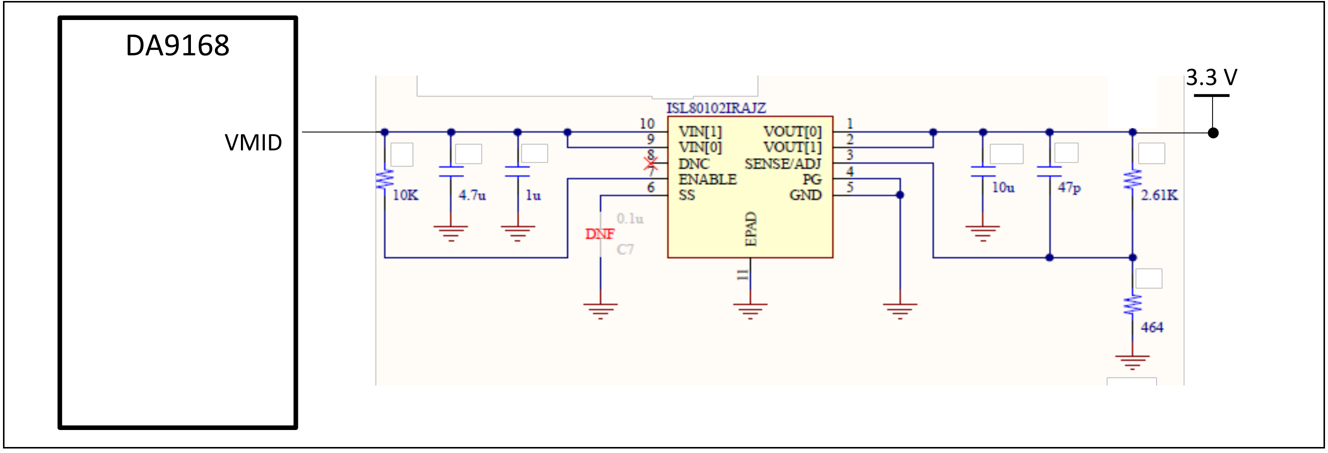


Figure 2-10 Circuit of the System Power Supply

2.7 Controlling Charging of the Battery and Calculating the Remaining Battery Charge in the Reference Design

In the reference design, when the demo board is connected to a USB Type-C source device such as a charger or a PC, the RA2L2 controls the charger IC (DA9168) mounted on the demo board through I²C to charge the battery. This section describes the specifications of the control of battery charging.

2.7.1 Processing Structure

Figure 2-11 shows the processing structure for controlling charging of the battery and calculating the charge remaining in the battery.

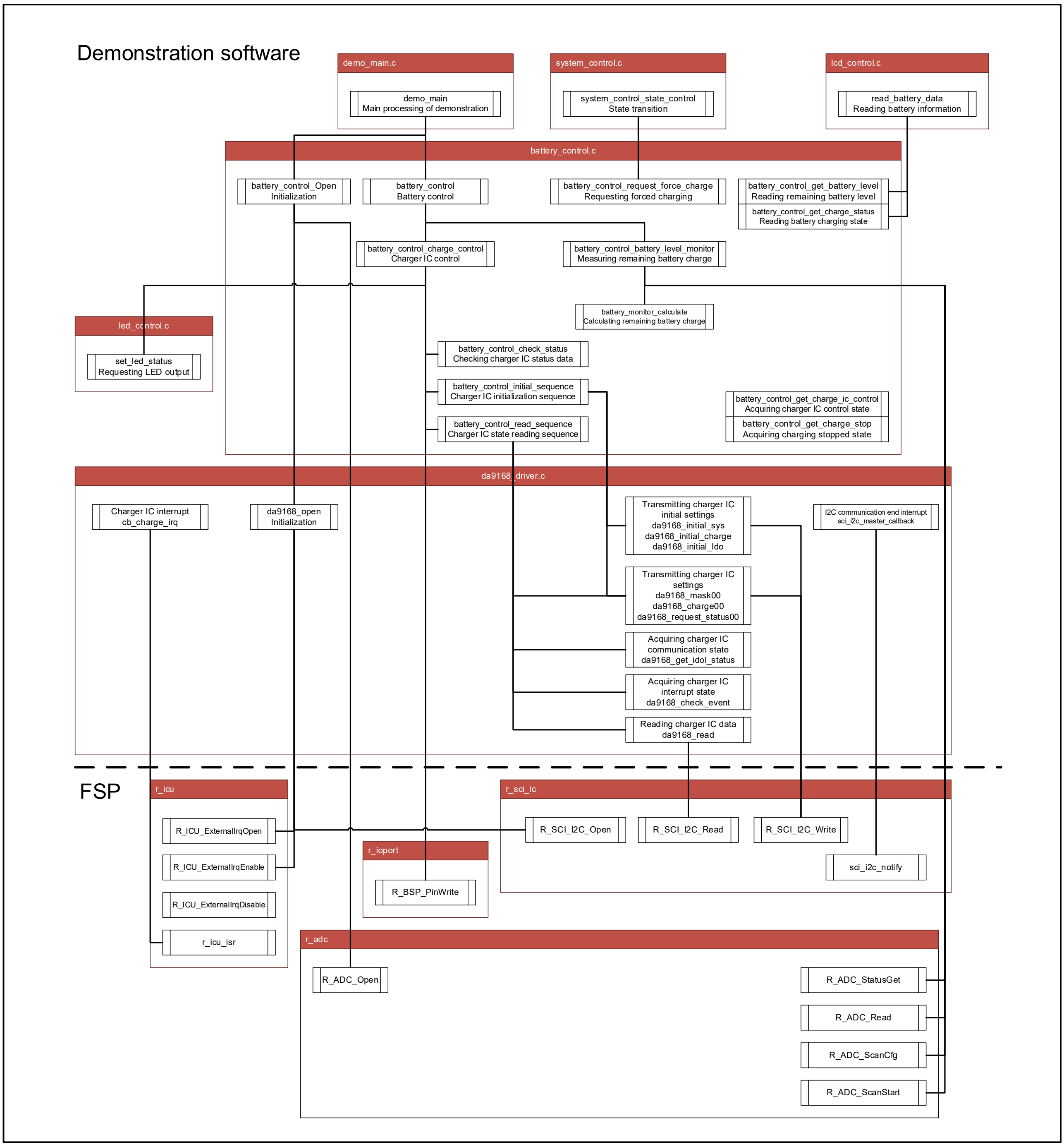


Figure 2-11 Processing Structure for Controlling Charging of the Battery and Calculating the Remaining Charge

2.7.2 Control of Charging

Charging is controlled by using I²C communications with the charger IC, output to the CHG_EN pin, and input of INT_N. The combination of the state of VBUS power supply, the state of the charger IC, and the battery voltage obtained as a result of the remaining charge calculation is used for judgement in the control of charging. The state of the charger IC is acquired through event notifications in the form of the INT_N input from the charger IC. Also refer to Table 2-3 and Figure 2-12.

When a USB connection is made while the demo board is operating with the battery power supply, charging of the battery begins. However, if the battery voltage becomes low (if VBAT_UV is detected), charging is temporarily stopped and the battery voltage is checked again after 500 ms have passed. A dead battery*[1] is detected if the voltage is 0.05 V or higher. If the voltage is no higher than 0.05 V, the battery is judged to have been disconnected (removed). Waiting for 500 ms is required until the voltage input to the battery has settled after the control of charging is stopped.

If connection of a dead battery has been detected, charging can be started upon reception of a forced charging request regardless of the battery voltage. This request is issued by long-pressing the power switch for three seconds during date setting from the system control process except during the initial activation.

If disconnection of the battery has been detected, charging does not begin even if a USB connection is made because no battery is considered present. Even in the battery-disconnected state, various demonstration operations can be executed with the USB bus power supply on the demo board.

Table 2-3 Combinations of Various States and Execution of the Control of Charging

<table>
<tr><th>VBUS Power Supply State</th><th>State of the Charger IC</th><th>Battery Voltage</th><th>State of the Demo Board</th><th>Control of Charging</th><th>Charging LED</th><th>Remark</th></tr>
<tr><td>—</td><td>—</td><td>—</td><td>Battery checking</td><td>Stopped</td><td>Off</td><td></td></tr>
<tr><td>Off</td><td rowspan="2">—</td><td>—</td><td>Charging is stopped.</td><td>Stopped</td><td>Off</td><td></td></tr>
<tr><td rowspan="5">On</td><td rowspan="2">—</td><td>Charging</td><td rowspan="2">Charging</td><td>On</td><td></td></tr>
<tr><td>S_CHG_DONE</td><td>Fully charged</td><td>Off</td><td></td></tr>
<tr><td>S_VBAT_OV</td><td></td><td>Overcharged</td><td>Stopped</td><td>Off</td><td></td></tr>
<tr><td rowspan="2">S_VBAT_UV</td><td>< 0.05 V</td><td>No battery</td><td>Stopped</td><td>Off</td><td></td></tr>
<tr><td>≥ 0.05 V</td><td>Forced charging</td><td>Charging</td><td>On</td><td>A dead battery is connected.</td></tr>
</table>

Note 1. "Dead battery" refers to the state of a battery in which the voltage is at a very low level. The demo board cannot operate with the battery power supply in the dead battery state and so requires the USB power supply in this state.

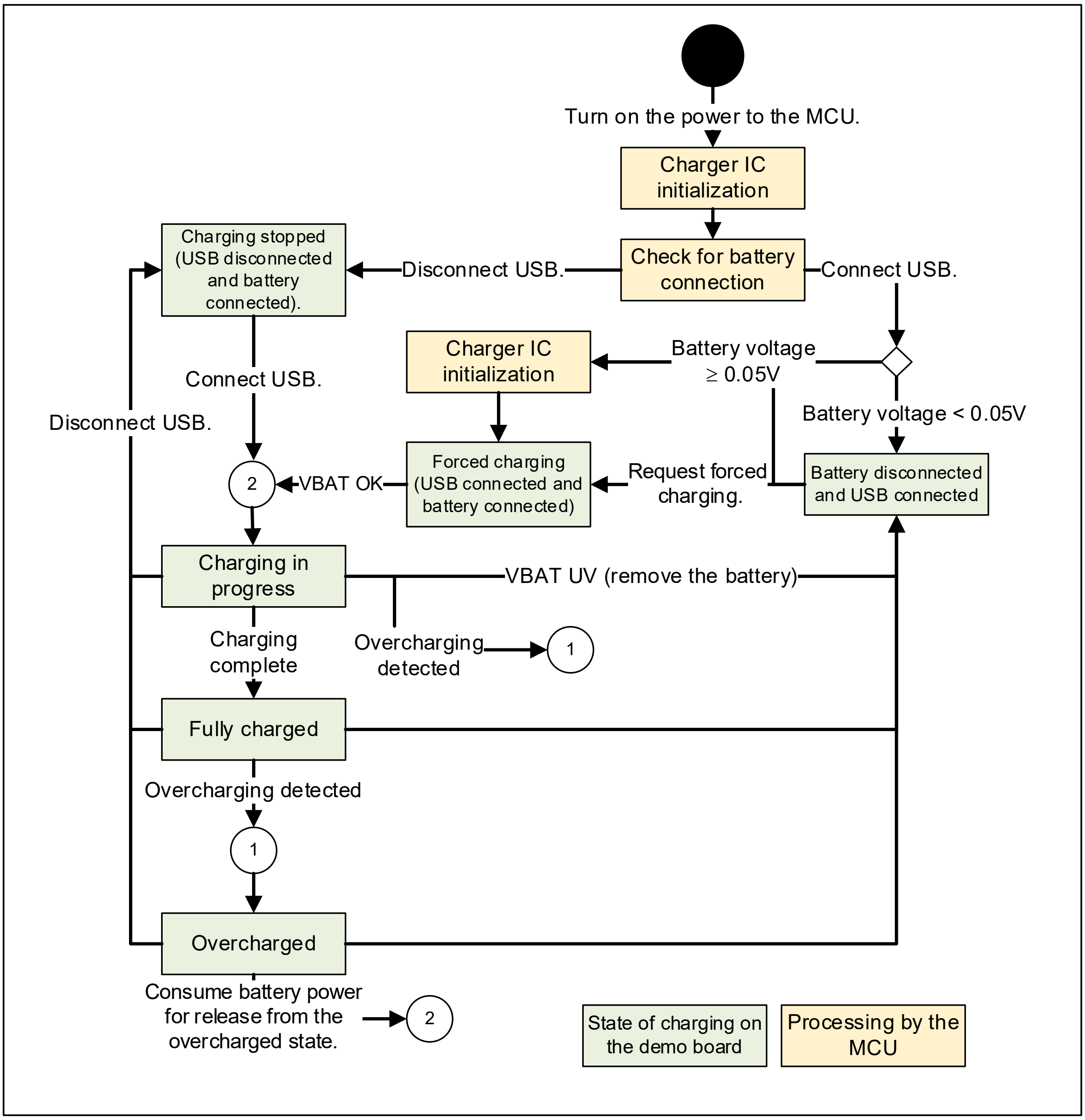


Figure 2-12 State Transitions for Controlling Charging

The charger IC (DA9168) used in the reference design has registers that can be set up through I2C and the settings for the charging operations and protection functions can be modified. Table 2-4 shows the register settings for the DA9168 that are used in the sample code. For details of the register settings, refer to the specifications of the charger IC (DA9168).

Table 2-4 List of Register Settings in the DA9168

Register in the IC	Address	Setting
PMC_MASK_00	0x0A	0xF7
PMC_MASK_01	0x0B	0xEE
PMC_MASK_02	0x0C	0xFF
PMC_MASK_03	0x0D	0xFE
PMC_MASK_04	0x0E	0xFF
PMC_SYS_00	0x0F	0x08
PMC_SYS_01	0x10	0x49
PMC_SYS_02	0x11	0x0A
PMC_SYS_03	0x12	0x34
PMC_SYS_04	0x13	0x19
PMC_SYS_05	0x14	0x92
PMC_SYS_06	0x15	0x56
PMC_CHG_00	0x16	0x0F
PMC_CHG_01	0x17	0x00
PMC_CHG_02	0x18	0x0F
PMC_CHG_03	0x19	0x28
PMC_CHG_04	0x1A	0x17
PMC_CHG_05	0x1B	0x00
PMC_CHG_06	0x1C	0x0C
PMC_LDO_00	0x1D	0x44
PMC_LDO_01	0x1E	0x22
PMC_LDO_02	0x1F	0x22
PMC_LDO_03	0x20	0x44
PMC_LDO_04	0x21	0xAA

Table 2-5 shows the registers that can be read through I2C. The values of the four bits listed in the table are used to obtain the states of charging operations.

Table 2-5 List of Status or Event Registers in the DA9168

Register in the IC	Address	Bit to be Used	Bit Name	Remark
PMC_STATUS_00	0x00	Not used	—	—
PMC_STATUS_01	0x01	Bit 4	S_VBAT_OV	Overcharged
PMC_STATUS_02	0x02	Not used	—	—
PMC_STATUS_03	0x03	Bit 0	S_CHG_DONE	Fully charged
PMC_STATUS_04	0x04	Not used	—	—
PMC_EVENT_00	0x05	Not used	—	—
PMC_EVENT_01	0x06	Bit 4	E_VBAT_OV	Overcharged
PMC_EVENT_02	0x07	Not used	—	—
PMC_EVENT_03	0x08	Bit 0	E_CHG_DONE	Fully charged
PMC_EVENT_04	0x09	Not used	—	—

2.7.3 Calculating the Charge Remaining in the Battery

The sample code calculates the charge remaining in the battery based on the battery voltage that is obtained from the results of A/D conversion of the voltage on an ADC pin.

A moving average of the results of A/D conversion is obtained and used in calculating the remaining charge. Note that the range of ± 1% for a boundary value between remaining charge levels is determined as a hysteresis section and changes within this section are not reflected in the display of the remaining charge level.

Table 2-6 shows the voltages defined for use in calculating the remaining charge and Table 2-7 shows the correspondence between the charge remaining in the battery and the remaining level indications.

$$\text{Moving average of A/D conversion results} = 0.25 * \text{A/D conversion results} + 0.75 * \text{previous A/D moving average}$$

$$\text{When moving average of A/D conversion results} < \text{MIN voltage, moving average of A/D conversion results} = \text{MIN voltage}$$

$$\text{When moving average of A/D conversion results} > \text{MAX voltage, moving average of A/D conversion results} = \text{MAX voltage}$$

$$\text{Battery voltage} = \frac{\text{reference voltage}}{2^{12} - 1} * \text{moving average of A/D conversion results}$$

$$\text{Remaining charge} = \frac{\text{battery voltage} - \text{MIN voltage}}{\frac{\text{MAX voltage} - \text{MIN voltage}}{100}}$$

Table 2-6 Voltages Defined for Use in Calculating the Remaining Charge

Item	Voltage	Remark
Reference voltage	3.3 V	AVCC0
Maximum (MAX) voltage	2.1 V	4.2 V*1 ÷ 2 Note 1. Maximum voltage of the battery
Minimum (MIN) voltage	1.2 V	2.4 V*2 ÷ 2 Note 2. Minimum voltage of the battery

Table 2-7 Correspondence between the Charge Remaining in the Battery and the Remaining Level Indications

Charge Remaining in the Battery	Remaining Level Indication	Remark
Up to 29%	0	Up to 30 – 1%
31% to 44%	1	30 + 1% to 45 – 1%
46% to 59%	2	45 + 1% to 60 – 1%
61% to 74%	3	60 + 1% to 75 – 1%
76% to 89%	4	75 + 1% to 90 – 1%
91% or more	5	90 + 1% or more

3. Specifications of Software Operations (Demonstration Operations)

The sample code for the reference design implements demonstrations not only of USB Type-C operation but also of the operations of the entire system intended for a small-size application powered by battery. This chapter describes the specifications of the software operations other than the USB Type-C CC detection, battery charging control, and remaining charge calculation operations described in Chapter 2. Sections 3.1 to 3.4 give an overview of software operations and section 3.5 and subsequent sections describe the specifications of the individual functions.

3.1 Flowchart of the Main Processing

Figure 3-1 is a flowchart of the main processing of the sample code. The LCD drawing processing is separated from the main cycle to improve the speed of drawing. The VBUS monitoring processing is also separated from the main cycle because the monitoring processing runs in a different cycle from the main cycle.

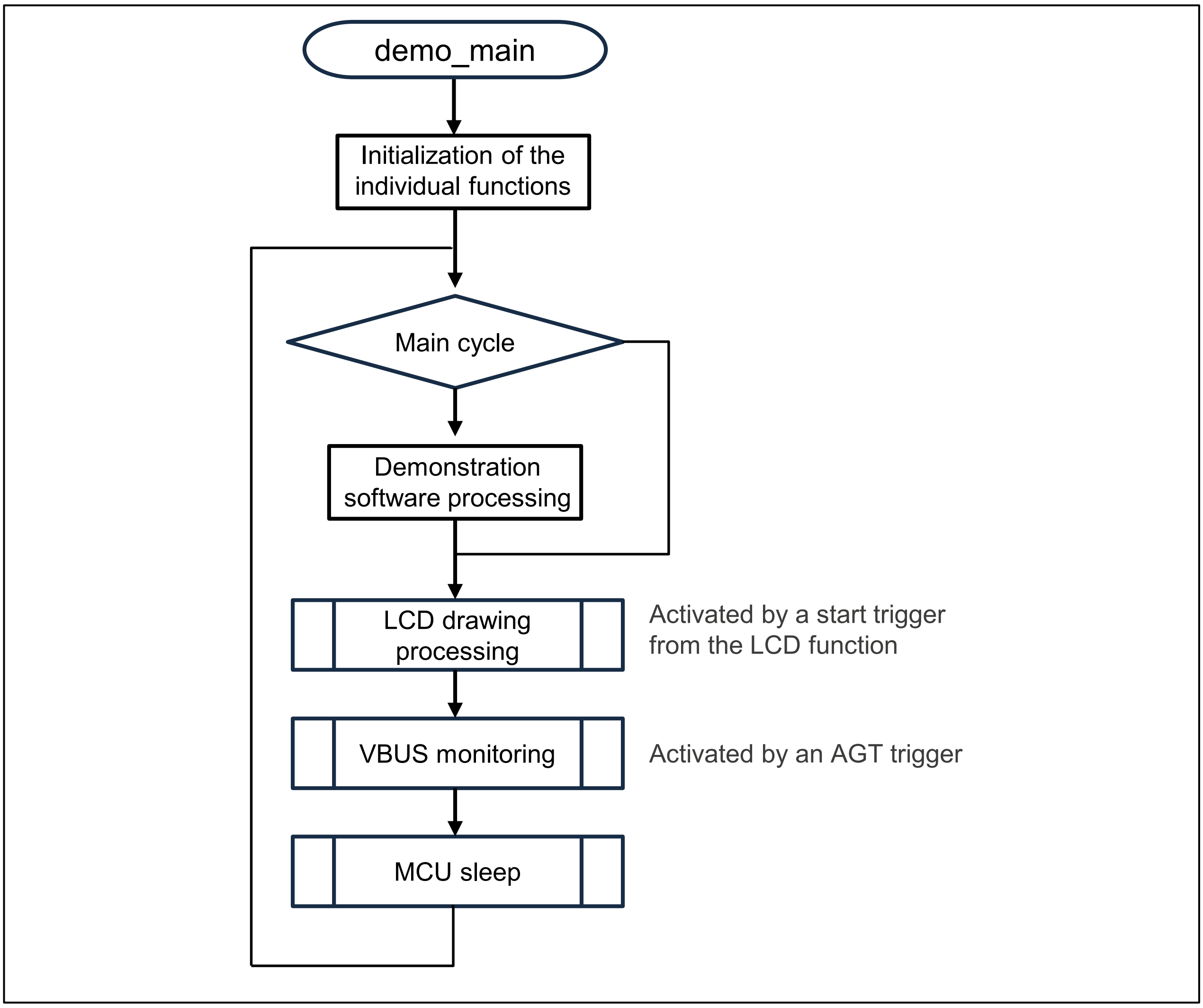


Figure 3-1 Flowchart of the Main Processing

3.2 Overview of Demonstration Operations and State Transitions

Table 3-1 shows an overview of the demonstration operations of the sample code.

Table 3-1 Overview of the Demonstration Operations

Demonstration Operation	Overview
Date and time setting mode	Mode for setting the date and time by using the joystick.
Sensing demo mode	Mode for displaying the temperature and humidity values and the results of detection by the motion sensor on the color LCD. In addition, the data from the sensor can be sent to a PC by connecting the board as a USB CDC device to the PC.
Mouse demo mode	Mode for operating the board as a USB mouse by connecting it as a USB HID device to a PC. The mouse cursor can be moved by operating the joystick.
USB notification	Mode for displaying the results of USB Type-C CC detection and the state of USB connection on the LCD as USB notification when a USB connection is made or cut off. The notification automatically ends in three seconds and execution returns to the previous demo mode.
Power-saving mode	Mode for reducing the power consumption by stopping the display on the LCD and measurement by the sensors.
Standby mode	Mode for the maximum reduction of power consumption by stopping the USB Type-C interface in addition to the steps taken for the power-saving mode.

Figure 3-2 shows the state transitions of the entire sample code.

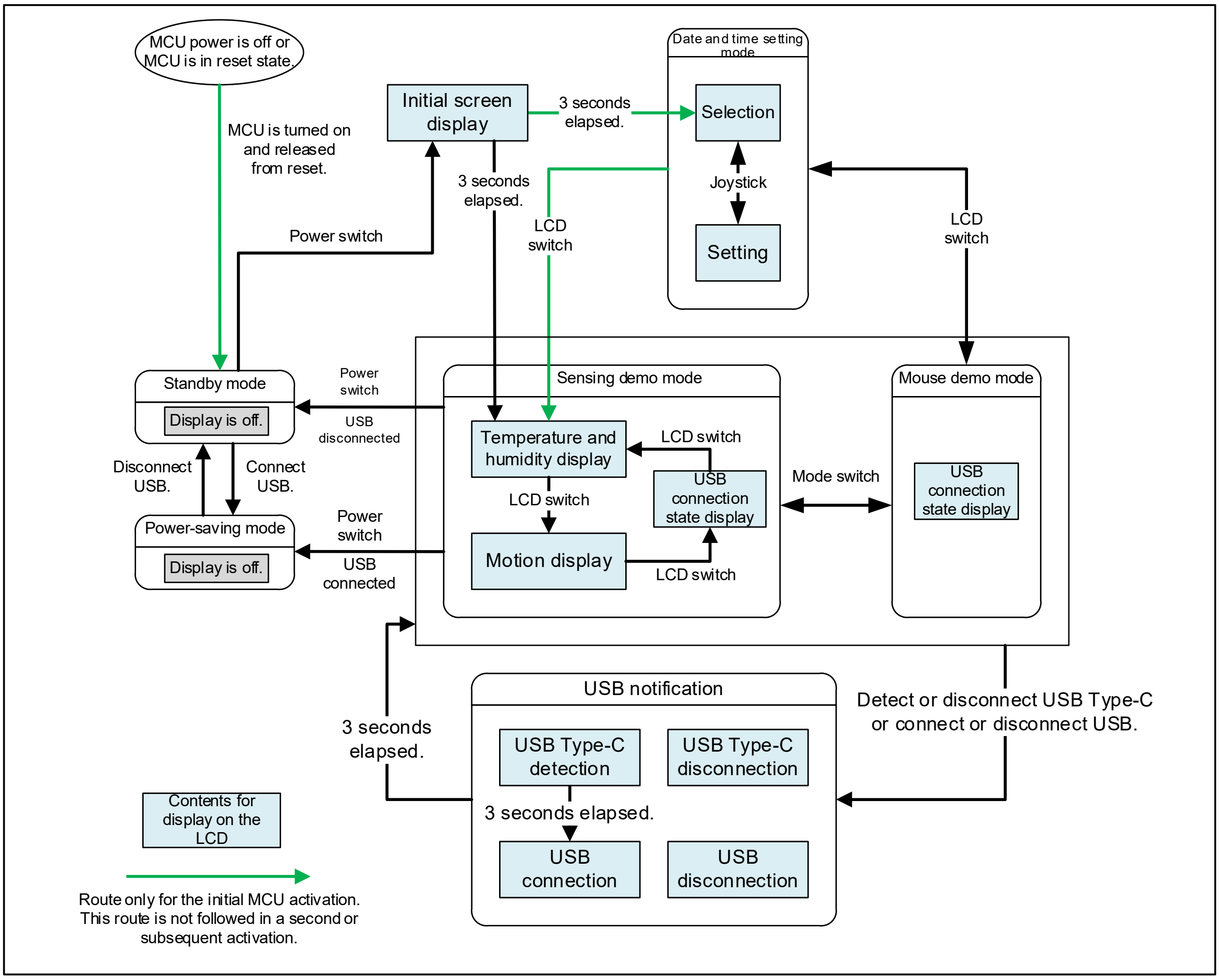


Figure 3-2 State Transitions of the Entire Sample Code

Table 3-2 lists the states of the major functions in the individual demo modes.

Table 3-2 States of the Major Functions in the Individual Demo Modes

Operation Mode	Sensing	LCD Display	Charging	USB Type-C Interface*1	USBFS
Sensing demo mode	Operating	On	Operating	Operating	Operating
Mouse demo mode	Operating	On	Operating	Operating	Operating
Date and time setting mode	Stopped	On	Operating	Operating	Operating
Power-saving mode	Stopped	Off	Operating	Operating	Operating
Standby mode	Stopped	Off	Stopped	Stopped	Stopped

Note 1. On-chip USB Type-C interface in the RA2L2.

3.3 Software Structure

Figure 3-3 shows the overall structure of the sample code software.

