Renesas RA Family

Segment LCD application with RA4L1

Introduction

This application note provides an overview of the segment LCD controller on the RA4L1 MCU and detailed guidance on using the segment LCD to interface to the RA4L1 MCU. It outlines the steps required to configure and control a segment LCD display, enabling users to present essential information effectively. Additionally, it demonstrates how to interface with the segment LCD using the Renesas Flexible Software Package (FSP) and the e^2 studio development environment.

Users can gain a comprehensive understanding of configuring and developing segment LCD applications using the EK-RA4L1 microcontroller. This includes learning how to properly set up the segment LCD driver, configure display parameters, write application code, manage data updates efficiently, and implement debugging techniques to ensure optimal performance.

By following the provided instructions, users can develop robust applications that fully utilize the EK-RA4L1's Segment LCD controller. The guide offers a step-by-step approach, making it accessible to both beginners and experienced developers.

Target Device

RA4L1

Required Resources

To build and run the application project accompanying this application note, you will need the following:

Development Tools and Software:

- e^2 studio IDE, version 2025-01 (25.01.0) or later.
- RA Family Flexible Software package (FSP) v5.8.0 or later
- GCC ARM Embedded Toolchain: Version 13.2.1.arm-13-7
- SEGGER J-Link® RTT viewer version: 8.12f or later

The FSP and e^2 studio are bundled in a downloadable platform installer available on Renesas' website at: renesas.com/ra/fsp

Hardware:

- EK-RA4L1 (https://www.renesas.com/ek-ra4l1) with the supplied segment LCD module(RTKLCDSEG1S00001BE).
- PC running Windows® 10, Tera Term console or similar application, and an installed web browser (Google Chrome, Internet Explorer, Microsoft Edge, Mozilla Firefox, or Safari).
- USB-C cable

Prerequisites and Intended Audience

This application note assumes you have some experience with the Renesas e^2 studio ISDE and RA Family Flexible Software Package (FSP). Before you perform the procedures in this application note, follow the procedure in the *FSP User Manual* to build and run the Blinky project. Doing so enables you to become familiar with e^2 studio and the FSP and validates that the debug connection to your board functions properly. Additionally, this application note assumes that you have some theoretical background on segment LCD applications

.**Contents**

1. Overview of Segment LCD

Segment LCDs (Liquid Crystal Displays) are widely used in applications where simple, low-power, and cost-effective displays are required. They use liquid crystals that modulate light to display information in segmented forms such as numbers, letters, and symbols. These displays are often found in appliances, industrial equipment, and handheld devices.

A segment LCD consists of predefined segments that can be individually controlled to display characters or symbols. Segments are typically arranged in patterns, such as seven-segment configurations for digits or custom layouts for specific applications.

Each segment or group of segments is connected to a segment control pin, which applies alternating voltage to turn segments ON or OFF. Segment LCD controllers and drivers are responsible for multiplexing in complex displays.

Segment LCDs are distinct from graphic LCDs as they are designed for predefined segment arrangements rather than pixel-based rendering.

Each segment is labeled (For example, A, B, C, D, E, F, and G for seven-segment displays), and the pinout corresponds to these labels. Proper documentation is critical for the correct wiring of the segment LCD pins to the MCU.

2. RA MCUs With Segment LCD Controllers

RA MCUs equipped with segment LCD controllers are specifically designed to handle the control and management of segmented displays. These displays present numeric, alphanumeric, and symbolic information in a power-efficient manner.

The segment LCD controller in RA MCUs allows direct drive and control of liquid crystal displays with segmented elements without requiring complex external components.

2.1 RA4L1 MCUs Segment LCD Controller Features

Here are the key features and benefits of RA4L1 MCUs with a segment LCD Controller.

- Voltage Driving Options: Driver voltage generator can switch between internal voltage boosting method, capacitor split method, and external resistance division method.
- Contrast level control: Voltage boost circuit reference voltage selectable from 23 steps (contrast adjustment).
- Different Display Modes: Liquid crystal waveform (waveform A or B) is selectable.
- Time Slice Display: The number of time slices (up to 8) used for multiplexing the segments.
- Bias Modes: 1/2, 1/3, or 1/4 can be selected for the internal voltage boosting method, as well as external resistance division and capacitor splitting.