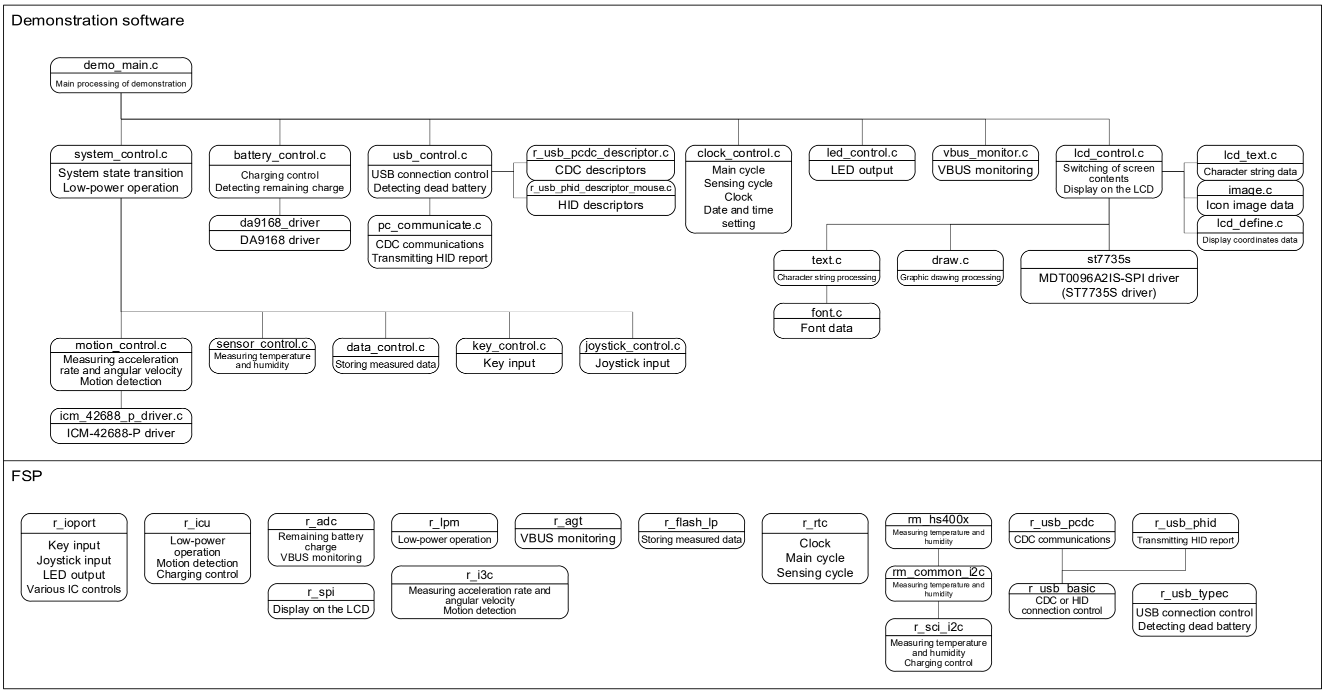


Figure 3-3 Overall Structure of the Sample Code Software

3.4 Structure of Folders and Files

Table 3-3 shows the overall structure of the sample code folders and files and gives descriptions of the files.

Table 3-3 Structure of Folders and Files

Folder or File Name	Description
ra2l2_usb_reference_demo_sample	
|— Src |— hal_entry.c	
|— demo_main.c	Main processing of demonstration
|— demo_main.h	Header file for demo_main.c
|— demo	
|— battery	
| |— battery_control.c	Charging control and detection of remaining charge
| |— battery_control.h	Header file for battery_control.c
| |— da9168_driver.c	DA9168 driver
| |— da9168_driver.h	Header file for da9168_driver.c
|— clock	
| |— clock_control.c	Main cycle, sensing cycle, clock, and date and time setting
| |— clock_control.h	Header file for clock_control.c
|— data	
| |— data_control.c	Storing measured data
| |— data_control.h	Header file for data_control.c
|— joystick	
| |— joystick_control.c	Joystick input
| |— joystick_control.h	Header file for joystick_control.c
|— key	
| |— key_control.c	Key input
| |— key_control.h	Header file for key_control.c
|— lcd	
| |— gfx	
| | |— assert	
| | | |— display_assert.h	Macros for checking parameters of the LCD API
| | |— draw	
| | | |— draw.c	Graphic drawing processing
| | | |— draw.h	Header file for draw.c
| | | |— draw_driver.h	Definitions of the interface for connecting draw.c and the hardware-specific driver (st7735s.c) for the LCD
| | |— driver	
| | | |— st7735s.c	LCD drawing setting API for MDT0096A2IS-SPI (st77635s)
| | | |— st7735s.h	
| | | |— st7735s_port.c	Definitions of the MDT0096A2IS-SPI (st77635s) pins and MCU function (SPI)
| | |— text	

| | | |— font.c	Font data
| | | |— font.h	Header file for font.c
| | | |— text.c	Character string processing
| | | |— text.h	Header file for text.c
| | | |— text_driver.h	Definitions of the interface for connecting text.c and the hardware-specific driver (st7735s.c) for the LCD
| | |— images.c	Icon image data
| | |— images.h	Header file for images.c
| |— lcd_control.c	Switching of screen contents and displaying on LCD
| |— lcd_control.h	Header file for lcd_control.c
| |— lcd_define.c	Display coordinates data
| |— lcd_define.h	Header file for icd_define.c
| |— lcd_text.c	Character string data
| |— lcd_text.h	Header file for lcd_text.c
|— led	
| |— led_control.c	LED output
| |— led_control.h	Header file for led_control.c
|— motion	
| |— icm_42688_p_driver.c	ICM-42688-P driver
| |— icm_42688_p_driver.h	Header file for icm_42688_p_driver
| |— motion_control.c	Measurement of acceleration rate and angular velocity and detection of motion
| |— motion_control.h	Header file for motion_control.c
|— sensor	
| |— sensor_control.c	Measurement of temperature and humidity
| |— sensor_control.h	Header file for sensor_control.c
|— sys	
| |— system_control.c	System state transition and low-power operation
| |— system_control.h	Header file for system_control.c
|— usb	
| |— pc_communicate.c	CDC communications and HID report transmission
| |— pc_communicate.h	Header file for pc_communicate.c
| |— r_usb_pcdc_descriptor.c	CDC descriptors
| |— r_usb_pcdc_descriptor_mouse.c	HID descriptors
| |— usb_control.c	USB connection control
| |— usb_control.h	Header file for usb_control.c
|— vbus	
| |— vbus_monitor.c	VBUS monitoring
| |— vbus_monitor.h	Header file for vbus_monitor.c

3.5 System Control

This section describes the specifications of the system control for the entire sample code. The RTC periodical interrupt at 1/128-second intervals is used to generate the main cycle of the system. The duration of long-pressing of the power switch and the LCD drawing time are also counted in units of this main cycle.

3.5.1 Processing Structure

Figure 3-4 shows the structure of the system control processes.

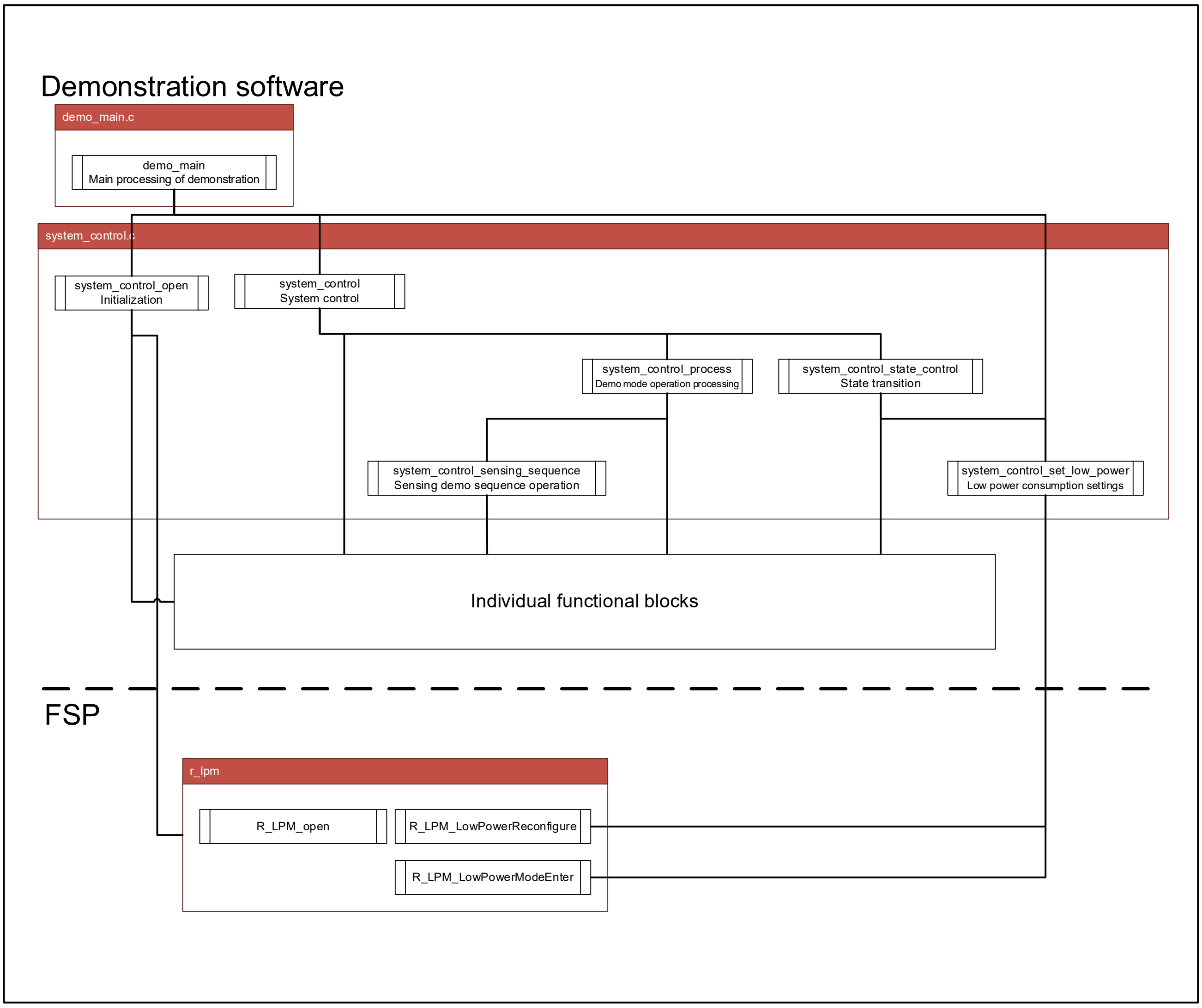


Figure 3-4 Structure of the System Control Processes

3.5.2 State Transitions

Figure 3-5 shows the state transitions for system control. The demonstration software is started by pressing the power switch in standby mode or power-saving mode. The system is placed in date and time setting mode when the power switch is pressed for the first time and is placed in sensing demo mode otherwise. Pressing the LCD switch in the first-time date and time setting mode places the system in sensing demo mode. Pressing the mode switch during the demonstration can switch the mode between the sensing demo and mouse demo.

Pressing the LCD switch in mouse demo mode causes a transition to date and time setting mode and pressing it in date and time setting mode causes a transition to mouse demo mode.

If the power switch is pressed for three seconds in sensing demo mode, the system is placed in power-saving mode. The system is automatically shifted from power-saving mode to standby mode when the following three conditions are satisfied.

1. All software processes in the RA2L2 have ended.
2. All switches are released (not being pressed).
3. USB Type-C is disconnected.

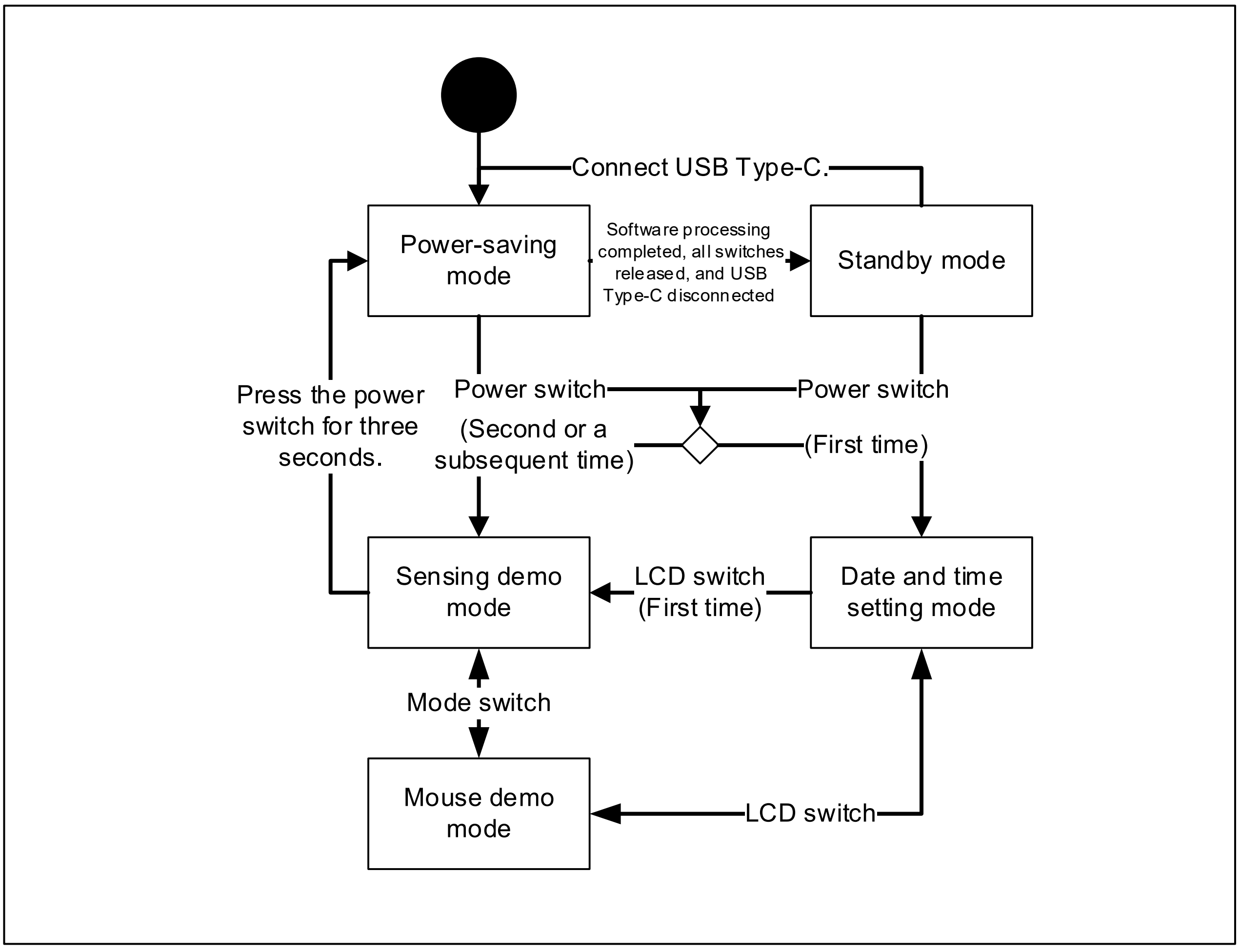


Figure 3-5 State Transitions for System Control

3.5.3 Low-Power Operation

Figure 3-6 shows the state transitions for low-power operation and Table 3-4 lists the states of operation of the peripheral modules in the individual operating modes of the MCU.

For low-power operation, the RA2L2 is placed in software standby mode if the system is in standby mode or placed in sleep mode if the system is in sensing demo, mouse demo, or date and time setting mode.

In standby mode, the system checks that the initialization and operations of the individual peripheral modules have ended and the input from the power switch has become off, after which the MCU is placed in software standby mode. However, if the USB Type-C cable is still connected, this transition to software standby mode does not proceed.

While the MCU is in software standby mode, the USB Type-C module is stopped. An input from the power switch or the INT input from the charger IC in software standby mode releases the MCU from software standby mode.

After release from software standby mode, the USB Type-C module is restarted. The INT input from the charger IC is handled as detection of USB Type-C connection and the battery control function is executed after release from software standby mode. If the USB Type-C connection is not present, the MCU returns to the software standby mode.

When release from software standby mode is caused by the input from the power switch, the system is placed in sensing demo mode. In sensing demo, mouse demo, or date and time setting mode, the MCU is placed in sleep mode while no execution is in progress after the processing of each peripheral module at main cycle intervals has ended. The MCU is released from sleep mode by an RTC interrupt for main cycle generation or a timer interrupt for VBUS monitoring.

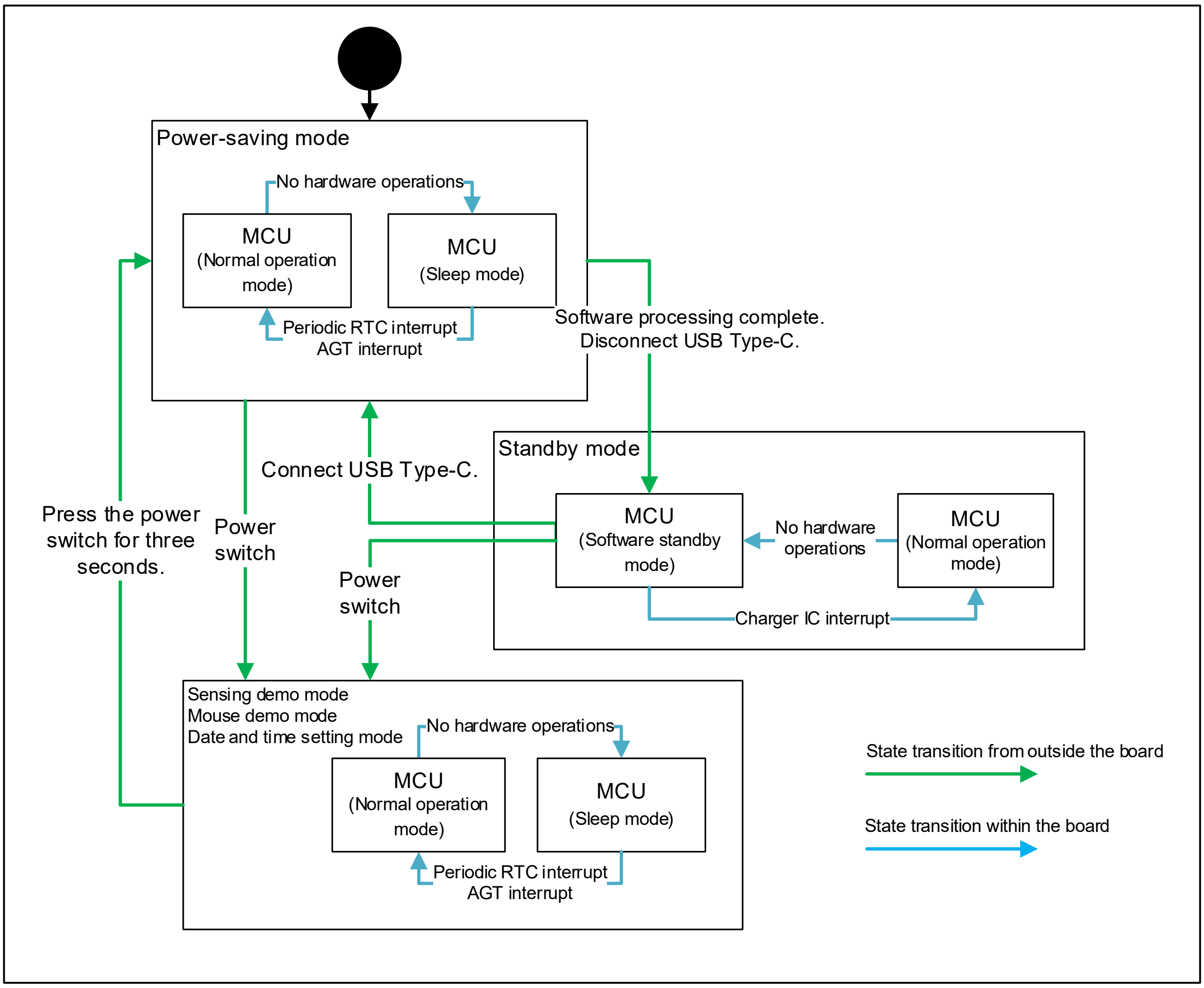


Figure 3-6 State Transitions for Low-Power Operation

Table 3-4 States of Operation of the Peripheral Modules in the Individual Operating Modes of the RA2L2

Item	Normal Operation Mode	Sleep Mode	Software Standby Mode
Sub-clock oscillator	Operating	Operating	Operating
High-speed on-chip oscillator	Operating	Operating	Stopped
CPU	Operating	Stopped	Stopped
SRAM	Operating	Operating	Stopped
Flash memory	Operating	Operating	Stopped
Watchdog time	Operating	Operating	Stopped
Realtime clock	Operating	Operating	Operating
Low power asynchronous general purpose timer	Operating	Operating	Operating
12-bit A/D converter	Operating	Operating	Stopped
Serial communications interface	Operating	Operating	Stopped
I3C bus interface	Operating	Operating	Stopped
USB2.0 full-speed module	Operating	Operating	Stopped
USB Type-C interface	Operating	Operating	Stopped
NMI, IRQn (n = 0 to 7) pin interrupt	Operating	Operating	Operating
Low voltage detection	Operating	Operating	Operating
I/O ports	Operating	Operating	Settings retained*1

Note 1. The previous settings of the ports immediately before the transition to software standby mode are retained.

3.6 Key Input

The inputs from the power, mode, and LCD switches are monitored at intervals of the main cycle in sensing demo mode, mouse demo mode, and date and time setting mode. When an input matches twice in succession, the input value is considered confirmed. Note that the reset switch is used for a hardware reset and the key input control processing does not handle the input from the switch. Table 3-5 lists the specifications of the key input.

Table 3-5 Specifications of the Key Input

Switch	Block Using the Input	Active Level	Remark
Power switch	System control	Low	
Reset switch	Hardware	Low	
Mode switch	System control	Low	
LCD switch	System control and LCD control	Low	The LCD control process receives this key input through the system control processing.

3.6.1 Processing Structure

Figure 3-7 shows the structure of the key input processes.

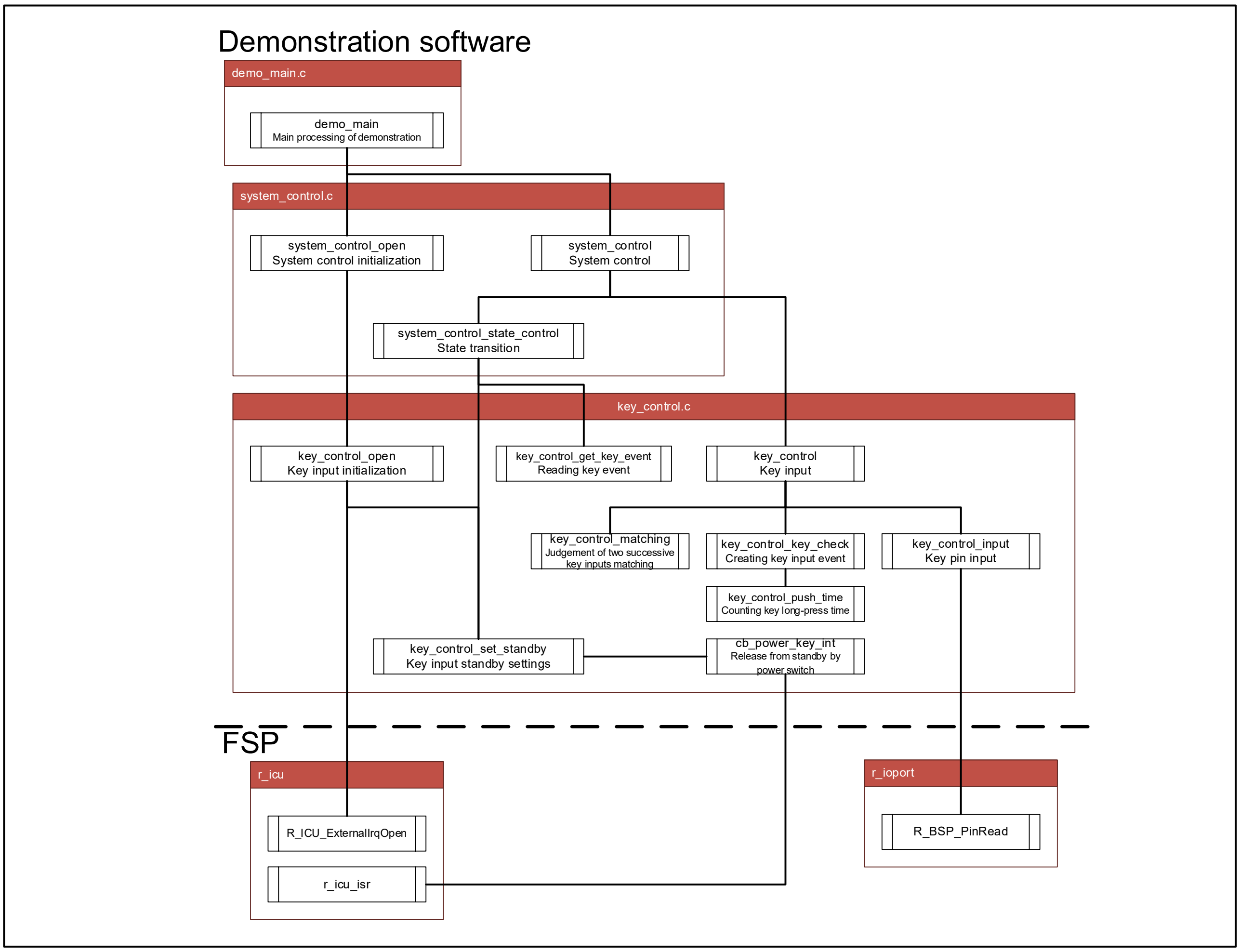


Figure 3-7 Structure of the Key Input Processes

3.6.2 Key Input Interrupt

If the interrupt due to the input from the power switch occurs in standby mode, the system is released from standby mode and placed in sensing demo mode.

3.7 Joystick Input

The input from the joystick is monitored at intervals of the main cycle in date and time setting mode. When the input matches twice in succession, the input value is considered confirmed. Figure 3-8 shows the specifications for the input directions of the joystick. Table 3-6 lists the MCU pins assigned to the joystick. Note that the input directions differ between mouse demo mode and date and time setting mode in the way shown in Figure 3-8.

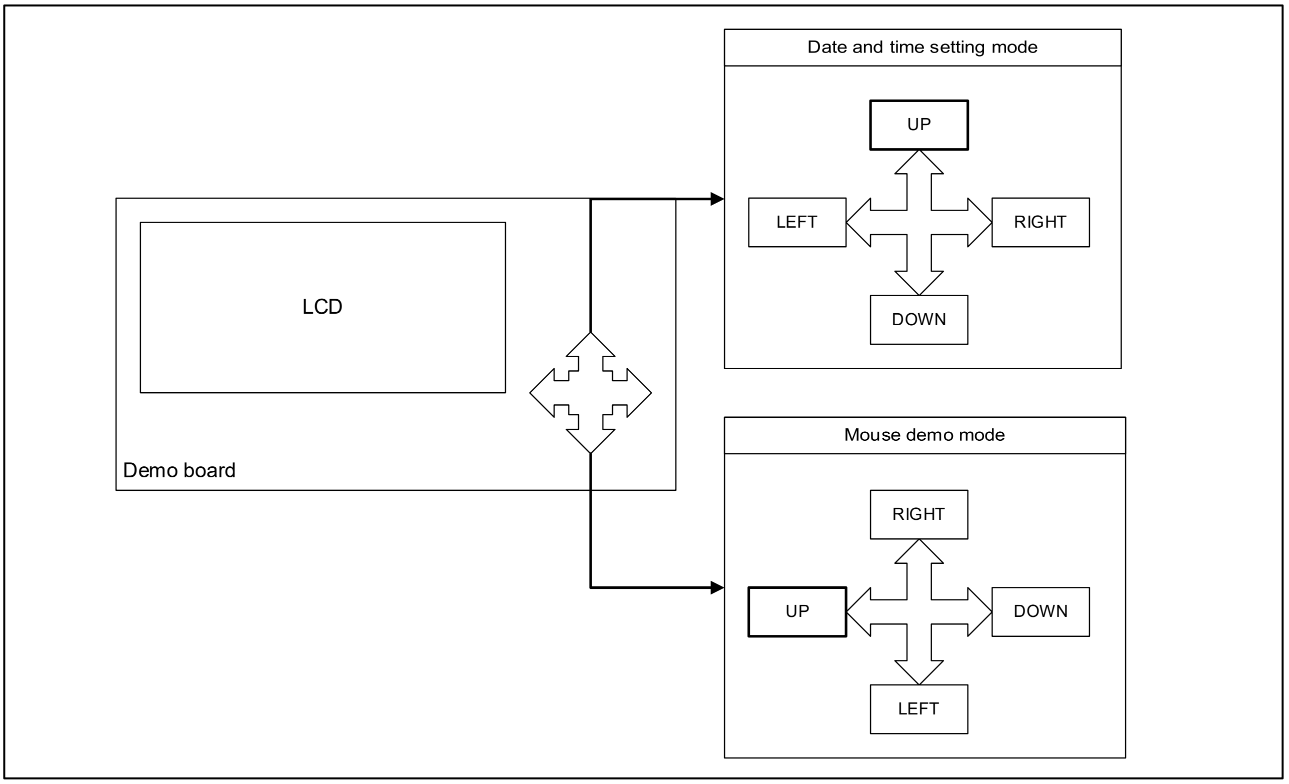


Figure 3-8 Specifications of Joystick Input

Table 3-6 Pins Assigned to the Joystick

Input	Date and Time Setting Mode	Mouse Demo Mode	Pin Name	Active Level
A	Up	Right	P013	Low
B	Right	Down	P012	Low
C	Left	Up	P011	Low
D	Down	Left	P010	Low
Center	Switch between selection and setting	Left-click of the mouse	P502	Low

Inputs A to D and Center in Table 3-6 indicate the directions of joystick input. For details, see Figure 5-4 in section 5.4, Joystick.

3.7.1 Processing Structure

Figure 3-9 shows the structure of the joystick input processes.

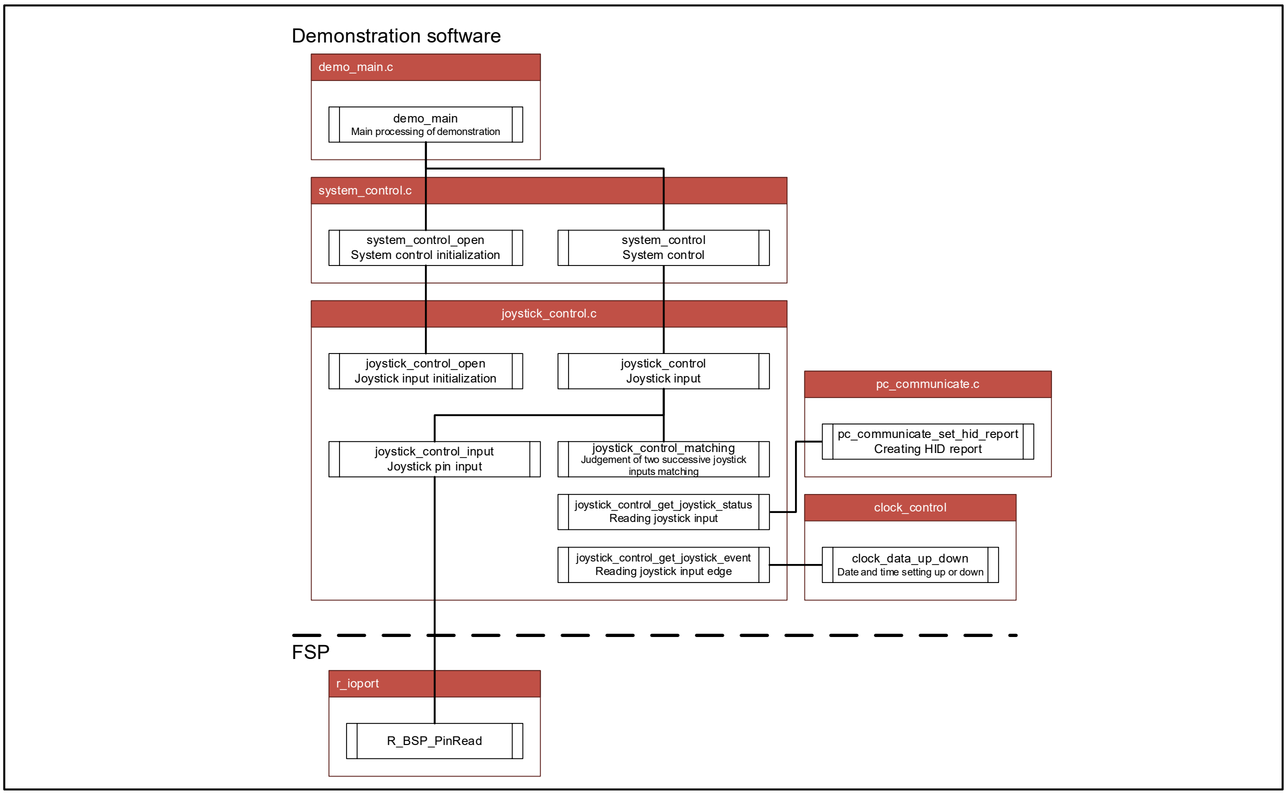


Figure 3-9 Structure of the Joystick Input Processes

3.7.2 Mouse Demo Operation

In mouse demo mode, the state of input from the joystick and an HID report are sent to the PC application. For the joystick input direction, input by pressing towards the left side of the board is handled as upward input.

3.7.3 Date and Time Setting Operation

The date and time can be set by pressing the joystick up, down, left, and right and pressing the center of the joystick. For the joystick input direction, input by pressing towards the upward side of the board is handled as upward input.

3.8 Status LEDs

Table 3-7 lists the specifications of status LED output control. The LED outputs are retained even on standby. The signal for the power LED is also used to activate the LCD’s backlight.

Table 3-7 Specifications of Status LED Output Control

Status LED	Control Block	Active Level	Remark
Power LED	System state transition	High	The signal for this LED is also used for LCD's backlight.
USB LED	USB control	High	
Charging LED	Battery control	High	

3.8.1 Processing Structure

Figure 3-10 shows the structure of the LED output processes.

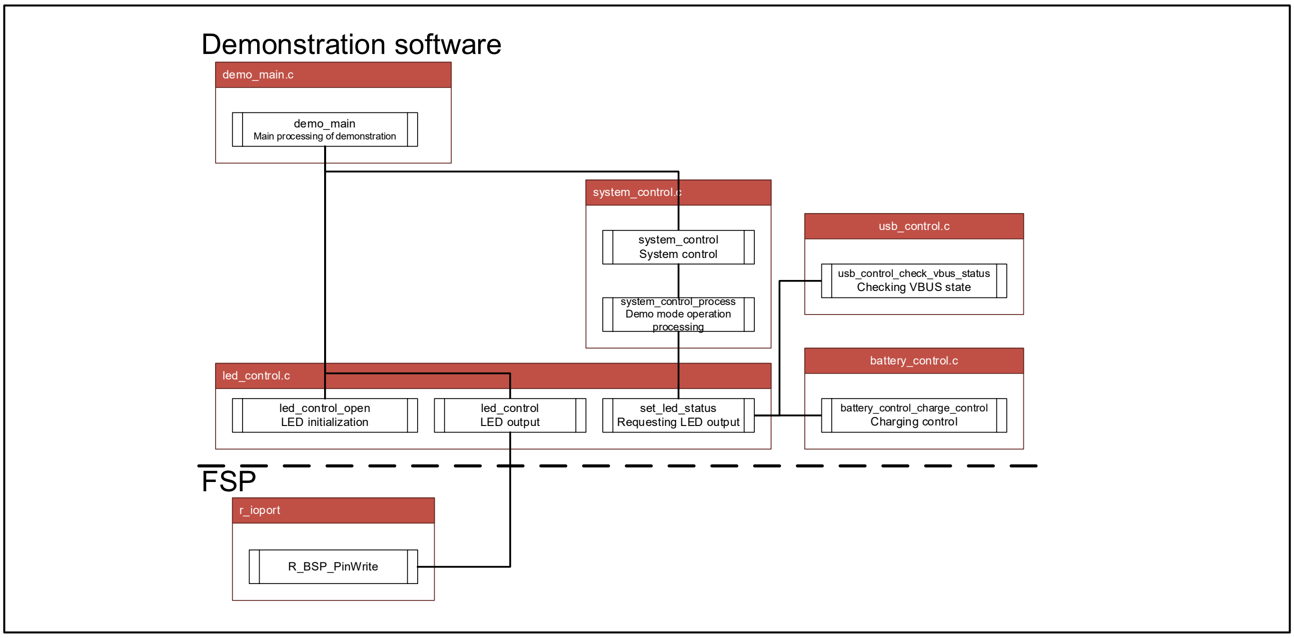


Figure 3-10 Structure of the LED Output Processes

3.9 Date and Time Control

The RTC is used to control the date and time. The time is handled in the 24-hour system. The periodic RTC interrupt at 1/128-second intervals is used to generate the main cycles.

3.9.1 Processing Structure

Figure 3-11 shows the structure of the date and time control processes.

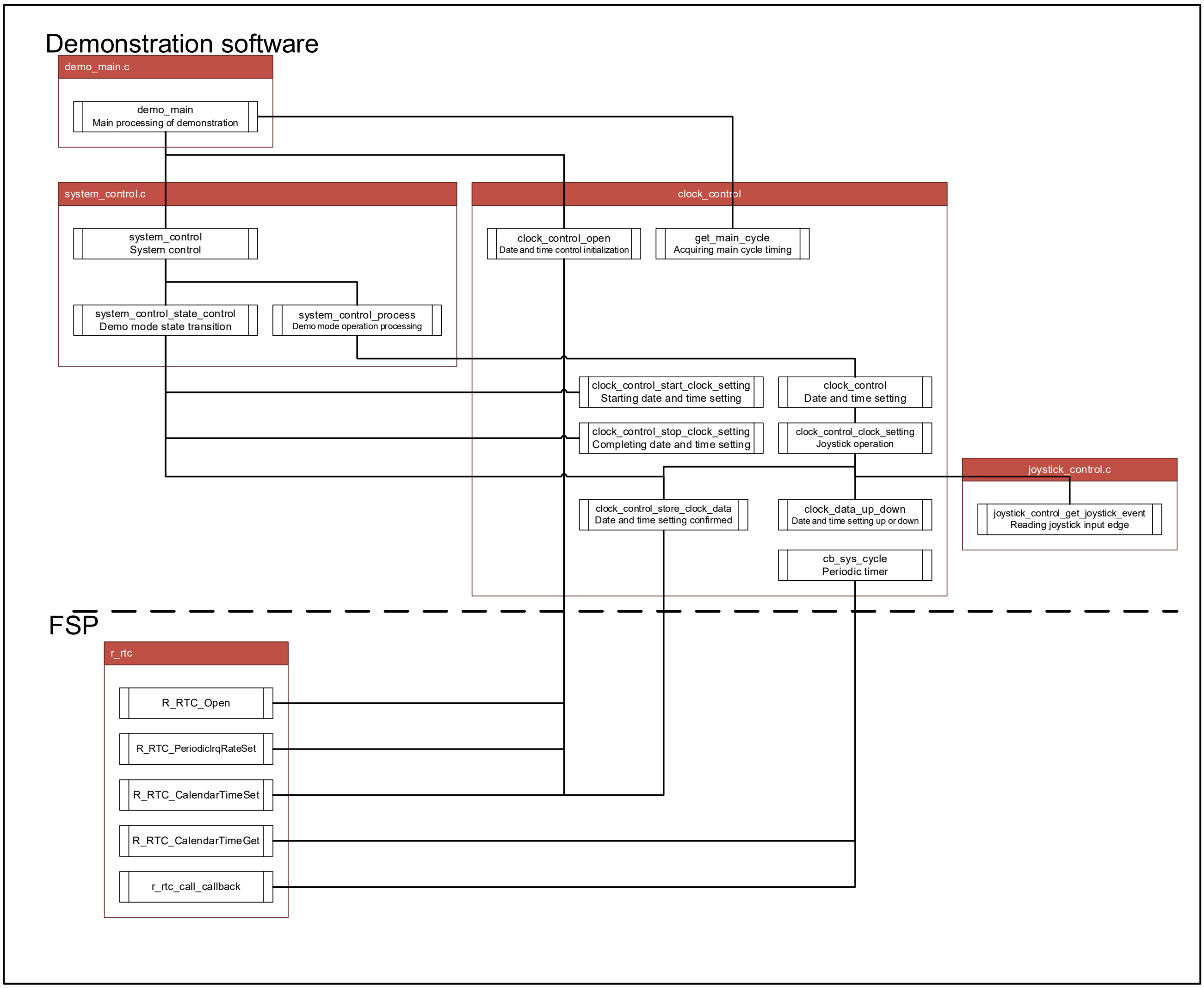


Figure 3-11 Structure of the Date and Time Control Processes

3.9.2 Date and Time Setting

When the power supply begins before the date and time have been set, the system is automatically placed in date and time setting mode. The system is also placed in date and time setting mode in response to the LCD switch being pressed in mouse demo mode. Pressing the LCD switch in date and time setting mode reflects the setting in the RTC and ends the date and time setting operation.

If date and time setting is started in the state where the date and time have not been set, the system moves to sensing demo mode after the setting has been made. If date and time setting is started in mouse demo mode, the system returns to mouse demo mode after the setting has been made.

Select an item first and then modify the set value in date and time setting mode. Press the joystick up, down, left, or right to select an item, that is, year, month, day, hour, minute, or second, and then press the center of the joystick to proceed with modification of the value.

Modify the value of the selected item by pressing the joystick up or down. Press the center to confirm the value and return to selection of an item.

Figure 3-12 is an overview of the date and time setting operations.

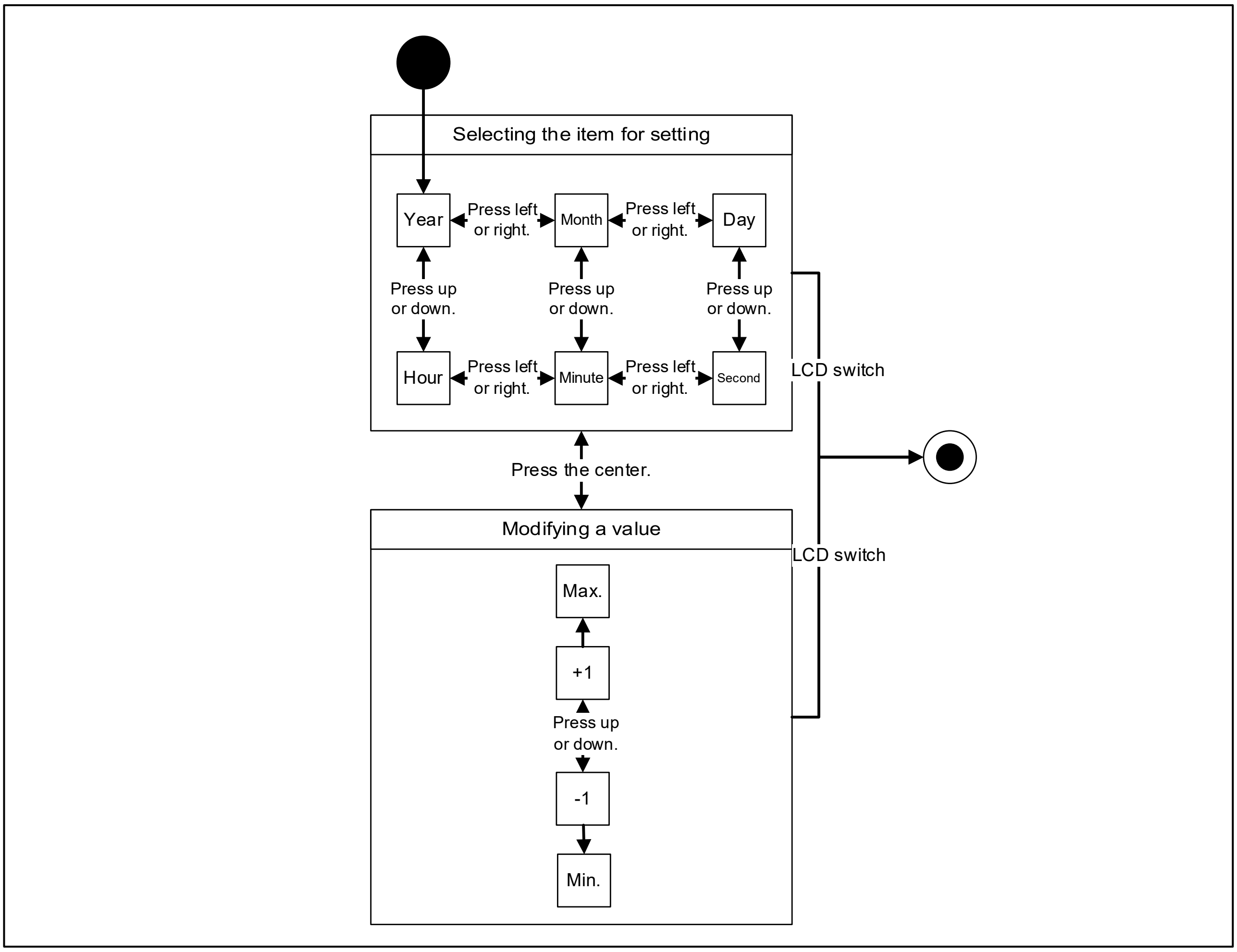


Figure 3-12 Overview of Date and Time Setting Operations

3.10 LCD Display

The sample code uses the LCD display control functions and LCD module driver to draw the icon image data and text data, which have been converted to C source code, on the LCD.

3.10.1 Processing Structure

Figure 3-13 shows the structure of the LCD display processes.

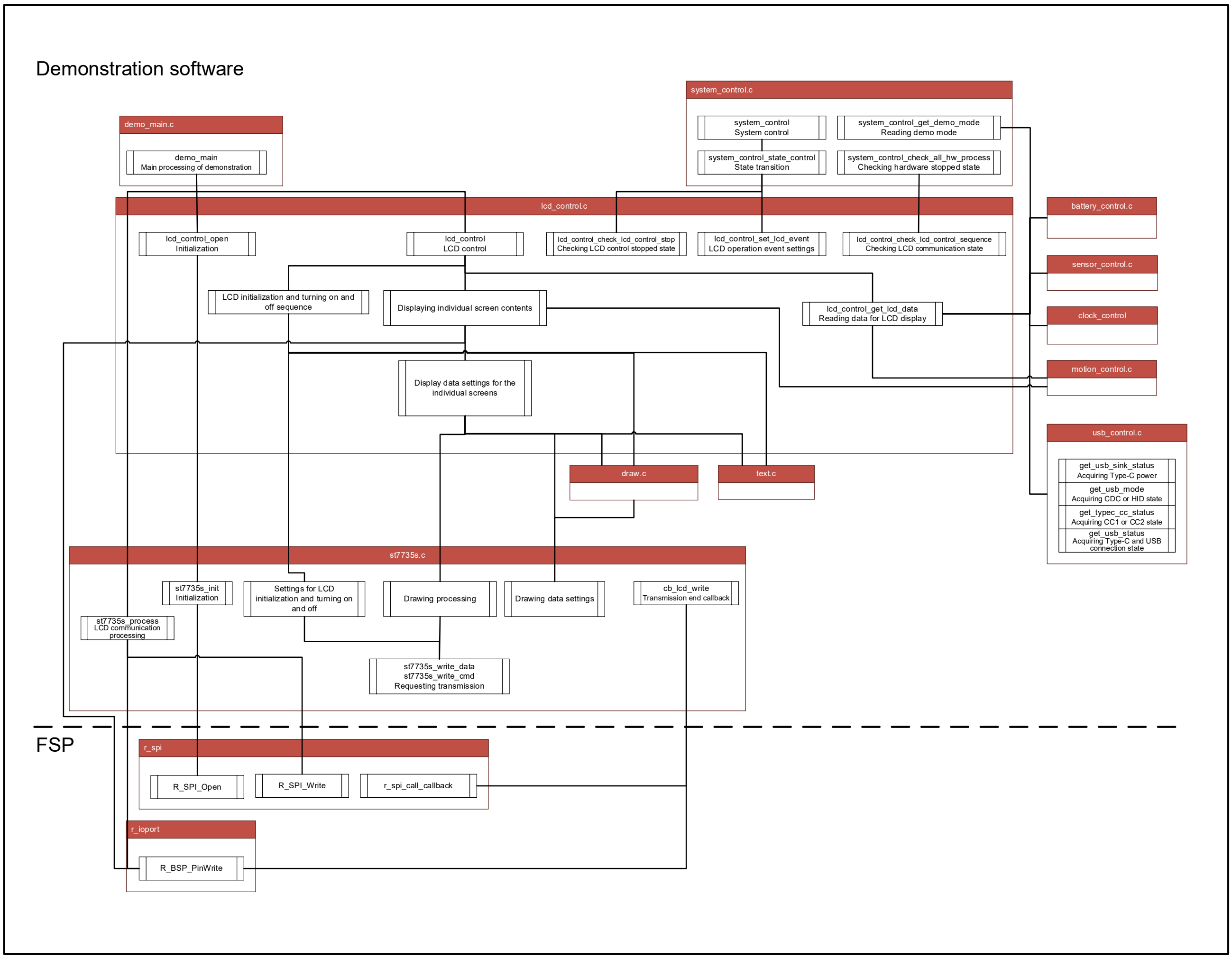


Figure 3-13 Structure of the LCD Display Processes

3.10.2 State Transitions

Figure 3-14 shows the state transitions for the display on the LCD. The screen contents to be displayed are switched by reception of a switch input notification from the system control process and of changes in the connection state from the USB control process. Some transitions include branches that depend on the state of the system control or date and time setting.

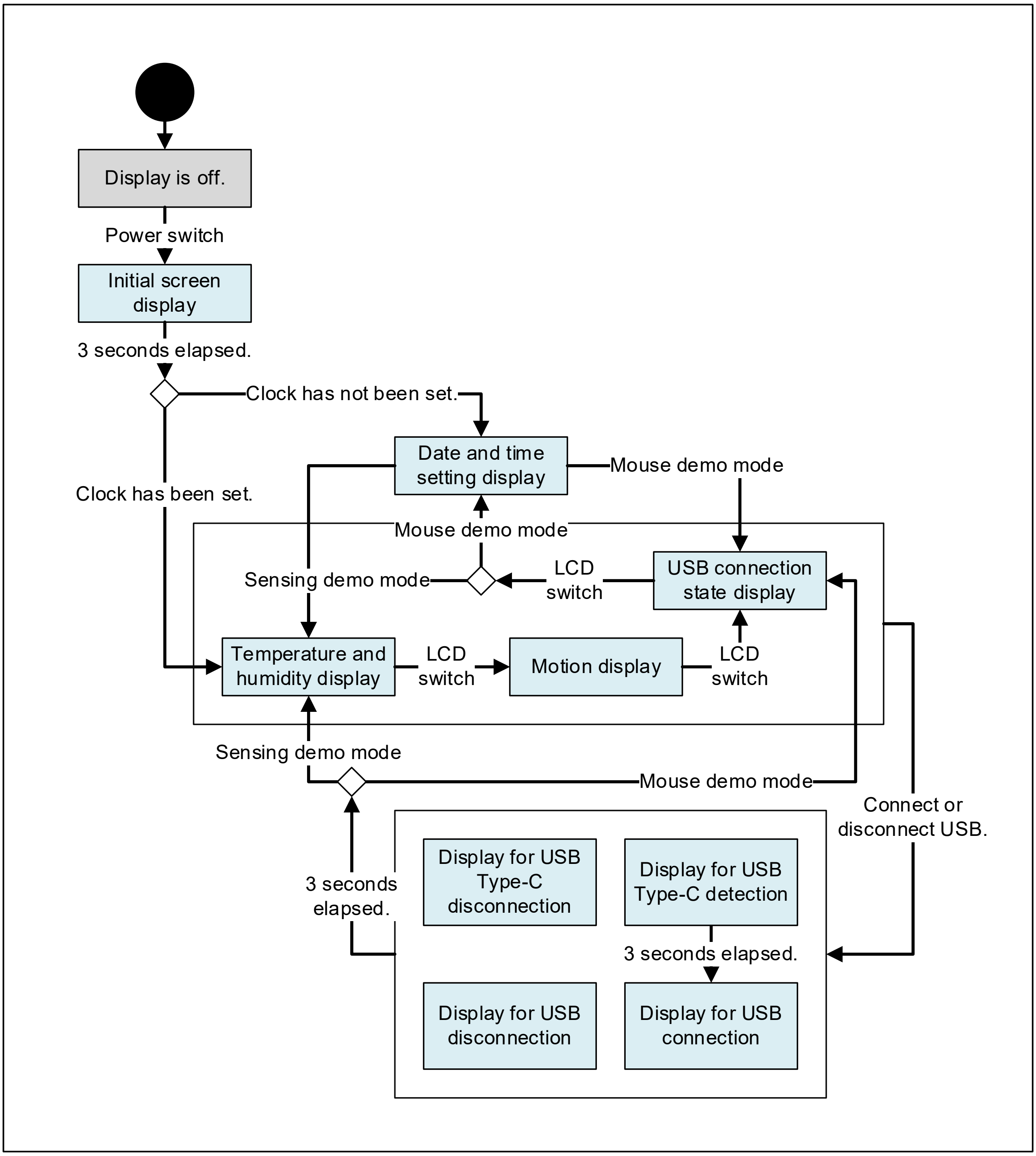


Figure 3-14 State Transitions for LCD Display

3.10.3 Overview of the Configuration of LCD Display

Six LCD configurations are defined to suit the various demonstration operations. Figure 3-15 is an overview of the configurations of LCD display. The meanings of the numbers are described in the figures on the following pages.

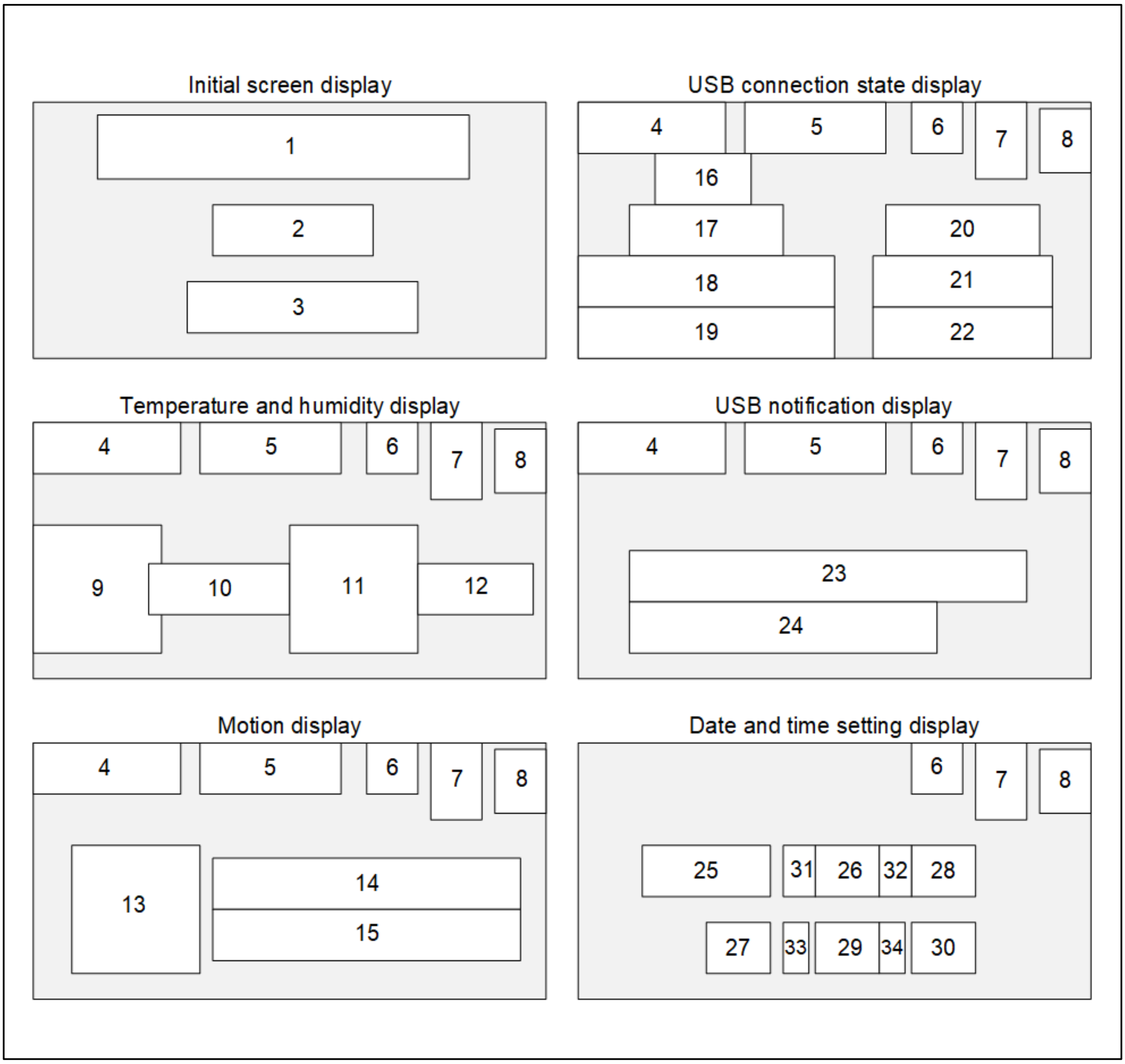
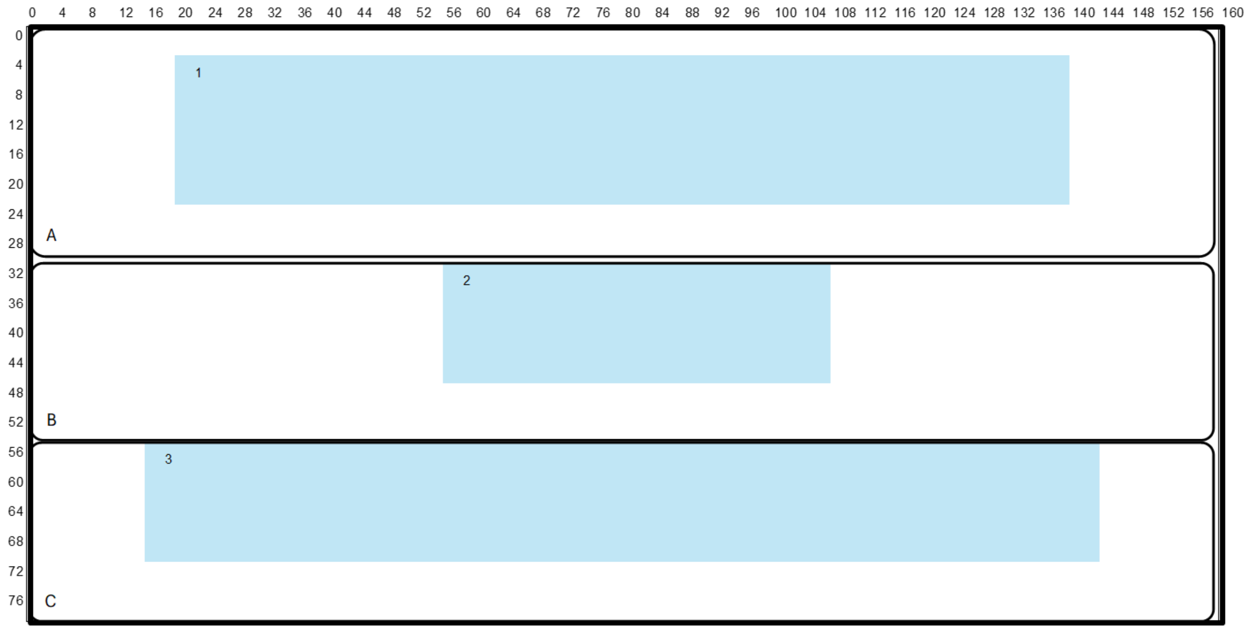


Figure 3-15 Configurations of LCD Display

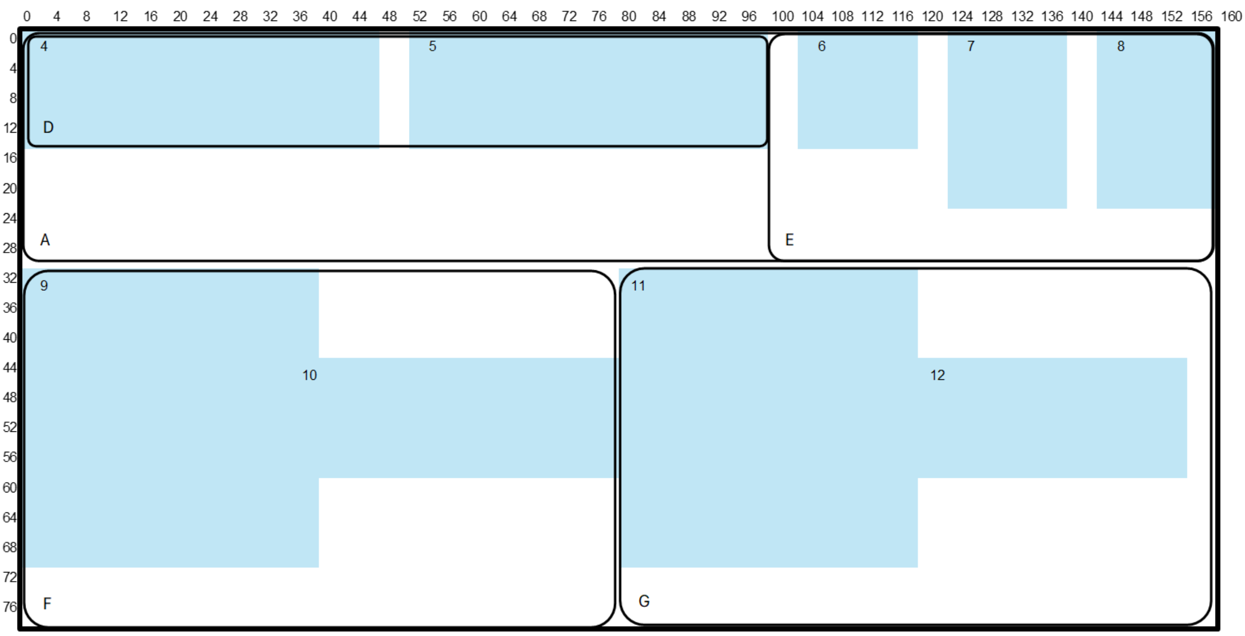
3.10.4 LCD Drawing Areas and Icon Display Locations

In each of the six screen display configurations, the LCD screen is divided into multiple display update areas to optimize the time for LCD drawing. The LCD drawing areas and icon display locations are shown in Figure 3-16 to Figure 3-21.



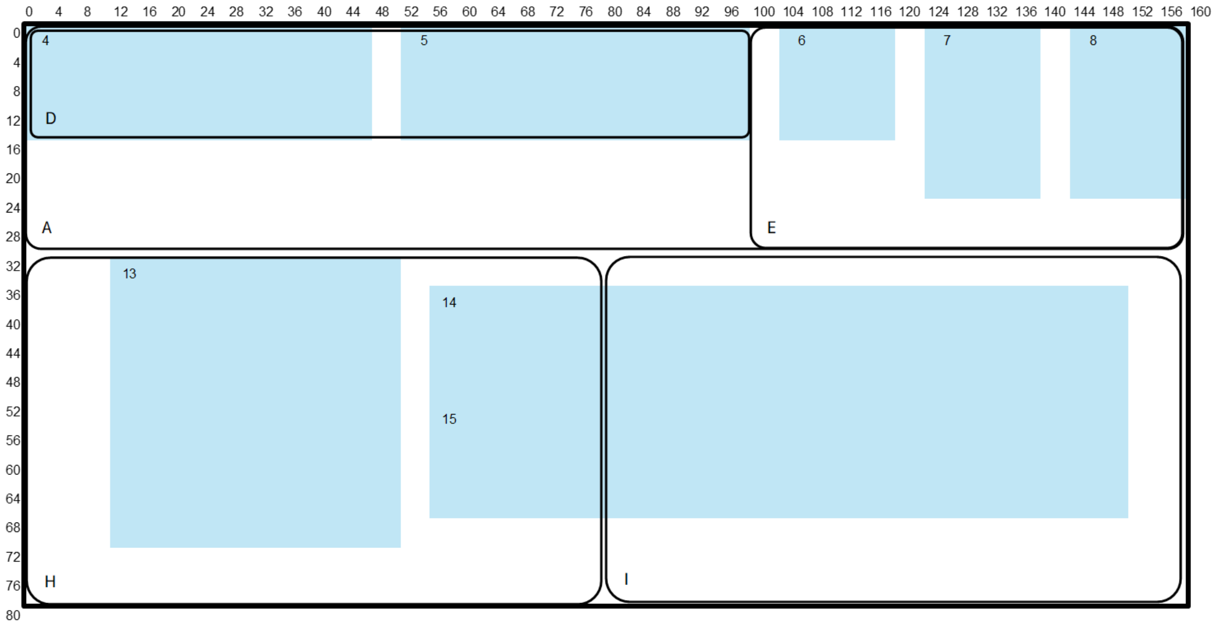
Icon			X	Y	Width	Height	
1	Logo icon	LogoImage	20	4	116	20	
2	RA2L2	McuText	56	32			
3	Demonstration kit	BoardText	18	56			
Drawing Area			X	Y	Width	Height	Size
A	Top	TopCanvas	0	0	160	32	10240
B	Center	UpperCanvas	0	32	160	24	7680
C	Bottom	LowerCanvas	0	56	160	24	7680

Figure 3-16 LCD Drawing Areas and Icon Display Locations (Initial Screen Display)



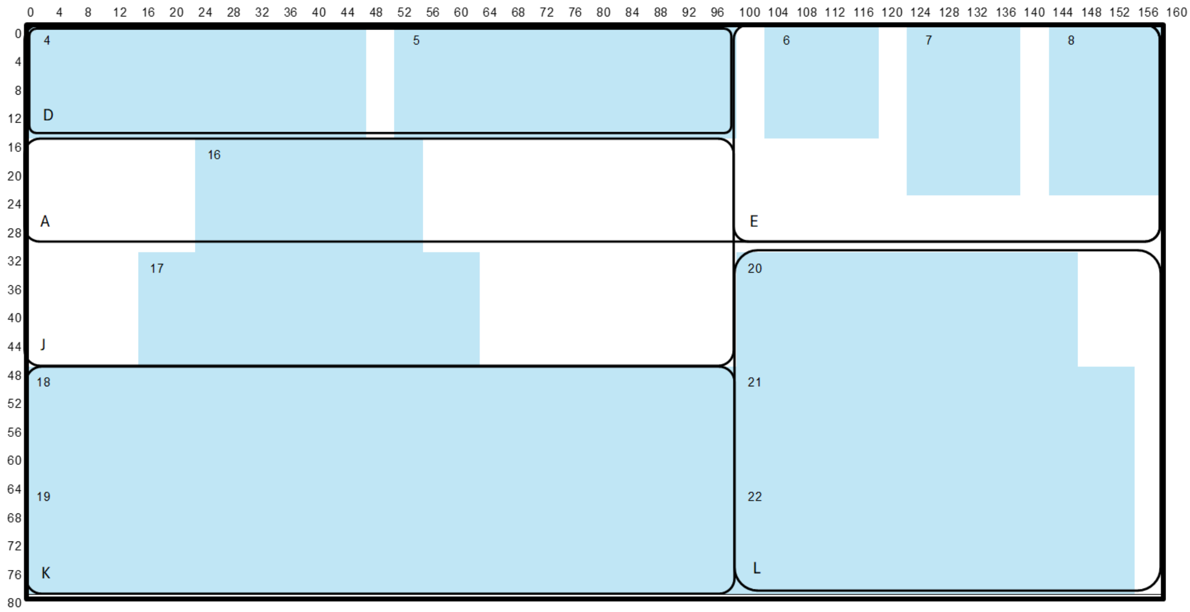
Icon			X	Y	Width	Height	
4	MM/DD	DateText	0	0			
5	hh:mm	TimeText	52	0			
6	Demo mode icon	ModeImage	104	0	16	16	
7	Charge icon	ChargeImage	124	0	16	24	
8	Remaining level icon	BatteryImage	144	2	16	20	
9	Temperature icon	TempImage	0	32	40	40	
10	Temperature	TempText	36	44			
11	Humidity icon	HumidImage	80	32	40	40	
12	Humidity	HumidText	120	44			
Drawing Area			X	Y	Width	Height	Size
A	Screen contents switching	TopCanvas	0	0	160	32	10240
D	Date and time updating	ClockCanvas	0	0	100	16	3200
E	Battery updating	BatteryCanvas	100	0	60	32	3840
F	Temperature	LeftCanvas	0	16	80	64	10240
G	Humidity	RightCanvas	80	32	80	48	7680

Figure 3-17 LCD Drawing Areas and Icon Display Locations (Temperature and Humidity Display)



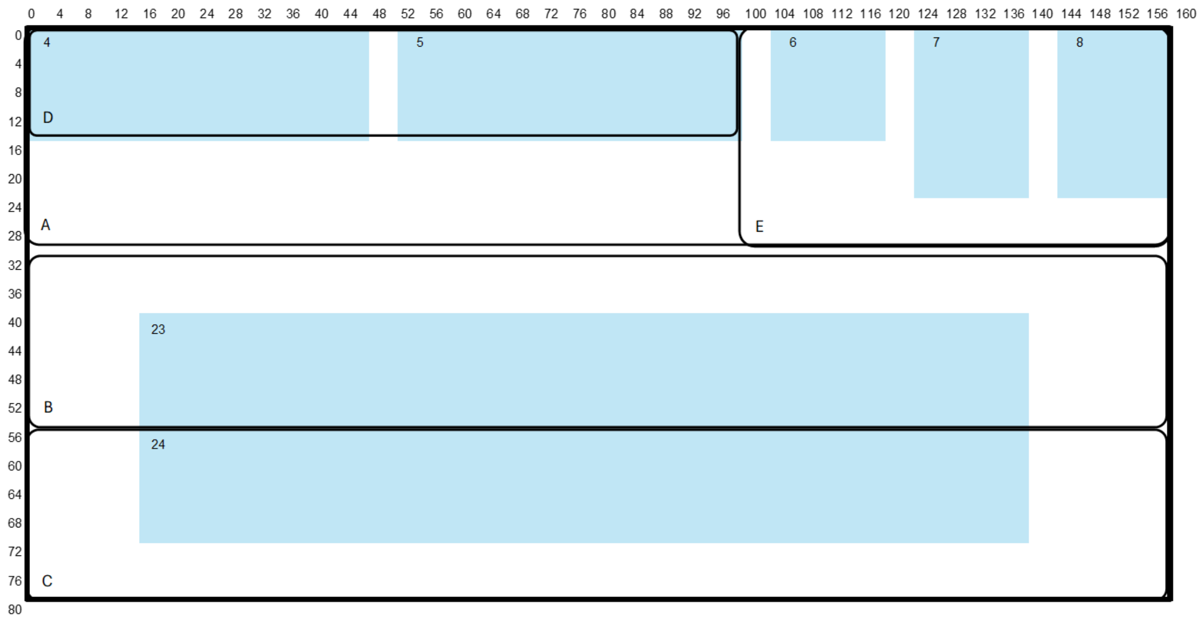
Icon			X	Y	Width	Height	
4	MM/DD	DateText	0	0			
5	hh:mm	TimeText	52	0			
6	Demo mode icon	ModeImage	104	0	16	16	
7	Charge icon	ChargeImage	124	0	16	24	
8	Remaining level icon	BatteryImage	144	2	16	20	
13	Motion icon	MotionImage	12	32	40	40	
14	Motion text 1	MotionText1	56	36			
15	Motion text 2	MotionText2	56	52			
Drawing Area			X	Y	Width	Height	Size
A	Screen contents switching	TopCanvas	0	0	160	32	10240
D	Date and time updating	ClockCanvas	0	0	100	16	3200
E	Battery updating	BatteryCanvas	100	0	60	32	3840
H	Motion display left	MotionCanvasL	0	32	56	48	5376
I	Motion display right	MotionCanvasR	56	32	104	48	9984

Figure 3-18 LCD Drawing Areas and Icon Display Locations (Motion Display)



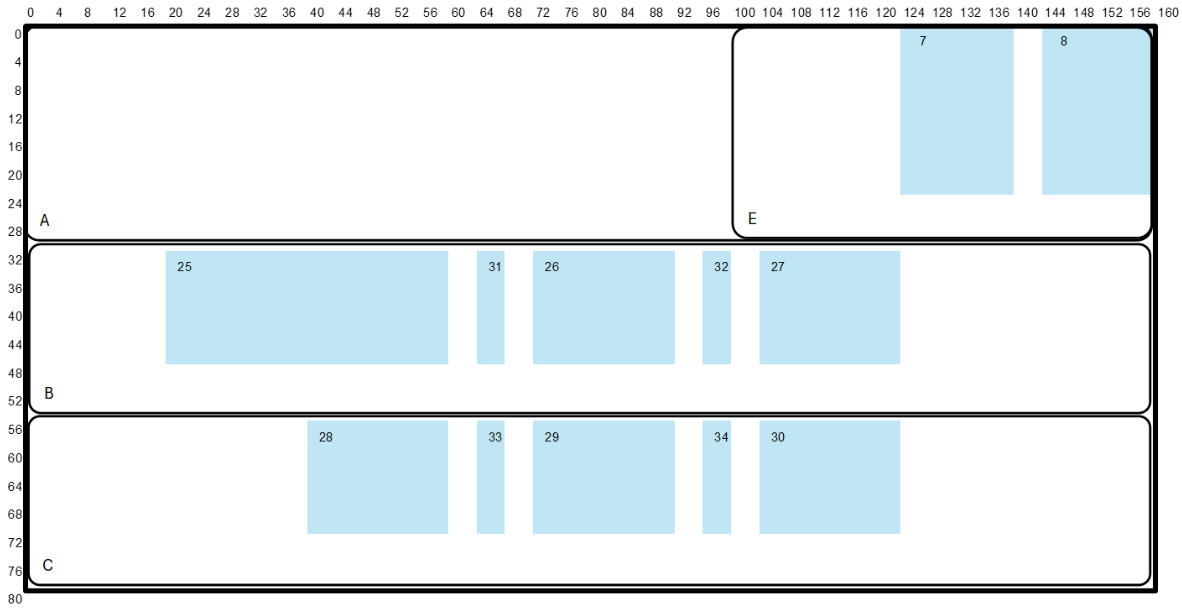
Icon			X	Y	Width	Height	
4	MM/DD	DateText	0	0			
5	hh:mm	TimeText	52	0			
6	Demo mode icon	ModeImage	104	0	16	16	
7	Charge icon	ChargeImage	124	0	16	24	
8	Remaining level icon	BatteryImage	144	2	16	20	
16	CC1 and CC2	CcText	24	16			
17	Type-C icon	TypecImage	16	32	48	16	
18	Type-C state 1	TypecText1	0	48			
19	Type-C state 2	TypecText2	0	64			
20	USB icon	UsbImage	100	32	48	16	
21	USB state 1	UsbText1	100	48			
22	USB state 2	UsbText2	100	64			
Drawing Area			X	Y	Width	Height	Size
A	Screen contents switching	TopCanvas	0	0	160	32	10240
D	Date and time updating	ClockCanvas	0	0	100	16	3200
E	Battery updating	BatteryCanvas	100	0	60	32	3840
J	Type-C upper	UsbLeftCanvas1	0	16	100	32	6400
K	Type-C lower	UsbLeftCanvas2	0	48	100	32	6400
L	USB	UsbRightCanvas	100	32	60	48	5760

Figure 3-19 LCD Drawing Areas and Icon Display Locations (USB Connection State Display)



Icon			X	Y	Width	Height	Size
4	MM/DD	DateText	0	0			
5	hh:mm	TimeText	52	0			
6	Demo mode icon	ModeImage	104	0	16	16	
7	Charge icon	ChargeImage	124	0	16	24	
8	Remaining level icon	BatteryImage	144	2	16	20	
23	Notification 1	InfoTextUpper	16	40			
24	Notification 2	InfoTextLower	16	56			
Drawing Area			**X**	**Y**	**Width**	**Height**	**Size**
A	Screen contents switching	TopCanvas	0	0	160	32	10240
D	Date and time updating	ClockCanvas	0	0	100	16	3200
E	Battery updating	BatteryCanvas	100	0	60	32	3840
B	Notification upper	UpperCanvas	0	32	160	24	7680
C	Notification lower	LowerCanvas	0	56	160	24	7680

Figure 3-20 LCD Drawing Areas and Icon Display Locations (USB Notification Display)



Icon			X	Y	Width	Height	
7	Charge icon	ChargeImage	124	0	16	24	
8	Remaining level icon	BatteryImage	144	2	16	20	
25	Year	YearText	20	32			
26	Month	MonthText	74	32			
27	Date	DayText	104	32			
28	Hour	HourText	40	56			
29	Minute	MinuteText	72	56			
30	Second	SecText	104	56			
31	Left slash	SlaLText	64	32			
32	Right slash	SlaRText	94	32			
33	Left colon	ColLText	64	56			
34	Right colon	ColRText	94	56			
Drawing Area			X	Y	Width	Height	Size
A	Screen contents switching	TopCanvas	0	0	160	32	10240
E	Battery updating	BatteryCanvas	100	0	60	32	3840
B	Center	UpperCanvas	0	32	160	24	7680
C	Bottom	LowerCanvas	0	56	160	24	7680

Figure 3-21 LCD Drawing Areas and Icon Display Locations (Date and Time Setting Display)

3.10.5 Flowchart of LCD Drawing

Figure 3-22 shows a flowchart of control for the temperature and humidity display shown in Figure 3-17 as an example of the procedure for LCD drawing.

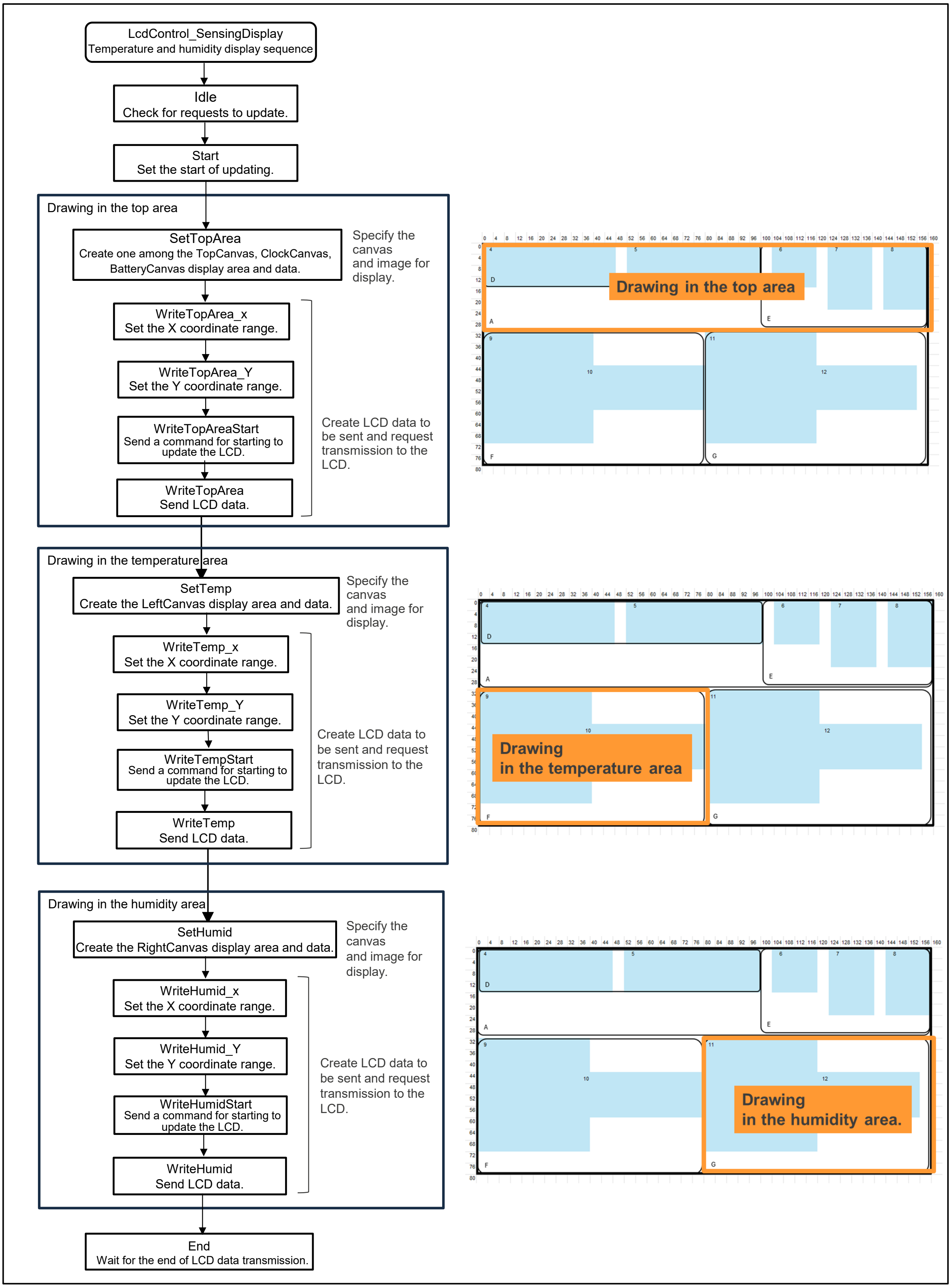


Figure 3-22 Flowchart of LCD Drawing for Temperature and Humidity Display

3.10.6 How to Create Data for LCD Display

The following describes how to create data to be displayed on the LCD.

1. Prepare icon image data to be displayed.
   The sample code uses the image data shown in Table 3-8.
2. Use any image data conversion tool or Web site to convert the image data prepared in step 1 to C source data.
3. Create a C source file of the converted C source data. The images.c file contains the converted C source data for use in the sample code (see Figure 3-23).
4. The LCD driver functions will send the data of the individual images from images.c to the LCD module through an SPI and thus draw the images.

```
/***********************************************************************************
 * File Name    : images.c
 * Description  : Contains function definitions.
 ***********************************************************************************/
/***********************************************************************************
* Copyright (c) 2025 Renesas Electronics Corporation and/or its affiliates
*
* SPDX-License-Identifier: BSD-3-Clause
***********************************************************************************/


#include <stdint.h>

/** @brief Image converted with: http://javl.github.io/image2cpp/
 * Image sourced from: internal Renesas markcom.
 */
const uint16_t g_renesas_title[] = {
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xF79E,
                0xE73E, 0xDEDD, 0xD6BD, 0xD6BD, 0xDEDD, 0xE71E, 0xF79F, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
```

Figure 3-23 images.c File

3.10.7 Icons to be Displayed

Table 3-8 shows the icon image data used in the sample code.

Table 3-8 Icon Image Data

Item No.	Icon	X	Y	Width	Height	Remark
1		20	4	116	20	Renesas logo
6	S	104	0	16	16	Sensing demo mode
	M					Mouse demo mode
7		124	0	16	24	Mark to indicate charging
8		144	2	16	20	Remaining battery level 0
						Remaining battery level 1
						Remaining battery level 2
						Remaining battery level 3
						Remaining battery level 4
						Remaining battery level 5
9		0	32	40	40	Temperature icon
11		80	32	40	40	Humidity icon
13		12	32	40	40	Tap detection
						Tilt detection
						Walk
						Run
						Board raised
						Board horizontal
17		16	32	48	16	Type-C is not detected. An oblique red line is placed over the icon.
						CC1
						CC2
20		100	32	48	16	
	No USB			48	16	An oblique red line is placed over the USB icon.

3.10.8 Character Strings to be Displayed

Table 3-9 shows the character strings used in the sample code.

Table 3-9 Character String Data

Item No.	Character String	X	Y	Width	Height	Remark
2	RA2L2	56	32	50	16	
3	USB-C Demo Kit	18	56	124	16	
4	01/01 to 12/31	0	0	50	16	
5	00:00 to 23:59	52	0	43	16	
10	-40°C to 125°C	36	44	44	16	Placed in the white space of the temperature icon.
12	100%	120	44	36	16	
14	No Motion	56	36	81	32	
	TAP			30		
	TILT			36		
	WALKING			66		
	RUNNING			66		
	BOARD			50		Not used
15	Detection	56	52	86	16	
	RAISE			46		Not used
	HORIZONTAL			96		
16	CC1	24	16	26	16	
	CC2			30		
18	Type-C	0	48	58	16	
	5V/3A			50		
	5V/1.5A			59		
	5V/default			100		
19	NoCharge	0	64	80	16	
	Source			60		
21	No USB	100	48	55	16	
	CDC			30		
	HID			26		
22	Device	100	64	56	16	
23	Type-C Attach	16	40	123	16	
	USB Connect			105		
	Type-C			58		
	Type-C / USB			108		
24	Success!	16	56	73	16	
	Disconnect			96		
25	2000 to 2099	20	32	40	16	Each item is highlighted while selected and blinks at 1-Hz intervals during setting.
26	01 to 12	72	32	20	16	
27	01 to 31	104	32	20	16	
28	00 to 23	40	56	20	16	
29	00 to 59	72	56	20	16	
30	00 to 59	104	56	20	16	
31	/	64	32	10	16	
32		94				
33	:	64	56	4	16	
34		94				

3.11 Temperature and Humidity Sensor Control

The reference design uses the HS4001 temperature and humidity sensor made by Renesas. The HS400X driver in the FSP is used without change in the sample code to control the sensor.

The results of measurement are acquired from the temperature and humidity sensor at 1-second intervals. Other process blocks can read each of the integer and fractional parts of the results in the int8_t type. Only the latest results from the sensor can be read.

3.11.1 Processing Structure

Figure 3-24 shows the structure of the processes of temperature and humidity sensor control.

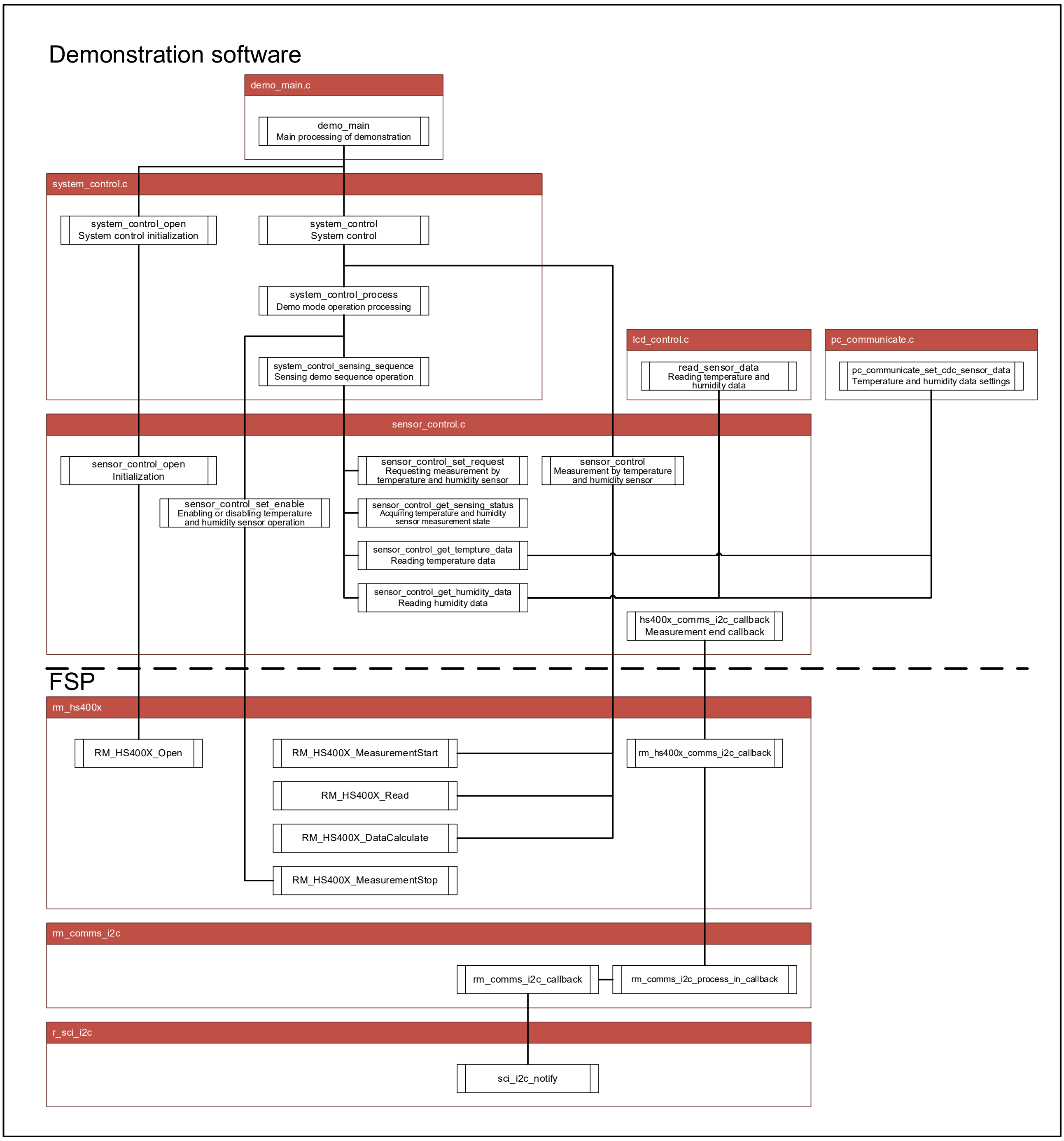


Figure 3-24 Structure of the Temperature and Humidity Sensor Processes

3.12 Motion Sensor Control

The reference design uses the ICM-42688-P motion sensor made by TDK Corporation. This sensor has the APEX Motion function, which can detect several types of motion such as taps or tilts. The sample code displays the detected motions on the LCD. Table 3-10 lists the types of motion that can be detected by the sample code.

Table 3-10 Types and Definitions of Motions

Motion Name	Definition of the Motion to be Detected
Tap	This motion is detected when the data acquired from the accelerometer have exceeded the threshold value that is specified for the tap motion*1.
Tilt	This motion is detected when the angle and duration of inclination acquired from the angular velocity sensor have exceeded the specified threshold values. The threshold value for the angle is set to 35 degrees (by default) and that for the duration is set to 4 seconds (by default) in the demo board.
Walk	This motion is detected when the data acquired form the accelerometer have exceeded the threshold value that is specified for the walk motion*1.
Run	This motion is detected when the data acquired form the accelerometer have exceeded the threshold value that is specified for the run motion*1.

Note 1. All threshold values for motion detection in the sample code are set to the default values of the ICM-42688-P. For details of the threshold values, refer to the specifications of the ICM-42688-P.

3.12.1 Processing Structure

Figure 3-25 shows the structure of the motion sensor control processes.

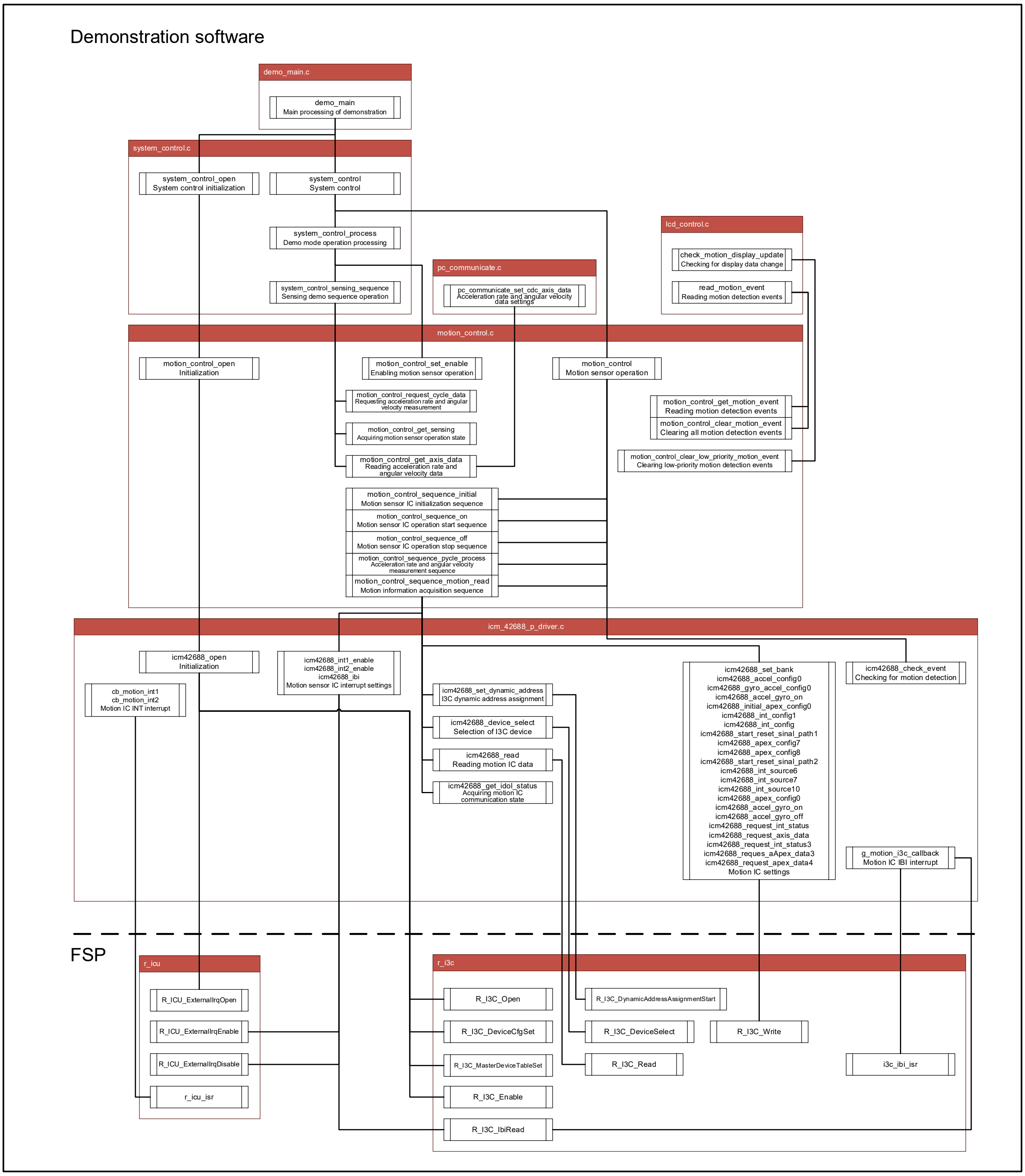


Figure 3-25 Structure of the Motion Sensor Control Processes

3.12.2 State Transitions

The motion sensor is only used in sensing demo mode, where it periodically detects the acceleration rate and angular velocity. The INT1 pin is used to indicate motion detection by default and motion information is acquired when a motion is detected. Figure 3-26 shows the state transitions.

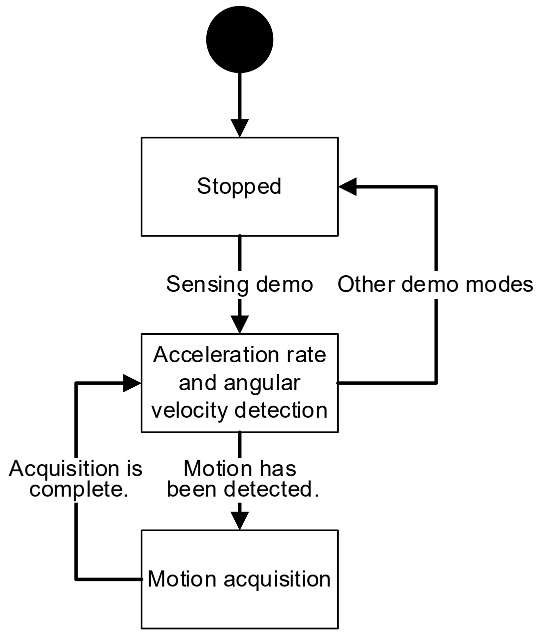


Figure 3-26 State Transitions for Motion Sensor Control

Note: The pin used for motion detection is specified by the MOTION_INT constant, which is defined in icm_42688_p_driver.h of the sample code. The INT1 pin is specified by default but the pin can be changed to INT2 or IBI.

3.12.3 Measurement of the Acceleration Rate and Angular Velocity

The results of measuring acceleration rate and angular velocity are acquired at 1-second intervals in sensing demo mode. The acceleration rate and angular velocity data for the last five seconds are retained and the latest results and the maximum values of the results measured over that time can be read. Table 3-11 shows the specifications of the registers for storing the results to be acquired from the sensor.

Table 3-11 Sensor Acquisition Result Registers

Register in the IC	Address	Setting Data Name	Remark
ACCEL_DATA_X1	0x1F	ACCEL_DATA_X	X-axis acceleration rate
ACCEL_DATA_X0	0x20		
ACCEL_DATA_Y1	0x21	ACCEL_DATA_Y	Y-axis acceleration rate
ACCEL_DATA_Y0	0x22		
ACCEL_DATA_Z1	0x23	ACCEL_DATA_Z	Z-axis acceleration rate
ACCEL_DATA_Z0	0x24		
GYRO_DATA_X1	0x25	GYRO_DATA_X	X-axis angular velocity
GYRO_DATA_X0	0x26		
GYRO_DATA_Y1	0x27	GYRO_DATA_Y	Y-axis angular velocity
GYRO_DATA_Y0	0x28		
GYRO_DATA_Z1	0x29	GYRO_DATA_Z	Z-axis angular velocity
GYRO_DATA_Z0	0x2A		

3.12.4 Detection of Motion Information

The INT interrupt from the motion sensor indicates motion detection and motion information is acquired in response to its detection in sensing demo mode. If detection of motion information and measurement of the acceleration rate and angular velocity occur at the same time, measurement of the acceleration rate and angular velocity takes priority because detection of motion information may occur multiple times in succession but the acceleration rate and angular velocity are measured periodically. Table 3-12 lists the specifications of the motion information registers.

Table 3-12 Motion Information Registers

Register in the IC	Address	Setting Data Name	Remark
APEX_DATA3	0x34	ACTIVITY_CLASS	Pedometer
APEX_DATA4	0x35	TAP_NUM	Tap detection
INT_STATUS3	0x38	STEP_DET_INT	Pedometer
		TILT_DET_INT	Tilt detection
		TAP_DET_INT	Tap detection

If several types of motion are detected, the priority is determined as listed in Table 3-13.

Table 3-13 Priority of Detected Motions

Motion	Priority	Remark
Walk	1	After a motion is displayed on the LCD, the results of detection of the other lower-priority motions are cleared.
Run	2	
Tap detection	3	
Tilt detection	4	

3.12.5 Register Settings

Table 3-14 shows the settings of the registers in the motion sensor. If the INT2 pin or IBI interrupt is to be used in motion detection, see the settings of INT_SOURCE7 or INT_SOURCE10 in the table. For the terms used with the motion sensor and the details of the specifications of the sensor, refer to the specifications of the ICM-42688-P.

Table 3-14 Motion Sensor Setting Registers

Register in the IC	Bank	Address	Setting Data Name	Setting	Remark
INT_CONFIG	0	0x14	—	00	
			INT2_MODE	0	Default
			INT2_DRIVE_CIRCUIT	0	Default
			INT2_POLARITY	0	Default
			INT1_MODE	0	Default
			INT1_DRIVE_CIRCUIT	0	Default
			INT1_POLARITY	0	Default
SIGNAL_PATH_RESET	0	0x4B	—	0	
			DMP_INIT_EN	1	DMP enabled
				0	DMP disabled
			DMP_MEM_RESET	1	DMP memory reset
				0	Default
			—	0	
			ABORT_AND_RESET	0	Default
			TMST_STROBE	0	Default
			FIFO_FLUSH	0	Default
			—	0	
PWR_MGMT0	0	0x4E	—	00	Reserved
			TEMP_DIS	0	Temperature sensor off
			IDLE	0	Default
			GYRO_MODE	11	LN mode
				00	Off
			ACCEL_MODE	10	LP mode
				00	Off
ACCEL_CONFIG0	0	0x50	ACCEL_FS_SEL	000	Default
			—	0	Reserved
			ACCEL_ODR	0111	200 Hz
GYRO_ACCEL_CONFIG0	0	0x52	ACCEL_UI_FILT_BW	0100	
			GYRO_UI_FILT_BW	0001	Default
APEX_CONFIG0	0	0x56	DMP_POWER_SAVE	0	
			TAP_ENABLE	1	Enabled
				0	Disabled

Register in the IC	Bank	Address	Setting Data Name	Setting	Remark
			PED_ENABLE	1	Enabled
				0	Disabled
			TILT_ENABLE	1	Enabled
				0	Disabled
			R2W_EN	1	Enabled
				0	Disabled
			—	0	
			DMP_ODR	10	50 Hz
INT_CONFIG1	0	0x64	—	0	
			INT_TPULSE_DURATION	0	Default
			INT_TDEASSERT_DISABLE	0	Default
			INT_ASYNC_RESET	0	INT1 and INT2 are used.
			—	0000	
INT_SOURCE6	4	0x4D	—	00	
			STEP_DET_INT1_EN	1	INT1 enabled
			STEP_CNT_OFL_INT1_EN	0	Default
			TILT_DET_INT1_EN	1	INT1 enabled
			WAKE_DET_INT1_EN	1	INT1 enabled
			SLEEP_DET_INT1_EN	1	INT1 enabled
			TAP_DET_INT1_EN	1	INT1 enabled
INT_SOURCE7	4	0x4E	—	00	
			STEP_DET_INT2_EN	1	INT2 enabled
			STEP_CNT_OFL_INT2_EN	0	Default
			TILT_DET_INT2_EN	1	INT2 enabled
			WAKE_DET_INT2_EN	1	INT2 enabled
			SLEEP_DET_INT2_EN	1	INT2 enabled
			TAP_DET_INT2_EN	1	INT2 enabled
INT_SOURCE10	4	0x51	—	00	
			STEP_DET_IBI_EN	1	IBI enabled
			STEP_CNT_OFL_IBI_EN	0	Default
			TILT_DET_IBI_EN	1	IBI enabled
			WAKE_DET_IBI_EN	1	IBI enabled
			SLEEP_DET_IBI_EN	1	IBI enabled
			TAP_DET_IBI_EN	1	IBI enabled

3.13 Data Control

The sample code stores the data measured by the temperature and humidity sensor and motion sensor in data flash memory at 5-second intervals.

For the temperature and humidity data, the values measured within one second after the latest storage timing are stored. For the motion data, the maximum values of the results measured over five seconds are stored. The date and time of storage are appended to the stored data.

The stored data can be read in order from oldest to latest. Once a value is read, the value cannot be read again. If the date and time setting is changed, all stored data are erased.

3.13.1 Processing Structure

Figure 3-27 shows the structure of the data control processes.

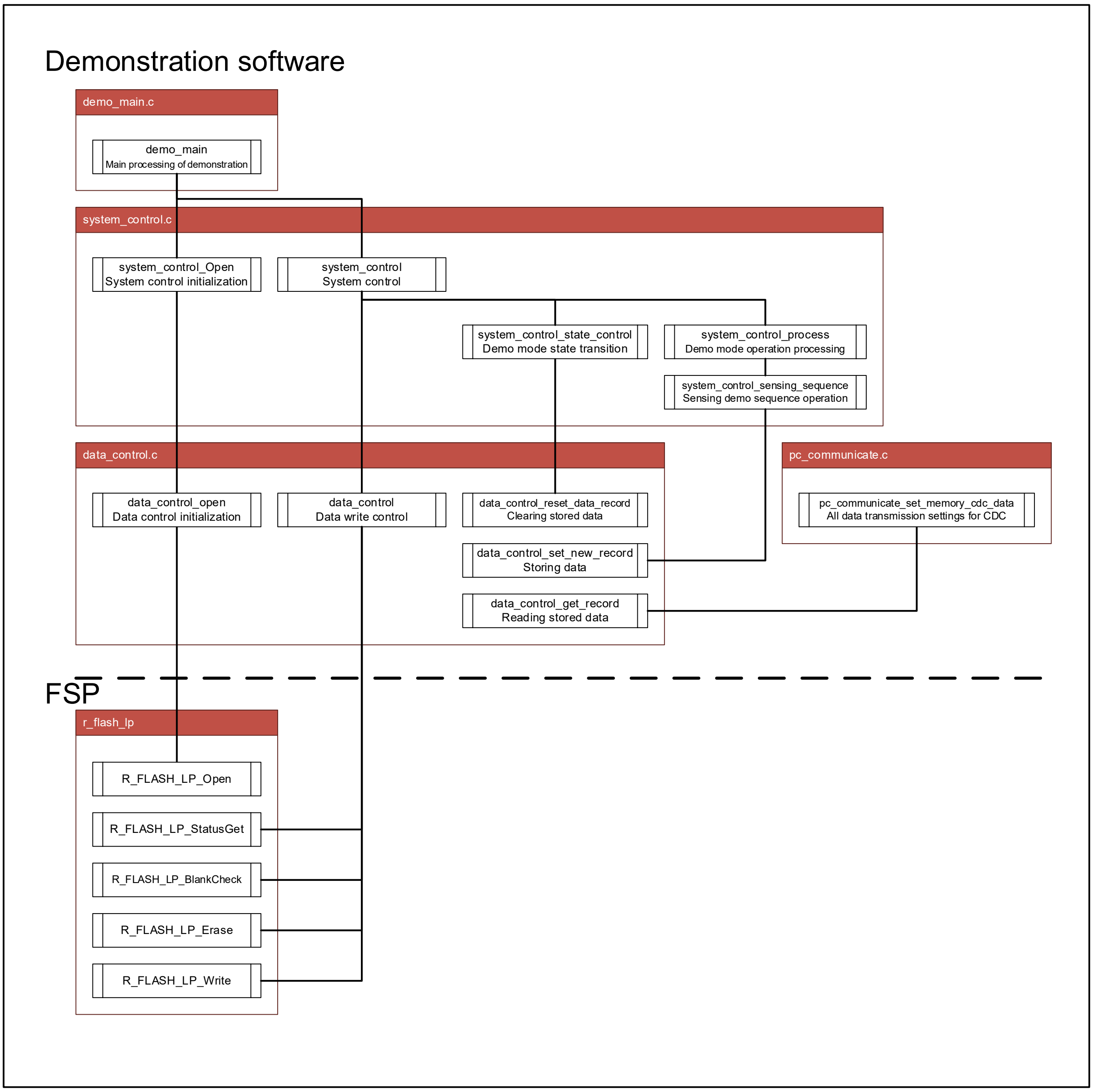


Figure 3-27 Structure of the Data Control Processes

3.13.2 Format of Data Records to be Stored

Table 3-15 shows the format of the data records to be stored.

Table 3-15 Format of the Data Records to be Stored

Stored Data Item	Data Type	Data Length	Data Range	Remark
Date (year)	uint8_t	1	0x00 to 0x99	BCD
Date (month)	uint8_t	1	0x01 to 0x12	BCD
Date (day)	uint8_t	1	0x01 to 0x31	BCD
Date (hour)	uint8_t	1	0x00 to 0x23	BCD
Date (minute)	uint8_t	1	0x00 to 0x59	BCD
Date (second)	uint8_t	1	0x00 to 0x59	BCD
Temperature (integer part)	int8_t	1	-40 to 125	
Temperature (fractional part)	int8_t	1	-99 to 99	
Humidity (integer part)	int8_t	1	0 to 100	
Humidity (fractional part)	int8_t	1	0 to 99	
Motion (ACCEL-X)	int16_t	2	-32766 to 32767	
Motion (ACCEL-Y)	int16_t	2	-32766 to 32767	
Motion (ACCEL-Z)	int16_t	2	-32766 to 32767	
Motion (GYRO-X)	int16_t	2	-32766 to 32767	
Motion (GYRO-Y)	int16_t	2	-32766 to 32767	
Motion (GYRO-Z)	int16_t	2	-32766 to 32767	

3.13.3 Allocation of Records in Data Flash Memory

Records are stored in data flash memory such that they do not overlap block boundaries. Table 3-16 shows the allocation of records in data flash memory.

Table 3-16 Record Allocation in Data Flash Memory

Address	Block	Record ID	Remark
0x000	0	0	
0x016		1	
:		:	
0x3DE		45	
0x3F4			Unused space
0x400	1	46	
0x416		47	
:		:	
0x7DE		91	
0x7F4			Unused space
0x800	2	92	
0x816		93	
:		:	
0xBDE		137	
0xBF4			Unused space
0xC00	3	138	
0xC16		139	
:		:	
0xCDE		183	
0xCF4			Unused space

3.13.4 Record Control

Before a record is stored, if the record will be the first record of the block, the block is checked to see whether it is empty. If the block is not empty, the block is erased. Figure 3-28 is an overview of the record control operation.

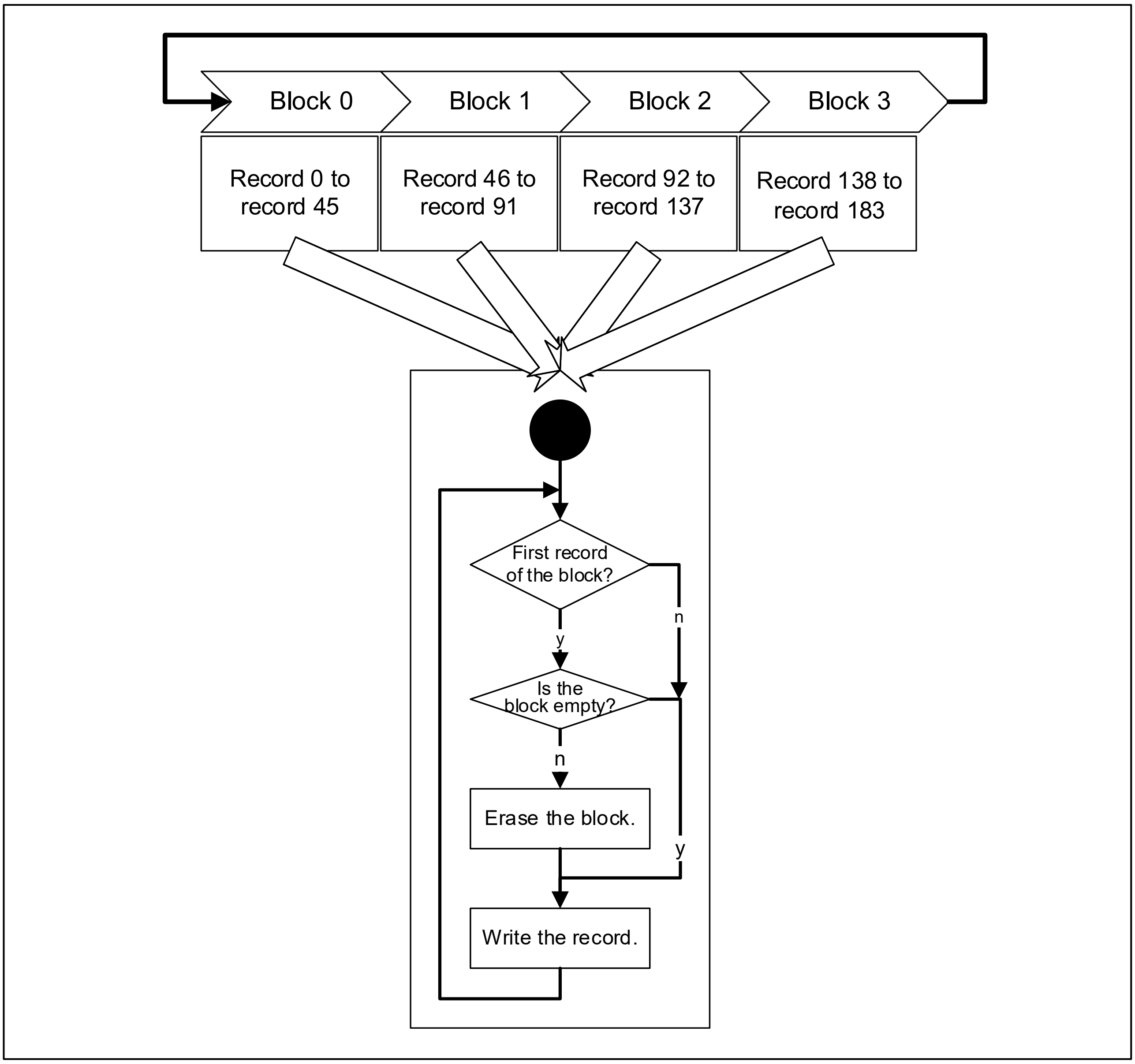


Figure 3-28 Overview of Record Control Operation

3.13.5 Control of Read Operations

Stored data are read in order from oldest to latest every time a read operation is requested. The state of whether each stored record has been read is controlled and a record that has been read is not read again.

If the latest record has already been read, a response indicating that there are no data to be read is returned. If the block that includes the oldest record is to be erased before a new record is stored, the oldest record in the next block that is not to be erased is handled as the oldest data.

Figure 3-29 is an overview of the control of read operations.

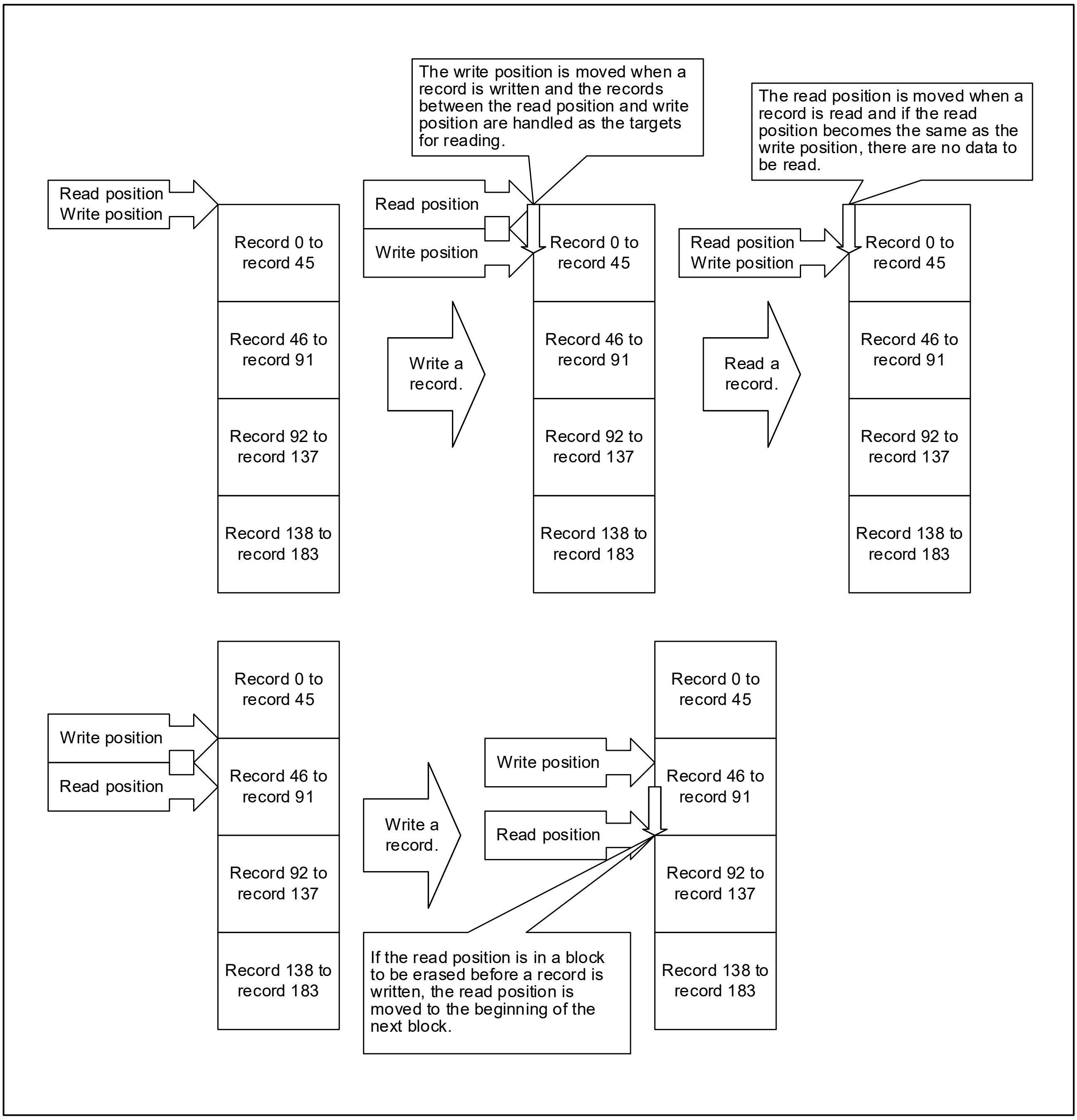


Figure 3-29 Overview of Control of Read Operations

3.14 Communications with the PC Application

The demonstration software uses CDC in communicating with the demonstration PC application in sensing demo mode. It sends HID reports to the PC application in mouse demo mode. Communications with the PC do not proceed in standby mode or during date and time setting.

3.14.1 Processing Structure

Figure 3-30 shows the processing structure for communications with the PC application.

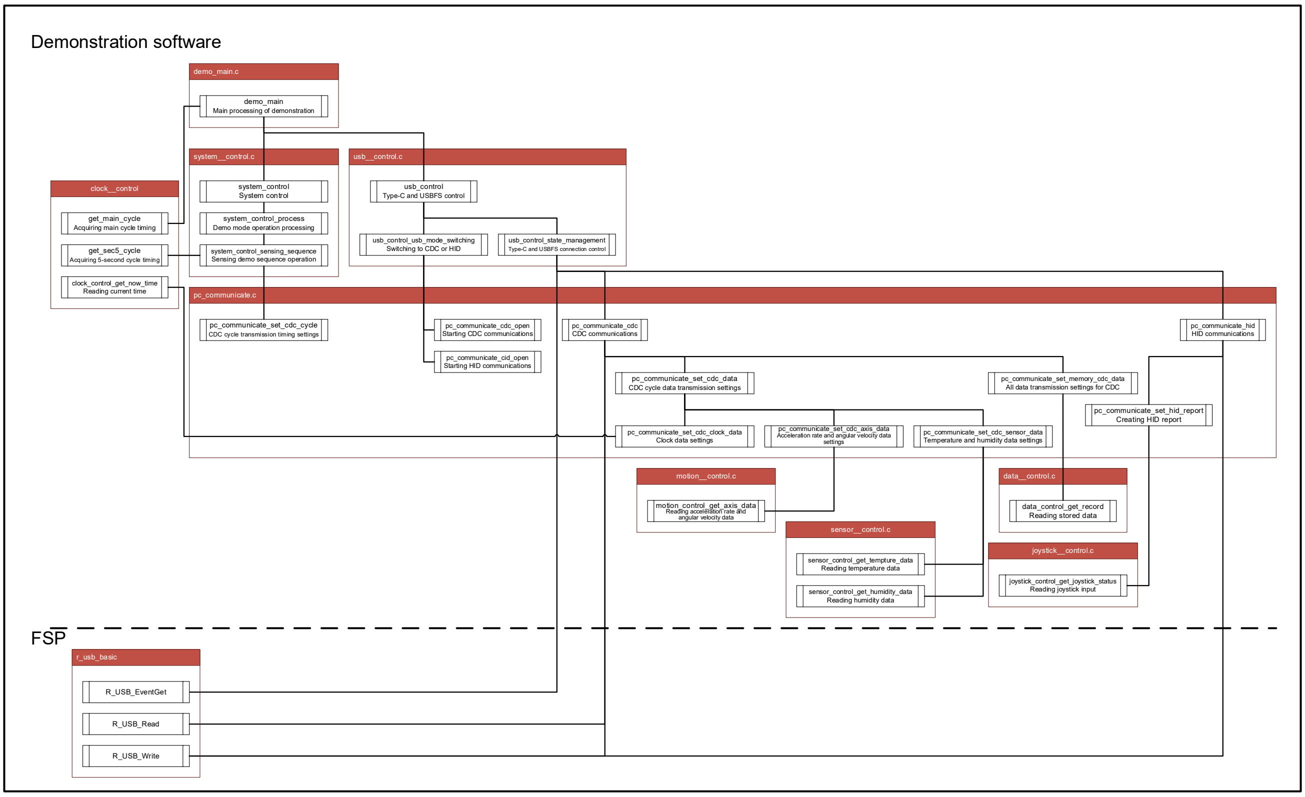


Figure 3-30 Processing Structure for Communications with the PC Application

3.14.2 CDC Communications

After reception of a communications start command, data are transmitted at 5-second intervals until reception of a communications stop command. If an all data read command is received, data are sent in the order of reading until there are no data to be read according to the data control process. Figure 3-31 shows the state transitions for CDC communications.

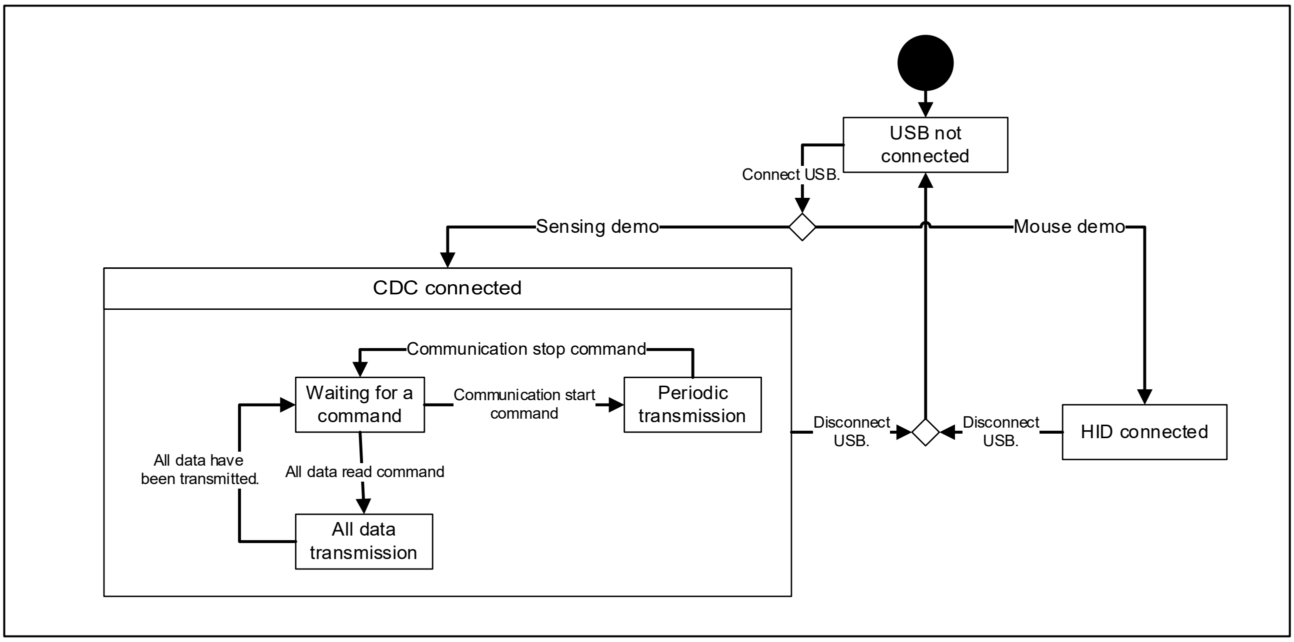


Figure 3-31 State Transitions for CDC Communications

3.14.2.1 Periodic Transmission

After reception of a communications start command, data are transmitted to the PC at 5-second intervals until reception of a communications stop command. The latest results of measurement are acquired from the temperature and humidity control process or motion sensor control process and sent to the PC. Figure 3-32 shows the state transitions in periodic transmission.

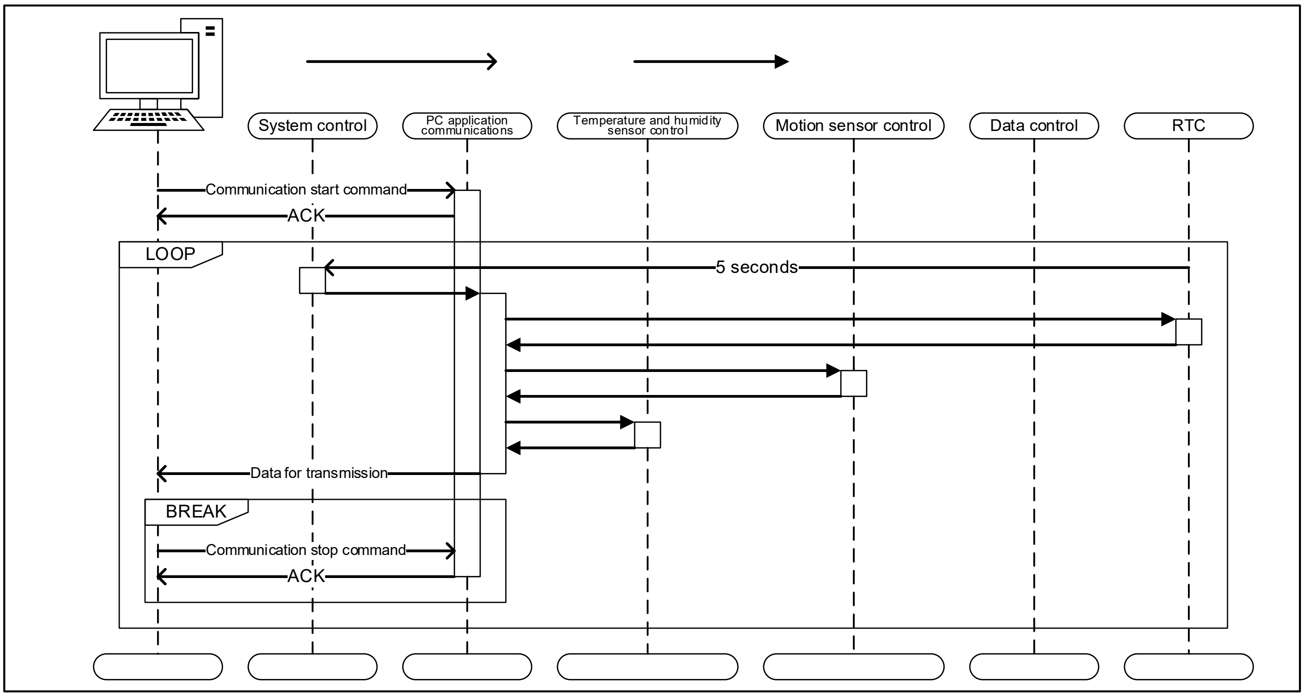


Figure 3-32 State Transitions for Periodic Transmission

3.14.2.2 All Data Transmission

After reception of an all data read command, data are sent to the PC at intervals of the main cycle. Records are read from the data control process and transmitted to the PC. Data transmission ends when there are no data to be read from the data control process. To indicate the end of data, data containing bytes filled with 0xFF except for the STX, ETX, and command bytes are sent. Figure 3-33 shows the state transitions for all data transmission.

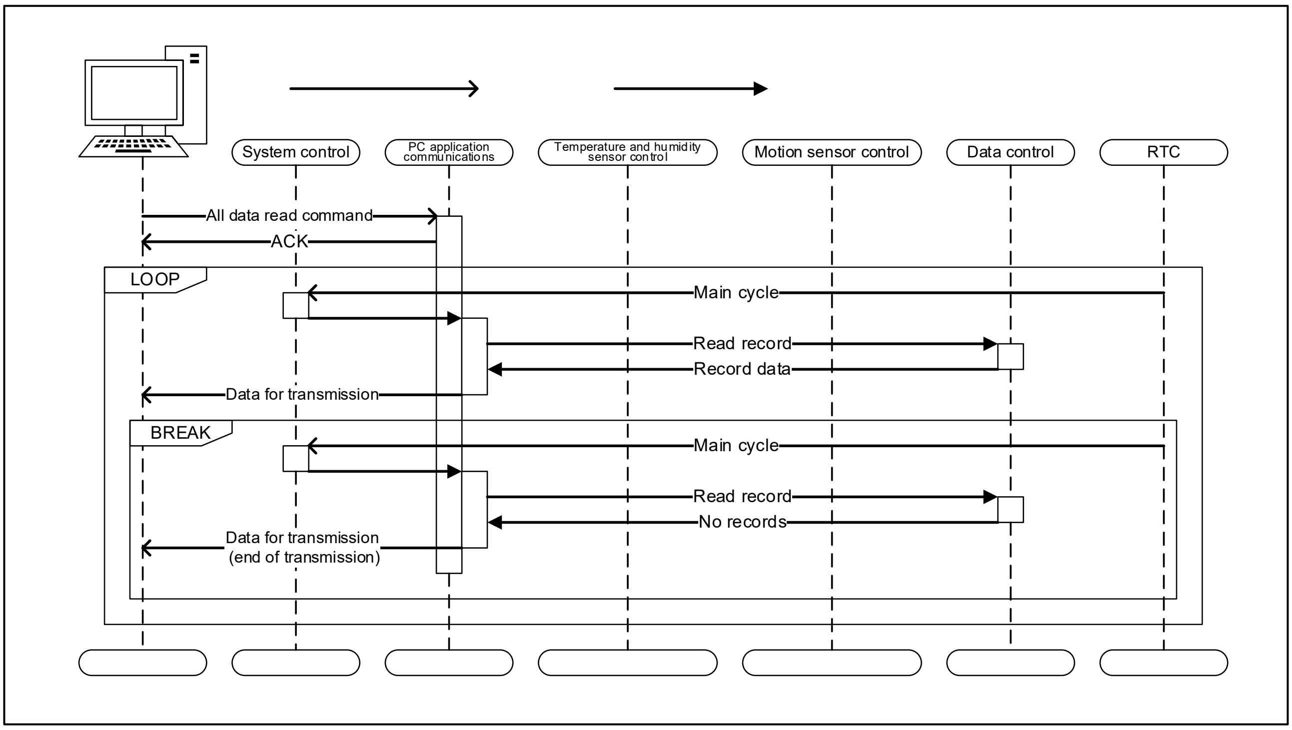


Figure 3-33 State Transitions for All Data Transmission

3.14.2.3 Format of Receive Commands

Table 3-17 shows the format of receive commands.

Table 3-17 Format of Receive Commands

Byte	Contents	Communication Start Command	Communication Stop Command	All Data Read Command
0	STX	0x02	0x02	0x02
1	Command	0x30	0x31	0x32
2	ETX	0x03	0x03	0x03

3.14.2.4 Format of Transmission Data

Table 3-18 shows the format of transmission data.

Table 3-18 Format of Transmission Data

Byte	Contents	Data for Transmission	Remark
0	STX	0x02	
1	Command	0x5A	
2	Year	0x00 to 0x99	
3	Month	0x01 to 0x12	BCD
4	Day	0x01 to 0x31	BCD
5	Hour	0x00 to 0x24	BCD
6	Minute	0x00 to 0x59	BCD
7	Second	0x00 to 0x59	BCD
8	Temperature	-40 to 125	Integer part
9		-99 to 99	Fractional part
10	Humidity	0 to 100	Integer part
11		0 to 99	Fractional part
12 to 13	Acceleration rate	-32766 to 32767	X
14 to 15		-32766 to 32767	Y
16 to 17		-32766 to 32767	Z
18 to 19	Angular velocity	-32766 to 32767	X
20 to 21		-32766 to 32767	Y
22 to 23		-32766 to 32767	X
24	ETX	0x03	

3.14.3 HID Report Transmission

Input from the joystick is checked at intervals of the main cycle. If an input is detected, an HID report is sent to the PC application. Figure 3-34 shows the state transitions for HID report transmission.

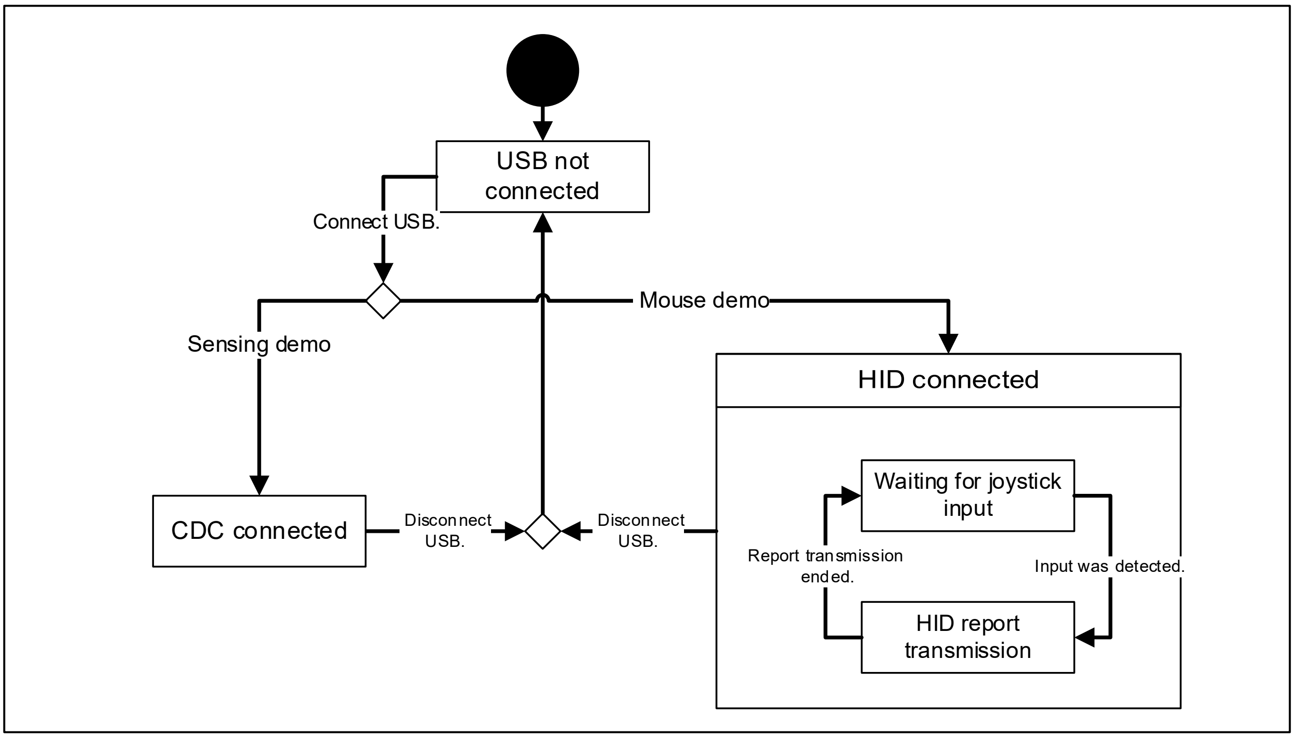


Figure 3-34 State Transitions for HID Report Transmission

3.14.3.1 Format of HID Report

Table 3-19 shows the format of the HID report.

Table 3-19 Format of the HID Report

Byte Position	Bit Position	Data Length	Item for Setting	Range of Setting	Remark
0	0	1 bit	Button	0 to 1	
0	1	7 bits	Fixed value	0	
1	0	8 bits	Amount of X-axis movement	-127 to 127	
2	0	8 bits	Amount of Y-axis movement	-127 to 127	

3.14.3.2 Settings in the HID Report

Table 3-20 shows the settings in the HID report

Table 3-20 Settings in the HID Report

<table>
<tr><th>Joystick Input</th><th>Item for Setting</th><th>Setting</th><th>Remark</th></tr>
<tr><td rowspan="3">No input</td><td>Button</td><td>0</td><td rowspan="3">Reports are not sent when nothing is input.</td></tr>
<tr><td>Amount of X-axis movement</td><td>0</td></tr>
<tr><td>Amount of Y-axis movement</td><td>0</td></tr>
<tr><td>Press in the center</td><td>Button</td><td>1</td><td></td></tr>
<tr><td>Press right</td><td rowspan="2">Amount of Y-axis movement</td><td>+5</td><td rowspan="4">The input direction is handled as being rotated 90° to the left relative to the orientation of the LCD.</td></tr>
<tr><td>Press left</td><td>-5</td></tr>
<tr><td>Press up</td><td rowspan="2">Amount of X-axis movement</td><td>+5</td></tr>
<tr><td>Press down</td><td>-5</td></tr>
</table>

3.15 VBUS Monitoring

This section describes the operation of the VBUS monitoring function. This function uses QE for USB, a USB development support tool, to enable monitoring of the VBUS voltage and current. For details on how to use the VBUS monitoring function through QE for USB, refer to Chapter 8 in QE for USB Usage Guide (R20AN0413).

3.15.1 Processing Structure

Figure 3-35 shows the structure of the VBUS monitoring processes.

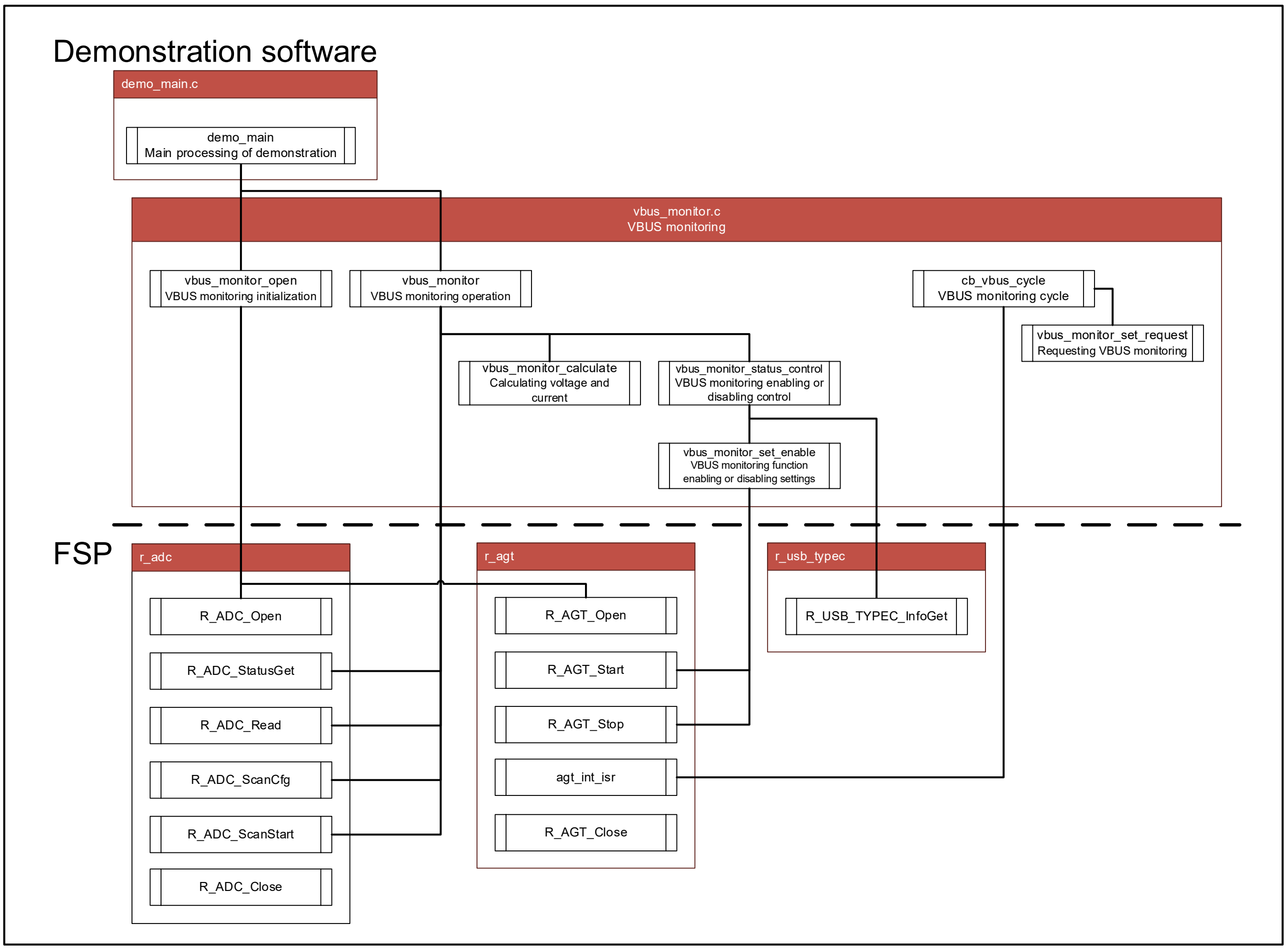


Figure 3-35 Structure of the VBUS Monitoring Processes

3.15.2 VBUS Monitoring Operation

The VBUS monitoring function operates while VBUS ON is detected by the USB control process. The results of A/D conversion of the values from the ADC pins that are assigned to VSENSE and ISENSE are measured at 40-ms intervals. The results of A/D conversion for VSENSE and ISENSE are only read once at this time and the obtained VSENSE and ISENSE values are converted to voltage and current values and stored as global data. Note that the voltage and current values are set to 0 while VBUS monitoring is stopped. Figure 3-36 shows the state transitions for VBUS monitoring.

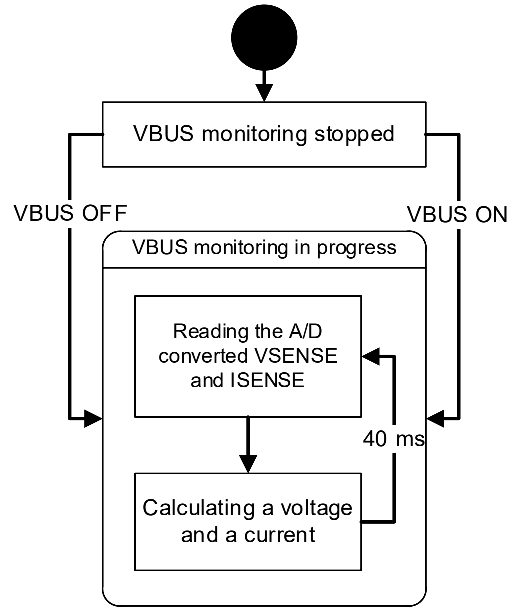


Figure 3-36 State Transitions for VBUS Monitoring

3.15.3 Calculation of VBUS Voltage and Current

The following equations are used to calculate the VBUS voltage and current values. Table 3-22 shows the specifications of the data (variables) for storing the voltage and current values.

$$\text{Voltage} = \frac{\text{reference voltage}}{2^{12} - 1} \times \text{A/D} - \text{converted VSENSE } \times 2$$

$$\text{Current} = \frac{\frac{\text{reference voltage}}{2^{12} - 1} \times \text{A/D} - \text{converted ISENSE}}{\text{resistance} \times 20}$$

Table 3-21 Values Used in the Equations

Item	Value	Remark
Reference voltage	3.3 V	AVCC0
Resistance	0.05 Ω	

Table 3-22 Specifications of Data

Data	Data Name	Data Type	Remark
Voltage	g_VbusMonitor_VsenseData	uint16_t	Unit: mV
Current	g_VbusMonitor_IsenseData	uint16_t	Unit: mA

3.16 USB Vendor ID and Product ID Setting

The sample code prepares the definitions of the USB vendor ID and product ID in r_usb_pcdc_description.c (see the orange rectangle in Figure 3-37). The values in the figure are the default settings of the vendor ID and produce ID, so change them to the IDs owned by the user when the sample code is used on the user's actual product.

```
/*******************************************************************************
 * File Name    : r_usb_pcdc_descriptor.c
 * Description  : Contains function definitions.
 *******************************************************************************/
/*******************************************************************************
* Copyright (c) 2025 Renesas Electronics Corporation and/or its affiliates
*
* SPDX-License-Identifier: BSD-3-Clause
*******************************************************************************/
/*******************************************************************************
 Includes   <System Includes> , "Project Includes"
 ******************************************************************************/
#include "r_usb_basic.h"
#include "r_usb_basic_api.h"
#include "r_usb_basic_cfg.h"

/*******************************************************************************
 Macro definitions
 ******************************************************************************/
/* bcdUSB */
#define USB_BCDNUM                        (0x0200U)
/* Release Number */
#define USB_RELEASE                       (0x0200U)
/* DCP max packet size */
#define USB_DCPMAXP                       (64U)
/* Configuration number */
#define USB_CONFIGNUM                     (1U)
/* Vendor ID */
#define USB_VENDORID                      (0x0000U)
/* Product ID */
#define USB_PRODUCTID                     (0x0002U)
```

Figure 3-37 Definitions of the USB Vendor ID and Product ID in the Sample Code

3.17 Amounts of ROM and RAM Usage

Table 3-23 shows the amounts of ROM and RAM used by the sample code.

- Compiler used: GCC Arm Embedded 13.2.1.arm-13-7
- Optimization level of the compiler: Optimize more (-O2)

Table 3-23 Amounts of Memory Usage

Function	ROM Usage (Bytes)	RAM Usage (Bytes)
Main processing, system control, key input, etc.	3684	248
USB driver (FSP)	16157	2227
USB control	1058	21
CDC communications	922	409
HID report transmission	384	157
Battery control (charging control and remaining charge calculation)	1983	31
Sensor control (temperature and humidity sensor control and motion sensor control)	3323	185
Data control (storing and reading)	756	34
LCD display	57481	10446
Total	85748	13778

4. Procedure for Starting to Use the Sample Code

This chapter describes the steps from importing the project files for the sample code to building a project in the e^2 studio. The series of steps is as follows:

(1) Download and install the latest versions of the e^2 studio and FSP

(2) Download the sample code

(3) Start the e^2 studio and import the project files for the sample code

(4) Build the project

4.1 Downloading and Installing the Latest Versions of the e^2 studio and FSP

First, download the latest version of the FSP platform installer from the Web page for the RA Flexible Software Package (FSP).

- Web page for RA Flexible Software Package (FSP)

Execute the downloaded file and proceed to install each item by following the procedure described by the installer. The installation procedure will install the latest versions of the e^2 studio and FSP. The following versions of the e^2 studio and FSP were used in developing this application note.

- e^2 studio: e^2 studio 2025-04.1
- FSP: v6.0.0

4.2 Downloading the Sample Code

Next, download the sample code from the Web page for the USB Type-C reference design for RA2L2 MCUs.

- Web page for USB Type-C Reference Design for RA2L2 MCUs

4.3 Starting the e² studio and Importing the Project Files for the Sample Code

Start the e² studio and select [Import projects] (see Figure 4-1).

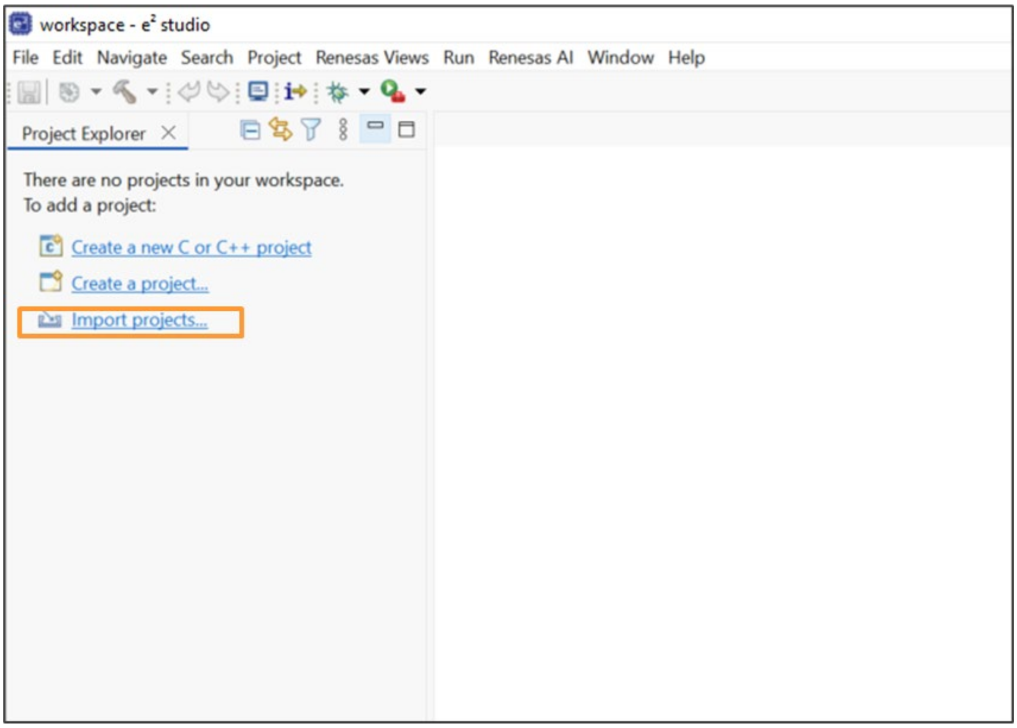


Figure 4-1 Selecting the Importing of Projects

Select the folder containing the downloaded set of sample code and import the project files for the sample code.

If the project files have been imported normally, the folders of the sample code are displayed as shown in Figure 4-2, and each of the source files can be confirmed.

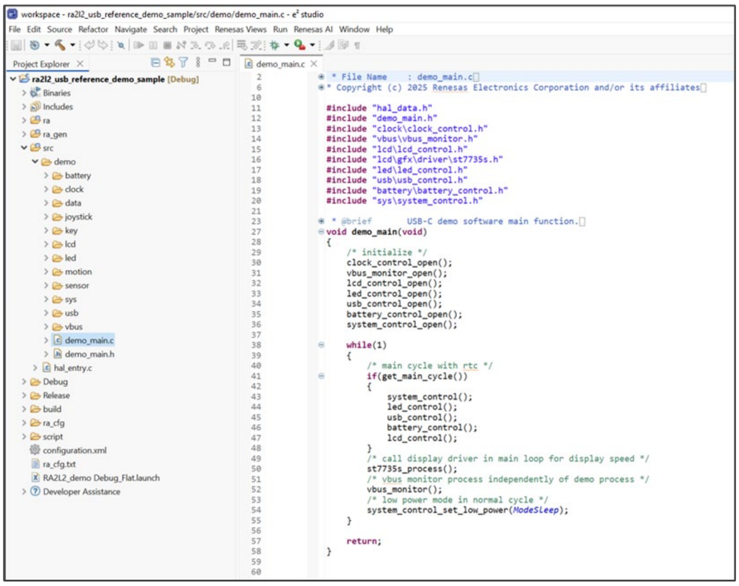


Figure 4-2 e² studio Window after the Completion of Importing

4.4 Building a Project

Right-click on the imported project file then click on [Build Project] to execute the building process, as shown in Figure 4-3.

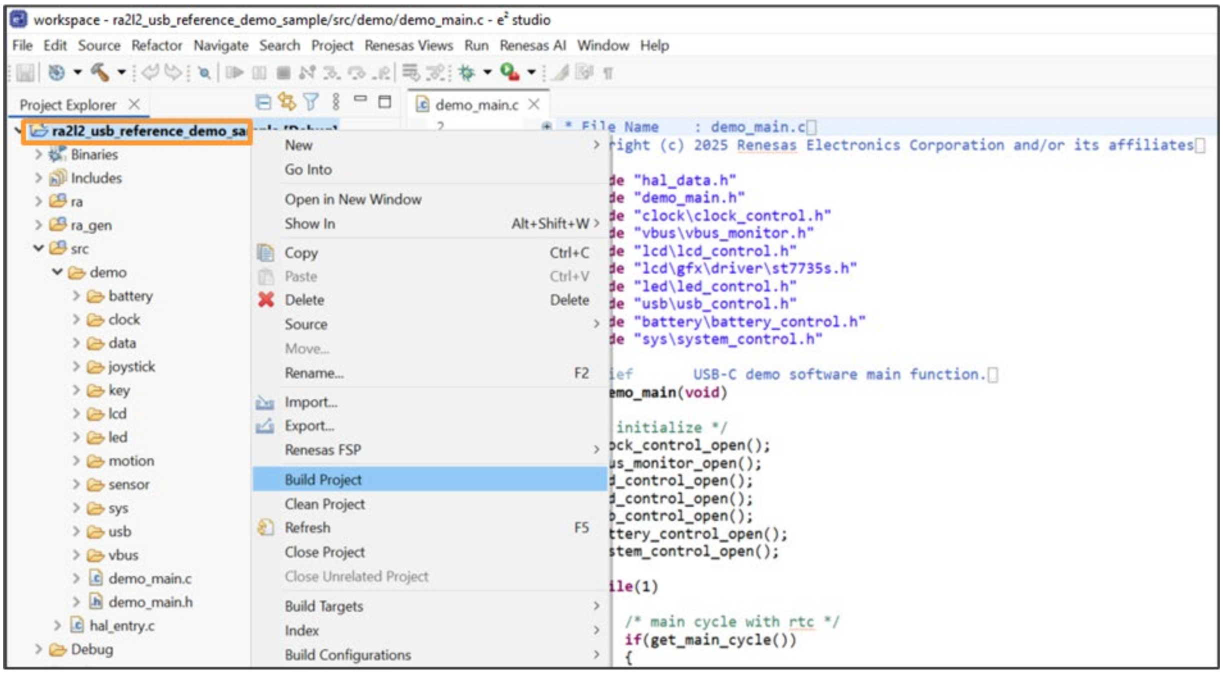


Figure 4-3 Building a Project

If the result of building is displayed as shown in Figure 4-4 (divided into two parts here because there are so many lines), building was successful. Note that Figure 4-4 below shows the displayed results of building the sample code for the first time. The building time and amount of displayed results will be reduced from the second time on.

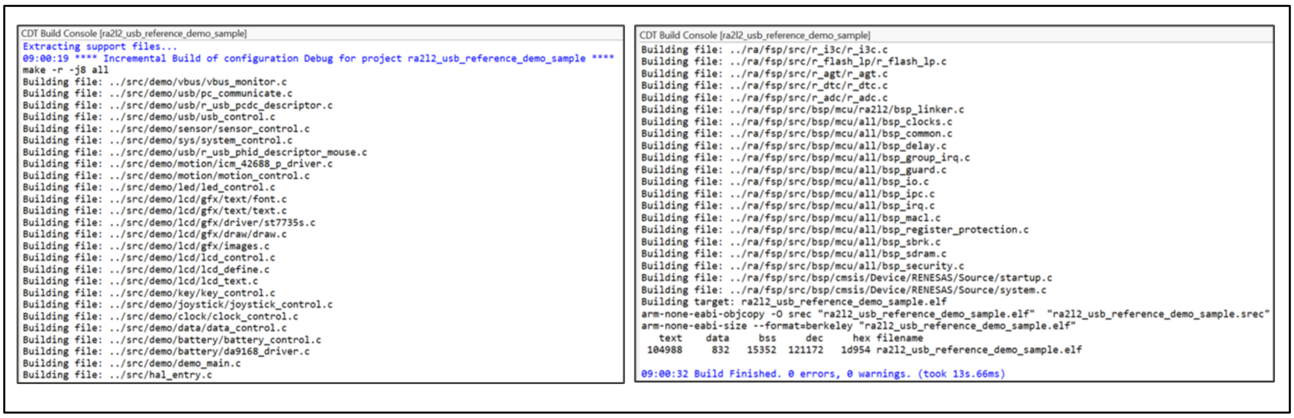


Figure 4-4 Results of Building

This ends the explanation of the procedure for starting the sample code for this reference design.

5. Hardware Specifications

This chapter explains hardware specifications by functional blocks of the demo board. Note that the descriptions of the USB and charging blocks from Chapter 2 are not repeated in this chapter.

5.1 MCU Current Consumption Measuring Circuit

A circuit for measuring current consumption by the MCU is placed between the MCU's power supply (3.3 V) and the VCC pin on this demo board. Figure 5-1 shows the placement of the MCU current consumption measuring circuit. When measuring the current consumption, remove the jumper post attached between pins 1 and 2 of the pin header (J5) and insert a multimeter for performing the measurement.

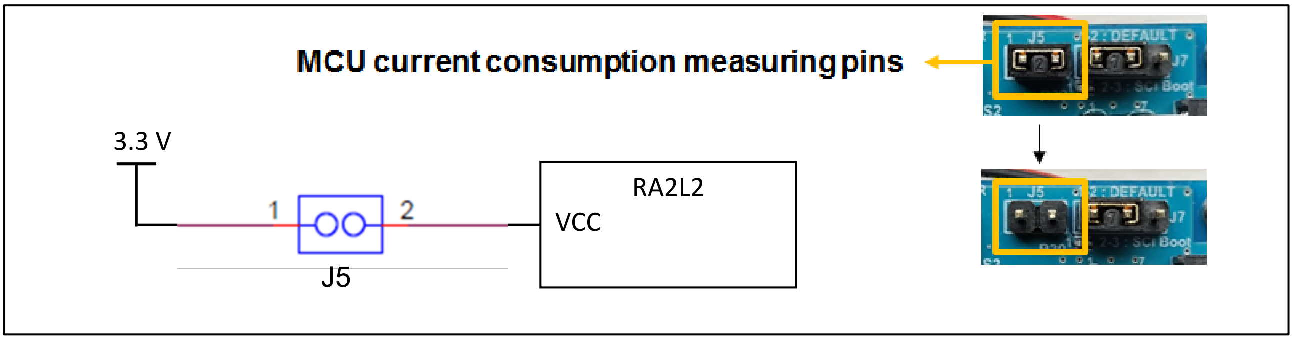


Figure 5-1 MCU Current Consumption Measuring Circuit

5.2 Sub-clock Circuit and Battery Voltage Measuring Circuit

Figure 5-2 shows the sub-clock circuit and battery voltage measuring circuit.

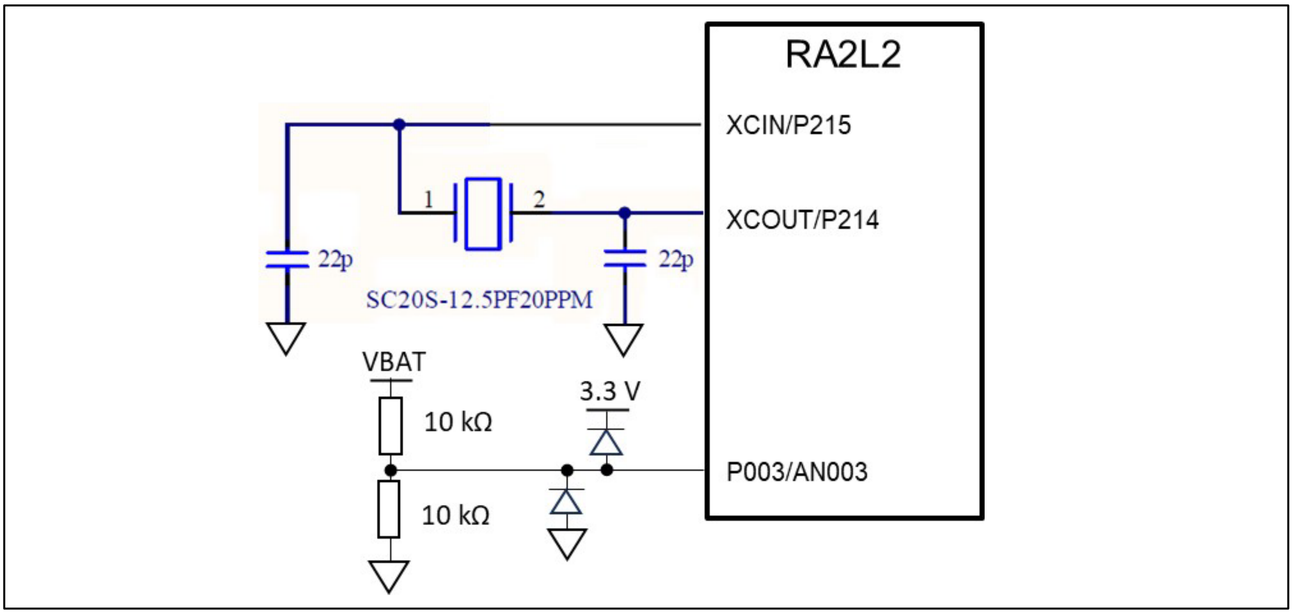


Figure 5-2 Sub-clock Circuit and Battery Voltage Measuring Circuit

5.3 Switches

Four push switches are used on this demo board. Figure 5-3 shows the peripheral circuit for the switches.

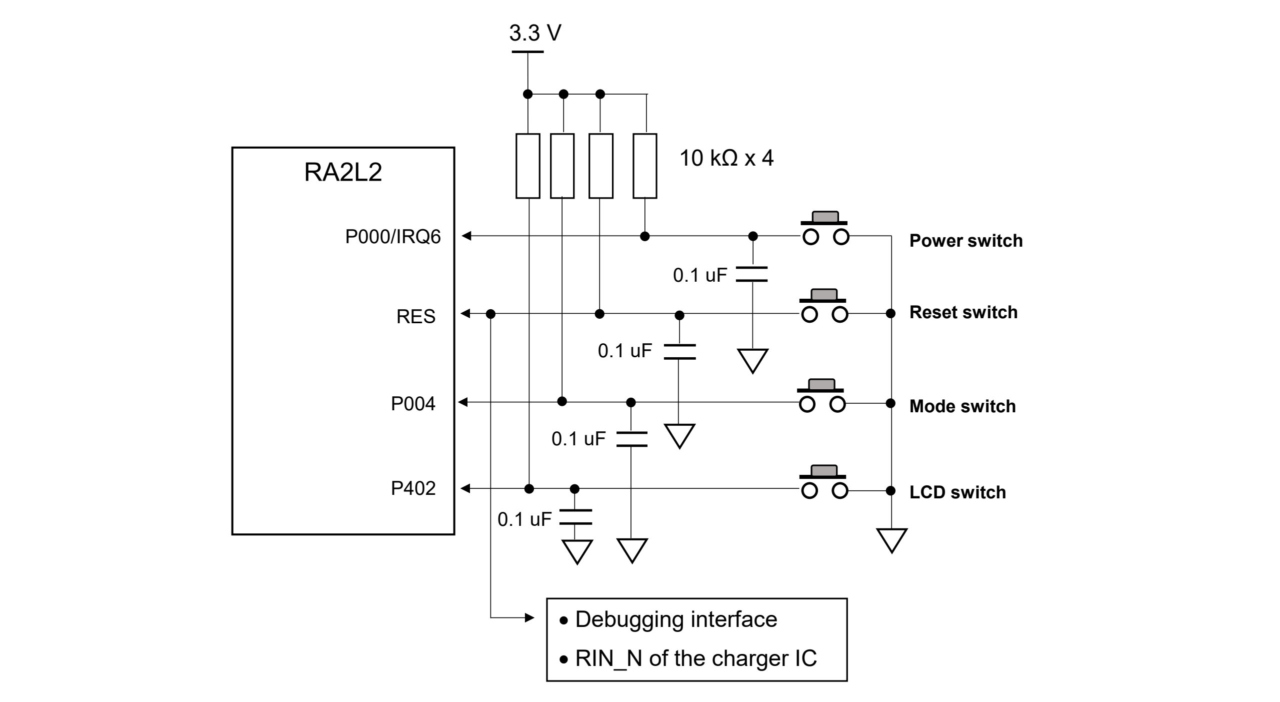


Figure 5-3 Peripheral Circuit for Switches

5.4 Joystick

A joystick is used as the means of moving the mouse cursor in mouse demo mode. Figure 5-4 shows the peripheral circuit for the joystick.

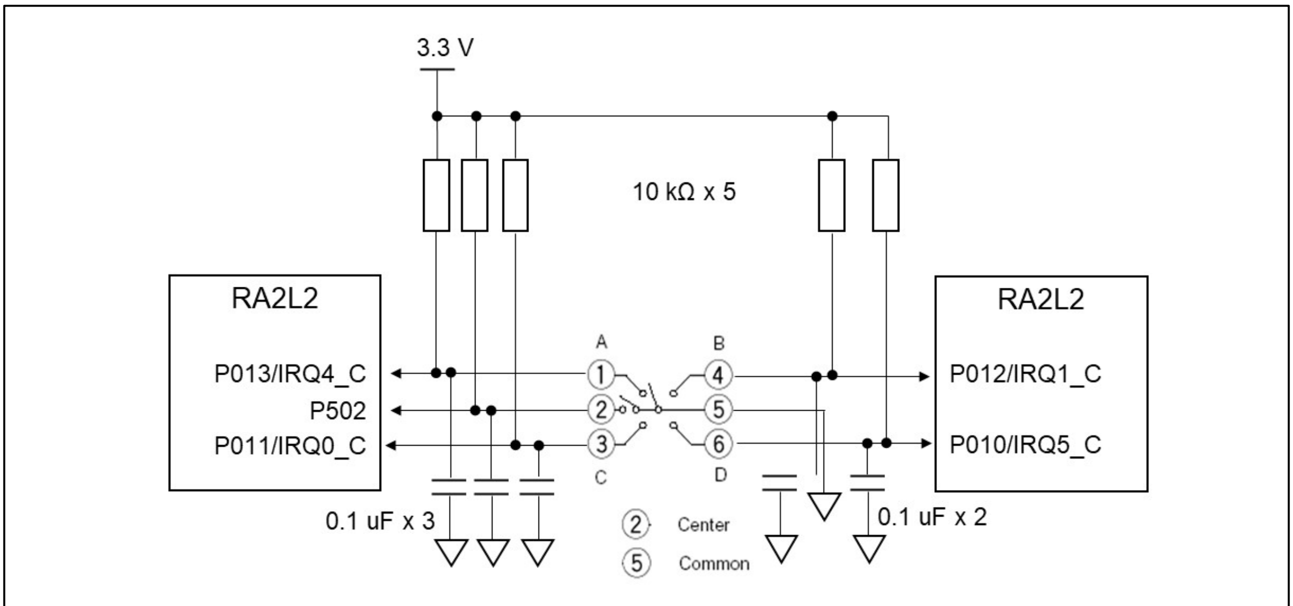


Figure 5-4 Peripheral Circuit for the Joystick

5.5 Status LEDs

The following three LEDs are used on this demo board. Figure 5-5 shows the peripheral circuit for the LEDs.

- Power LED (green): The light is on when the power of the demo board is turned on (state in which the MCU is activated) and off when the demo board is on standby.
- Charging LED (red): The light is only on during charging of the battery.
- USB LED (yellow): The light is on when the demo board is connected to the USB (VBUS power supply) and off when the demo board is not connected to the USB.

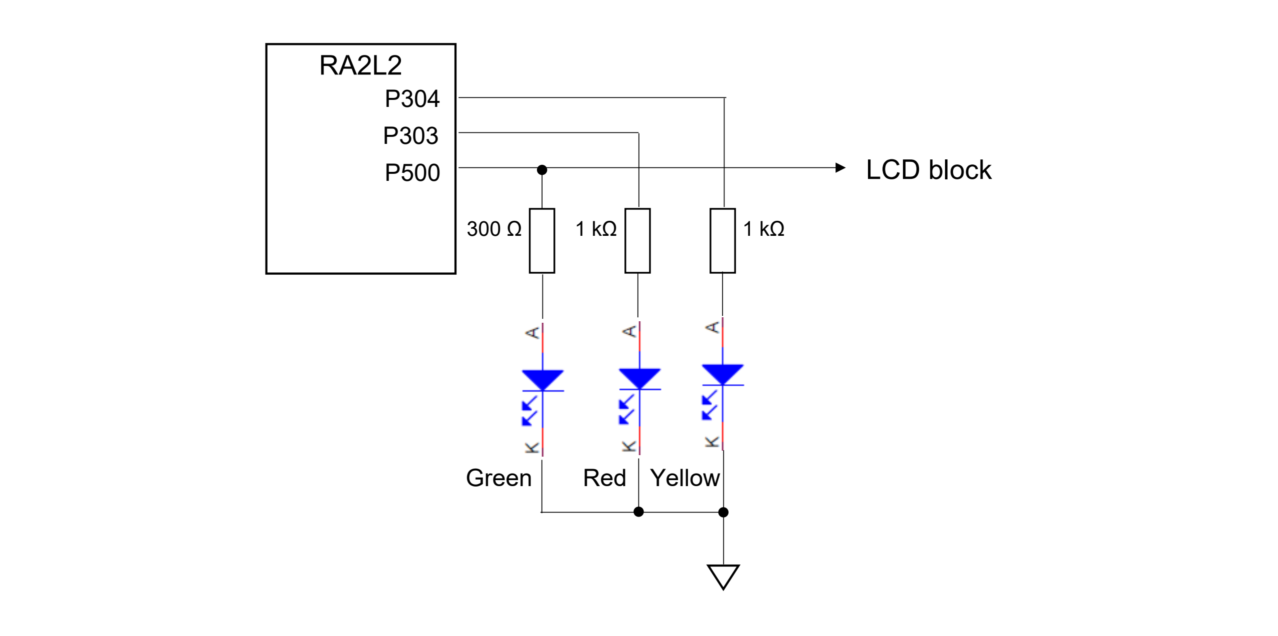


Figure 5-5 Peripheral Circuit for LEDs

5.6 Color LCD

This demo board uses the "MDT0096A2IS-SPI" color LCD from the Midas Display series by Midas Components. Figure 5-6 shows the circuit between the RA2L2 and LCD.

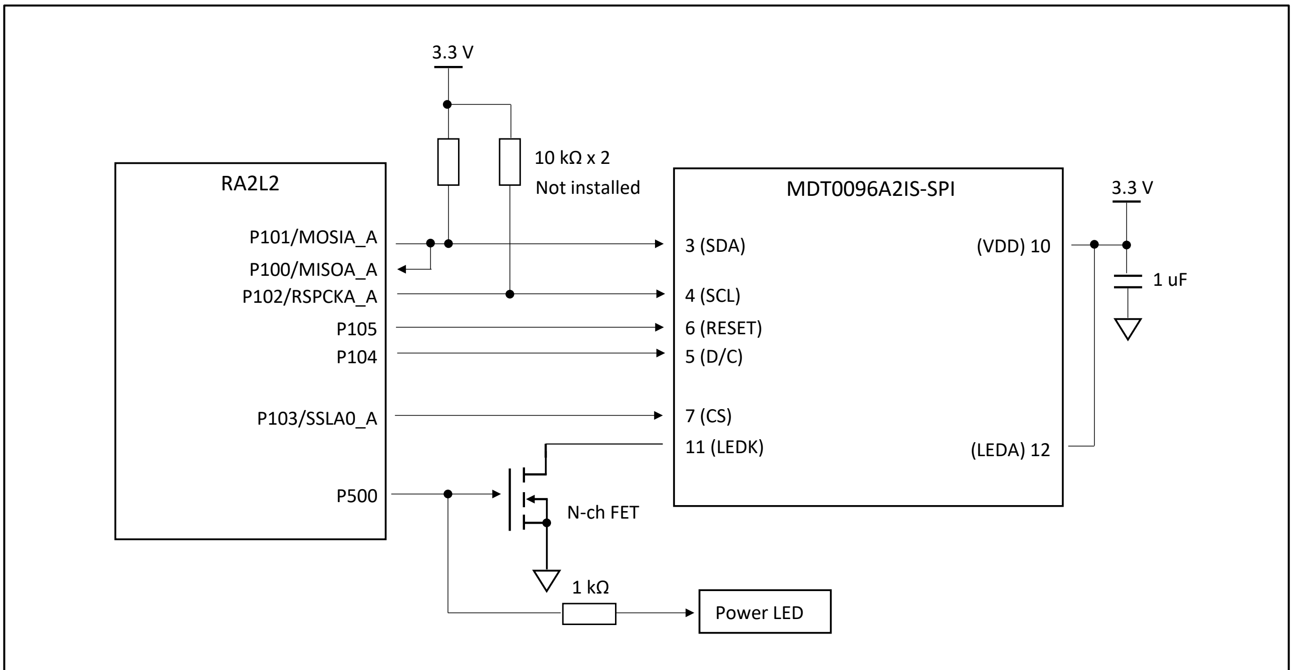


Figure 5-6 Circuit between the RA2L2 and LCD

5.7 Temperature and Humidity Sensor

This demo board uses the "HS4001" temperature and humidity sensor by Renesas. Figure 5-7 shows the circuit between the RA2L2 and temperature and humidity sensor.

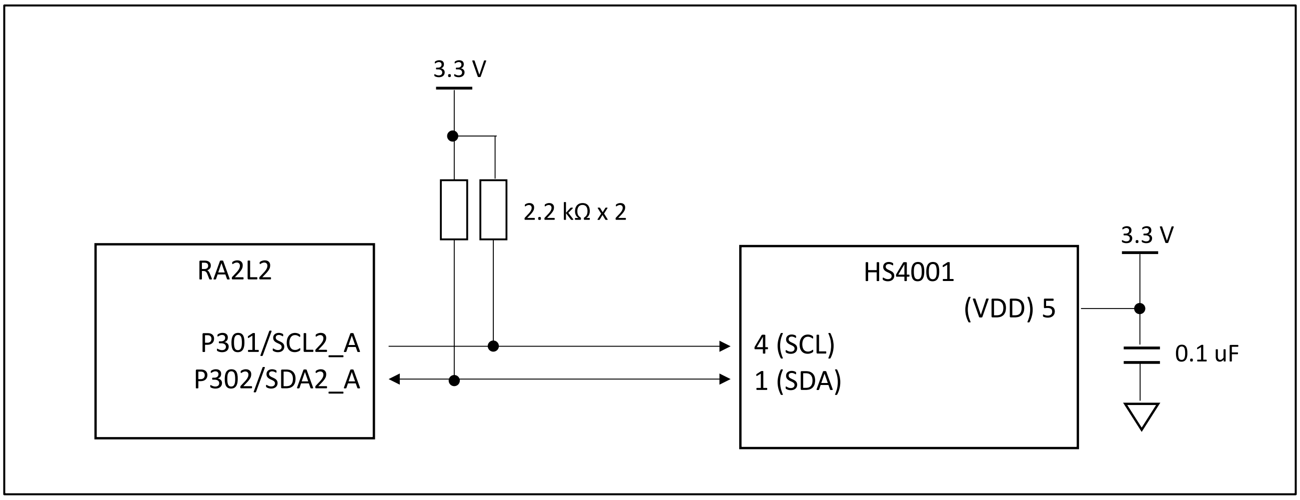


Figure 5-7 Circuit between the RA2L2 and Temperature and Humidity Sensor

5.8 Motion Sensor

This demo board uses the "ICM-42688-P" motion sensor by TDK Corporation. Figure 5-8 shows the circuit between the RA2L2 and motion sensor.

- Web page for ICM-42688-P

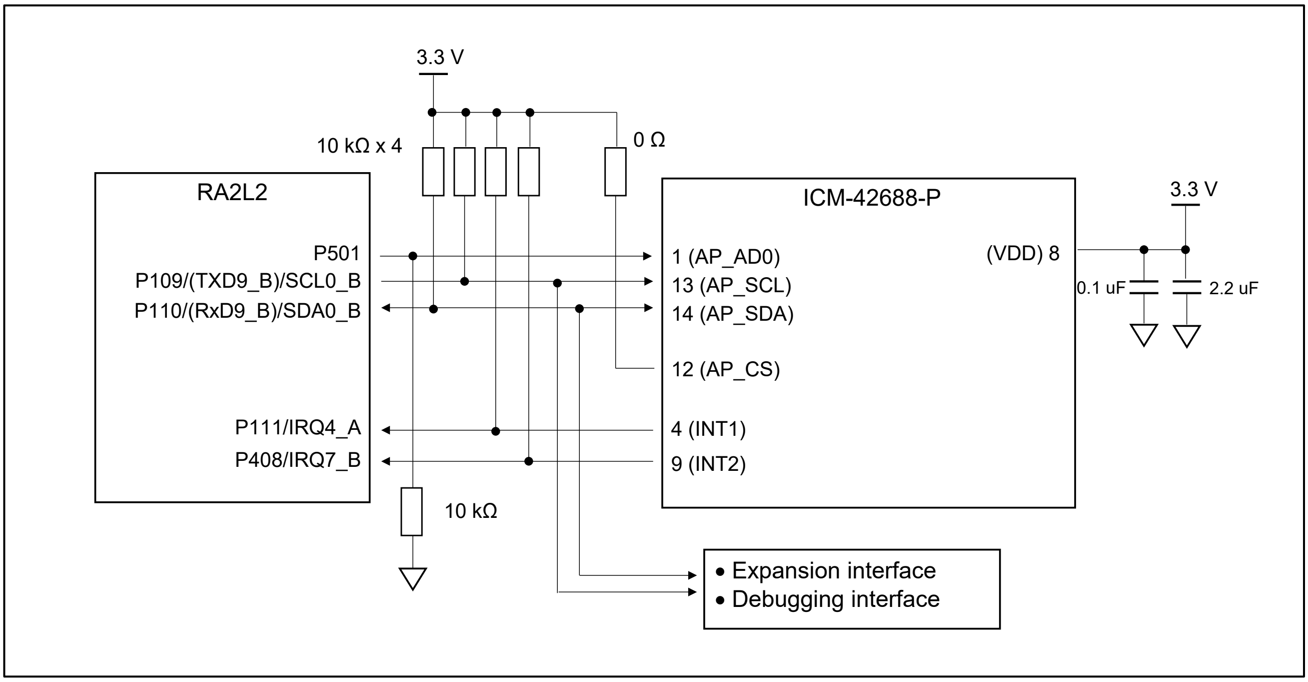


Figure 5-8 Circuit between the RA2L2 and Motion Sensor

5.9 Pmod Interface

This demo board is equipped with one Pmod interface port for expanding functionality. Type 2A and Type 3A are supported by default. The supported type can be switched to Type 6A by changing resistors to be installed from the green positions to the blue positions shown in Figure 5-9.

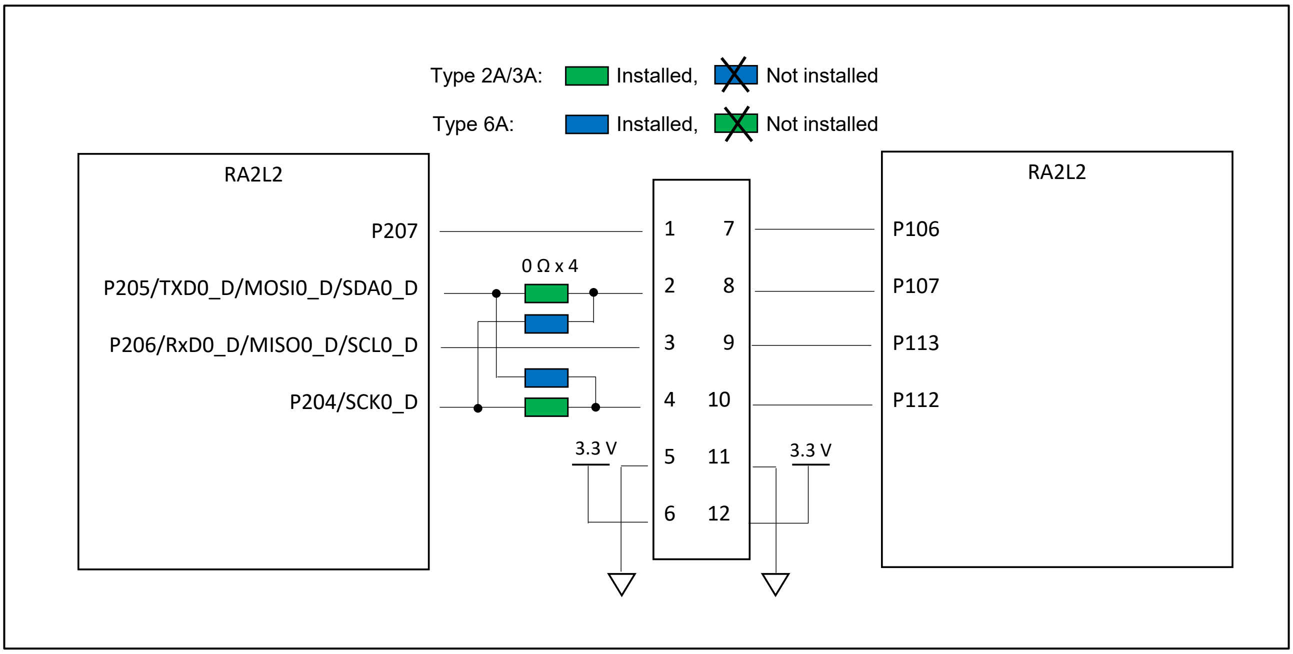


Figure 5-9 Circuit of the Pmod Interface

Table 5-1 shows the interface standards of Pmod Type 2A, 3A, and 6A.

Table 5-1 Pmod Interface Standards

Pmod Connector Pin No.	SPI (Type 2A)		UART (Type 3A)		I²C (Type 6A)	
Pin	Signal Name	Input/Output	Signal Name	Input/Output	Signal Name	Input/Output
1	CS	Output	CTS/GPIO	IO/Input	GPIO/INT	IO/Input
2	MOSI	Output	TXD	Output	GPIO/RESET	IO/Output
3	MISO	Input	RXD	Input	SCL	IO
4	SCK	Output	RTS/GPIO	Output/IO	SDA	IO
5	GND	¾	GND	¾	GND	¾
6	VCC	¾	VCC	¾	VCC	¾
7	GPIO/INT	IO/Input	GPIO	IO	GPIO	IO
8	GPIO/RESET	IO/Output	GPIO	IO	GPIO	IO
9	GPIO/CS2	IO/Output	GPIO	IO	GPIO	IO
10	GPIO/CS3	IO/Output	GPIO	IO	GPIO	IO
11	GND	¾	GND	¾	GND	¾
12	VCC	¾	VCC	¾	VCC	¾

5.10 Debugging Interface

An E2 Lite connector (dual-row, 10-pin, and 1.27-mm pitch connector) is mounted on the demo board as its debugging interface connector. Figure 5-10 shows the specifications of the debugging interface.

Figure 5-11 shows the circuit between the RA2L2 and E2 Lite connector. For the setting of the programming mode setting pins (on jumper block J7), close pins 1-2 during normal operation, debugging, and software programming by the E2 Lite, and close pins 2-3 during software programming via SCI boot. Pins 1-2 are closed in the default setting.

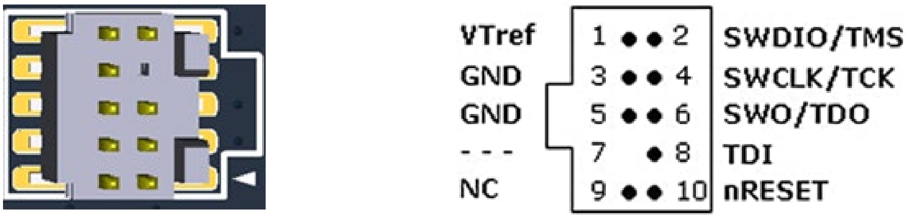
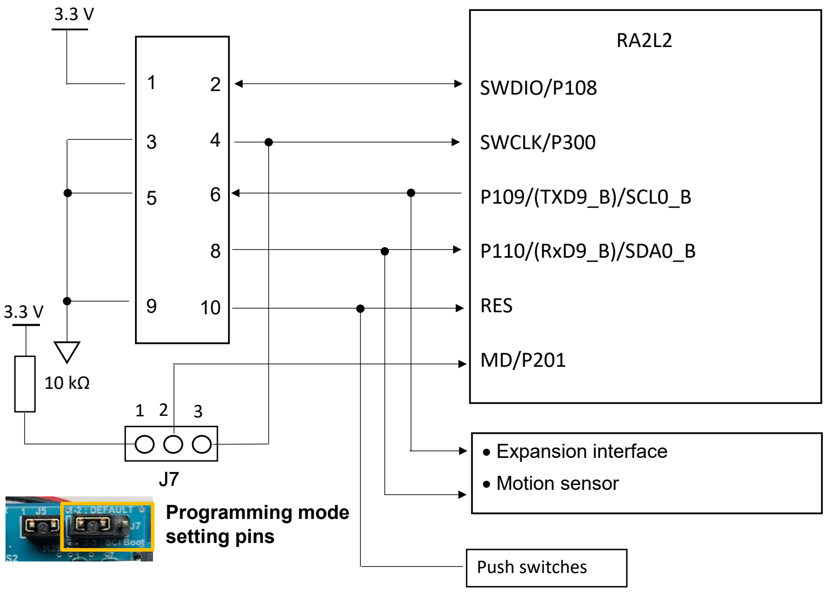


Figure 5-10 Specifications of the Debugging Interface



- **Pins 1-2 are closed: Normal operation, debugging, and programming by the E2 Lite**
- **Pins 2-3 are closed: Programming via SCI boot**

Figure 5-11 Circuit between the RA2L2 and E2 Lite Connector

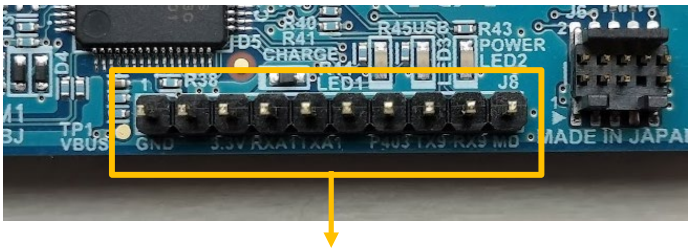
5.11 Expansion Interface

This demo board is equipped with an expansion interface connector (a single-row 10-pin 2.54 mm pin header), to which unused UARTA terminals (TXDA/RXDA) and SCI boot communication terminals (TXD9/RXD9/MD) are connected. This allows the UARTA terminals to be used for USB-UART serial conversion, and the SCI boot terminals to be used for writing software to the MCU. When using SCI boot, close pins 2-3 among the programming mode setting pins (J7).

Table 5-2 lists the specifications of the expansion interface connector, and Figure 5-12 shows the outward appearance of the expansion interface connector.

Table 5-2 Specifications of the Expansion Interface Connector

Pin	Pin Name or Signal Name	Function	Remarks
1	GND	GND	
2	—	—	Unused
3	3.3V	Power supply	Has a protective diode
4	P015/RXDA1_A	UART reception	
5	P014/TXDA1_A	UART transmission	
6	—	—	Unused
7	P403	—	
8	P109/(TXD9_B)/SCL0_B	SCI boot	
9	P110/(RxD9_B)/SDA0_B	SCI boot	
10	MD/P201	SCI boot	



Expansion interface connector

Figure 5-12 Outward Appearance of the Expansion Interface Connector

5.12 Circuit Diagram of the Board

Figure 5-13 shows a circuit diagram of the demo board. Various types of PCB design data, such as BOM and Gerber data, can be downloaded from the Web page for this reference design.

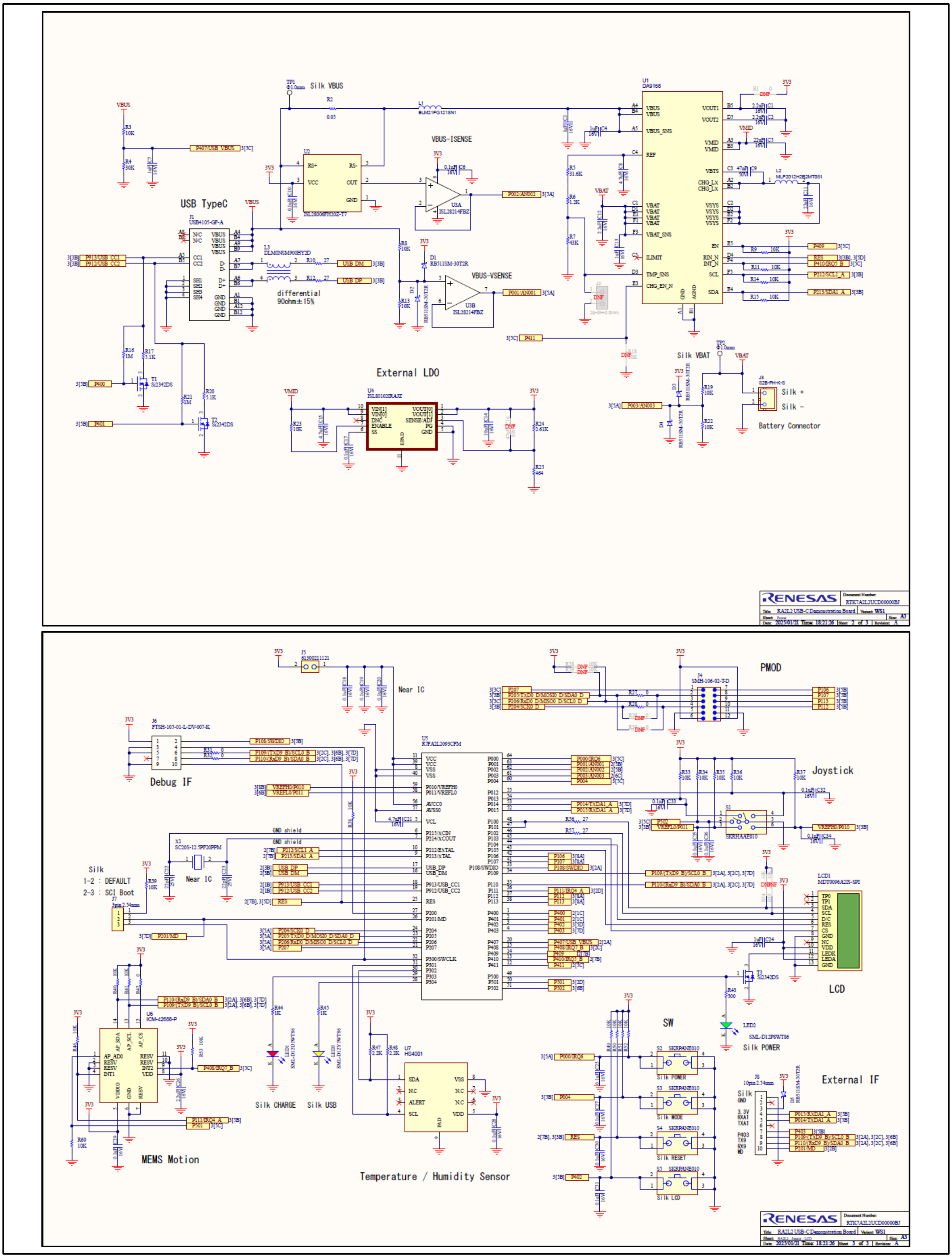


Figure 5-13 Circuit Diagram of the RA2L2 USB-C Demonstration Board

6. Reference: Information on Reducing Overall System Power Consumption

An external LDO for generating the 3.3V system power supply and operational amplifiers for measuring the VBUS current and voltage are used on the demo board. The products in Table 6-1 are used on the demo board from the viewpoint of providing commonality with components of the evaluation kit for the RA2L2 MCU group (EK-RA2L2). For users considering ways to further reduce overall system power consumption, Table 6-1 lists information for reference on substitutes with lower power consumption than that of the products actually on the board. Figure 6-1 shows the differences between the circuits when the substitute products are used.

The alternative products listed below are not part of the evaluation for this reference design. It is the user's responsibility to perform a thorough assessment before use.

Products on the demo board:

- External LDO: ISL80102IRAJZ (Web page for the product)
- Operational amplifier: ISL28214FBZ (Web page for the product)

Substitutes for reference:

- External LDO: RAA2142204GP3 (Web page for the product)
- Operational amplifier: ISL28230CBZ (Web page for the product)

Table 6-1 Product Information of the External LDO and Operational Amplifier

Product	Products on the Demo Board		Substitutes for Reference	
	Part Number	Quiescent Current (IQ)	Part Number	Quiescent Current (IQ)
External LDO (U4)	ISL80102IRAJZ	9 mA	RAA2142204GP3	0.08 mA
Operational amplifier (U3)	ISL28214FBZ	0.39 mA	ISL28230CBZ	0.025 mA

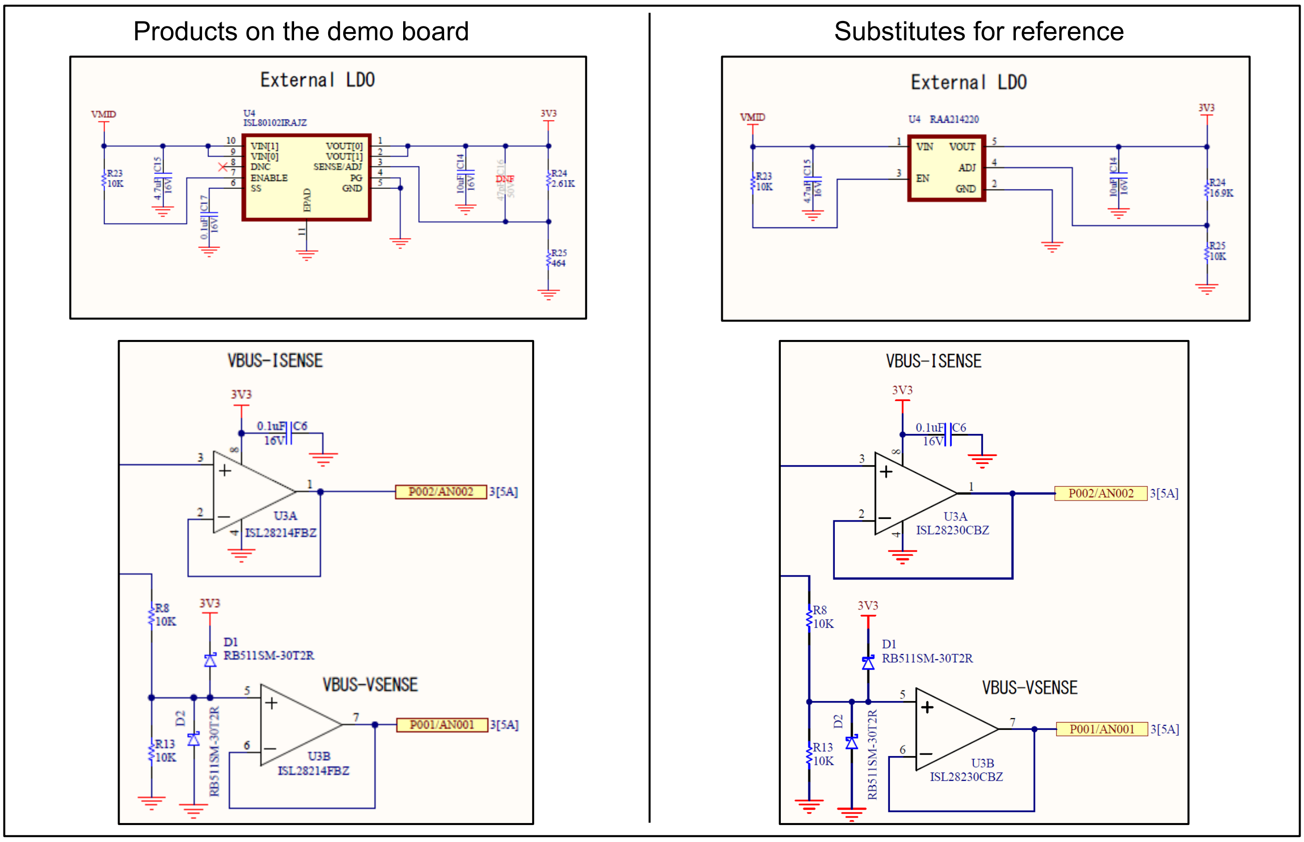


Figure 6-1 Differences between the Circuits of the External LDO and Operational Amplifiers

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug. 28, 2025	—	First edition issued.

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.

Corporate Headquarters

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

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/.