2.2 SLDC Overview

The segment LCD controller (SLCDC) utilizes two to four reference voltages to provide AC signals for driving traditional segment LCD panels. Depending on the LCD and MCU package (such as 100, 72, or 64-pin MCUs), up to 384 individual segments can be driven. A built-in link to the RTC allows for up to 152 segments to switch between two patterns at regular intervals. An on-chip boost driver can be used to provide configurable reference voltages up to 5.25V, allowing for simple contrast adjustment.

The SLCDC module can perform the following functions:

- Initialize, start, and stop the SLCDC.
- Set and modify the output pattern.
- Blink between two patterns based on a periodic RTC interrupt signal.
- Adjust display contrast (only when using internal voltage boosting).

- Select reference voltage mode: VL1 reference mode (1/3 or 1/4 Bias) and VL2 reference mode (1/3 Bias) can be selected at internal voltage boosting method; conventional VCC reference mode (1/3 Bias) and VL4 reference mode (1/3 Bias) can be selected at capacitor split method.

2.3 Interfacing Segment LCD Controller to Segment LCD Modules

The RA4L1 MCU, featuring a built-in segment LCD controller, connects to the LCD module via dedicated pins configured as segment pins, common pins, and driving voltage pins for control and data operations. For simple, low-segment-count displays, the controller drives each segment directly. In contrast, for higher-segment-count displays, a multiplexed driving method is used, where the controller multiplexes the segments. The controller provides segment pins (SEGXX), common pins (COMXX), and power pins (VLX) for interfacing with the LCD module.

The RA4L1 100-pin device supports 48-52 segment pins, 1-8 common signals, and four reference voltage mode pins. Depending on the multiplexing scheme and bias mode (1/2, 1/3, 1/4), along with the number of time slices (Static, 2, 3, 4, 6, 8), it can drive up to 384 pixels (individual LCD segments). For detailed information on different multiplexing schemes for different pin-packaged MCUs, refer to the RA4L1 User’s Manual.

2.4 The Process of Interfacing and Driving Segment LCDs Using the RA4L1

To successfully interface and control a segment LCD module using the RA MCU, a comprehensive understanding of multiple components is essential. By understanding the board hardware, LCD module specifications, FSP configuration, and software drivers, developers can effectively interface and control a segment LCD module, ensuring optimal performance and reliable display output.

2.4.1 Exploring Hardware Resources

It is important to review the RA4L1 MCU User's Manual, the board schematics (EK-RA4L1 schematics for this application), the LCD module's connection details/schematics, and the Hardware Board User's Manual (EK-RA4L1).

The MCU User's Manual contains essential information about the device package options (such as 100, 72, or 64-pin MCUs) used on the hardware board, as well as the pin function layout for the segment LCD interface. It also provides details on configuring the LCD driver voltage generator, available waveform options, bias and duty cycle modes, and techniques for managing multiple segments, including static segment control and multiplexing. Furthermore, it outlines the assignment of segment and common signal pins.

Reviewing the board schematics is also essential for identifying segment and common pin connections used in the design. These schematics provide details about the MCU pins interfacing with the LCD module, connector specifications for LCD mounting, jumper settings, and other relevant configurations. The EK-RA4L1 Board User’s Manual further outlines port and pin assignments used in the hardware design.

For application software design, it is necessary to review the SLCD layout diagram and its segment mapping from the LCD module. This information is critical for creating a mapping table for user applications, ensuring proper segment display control.

Note: The reference documentation used for creating this application note is in the reference section of the document.

2.4.2 Creating the LCD Segment-to-MCU Pin Mapping Table

The segment driving pins are distributed across multiple I/O ports of the RA4L1 MCU. Figure 1. SLCD display function pins for RA4L1 100-pin part MCU illustrates the segment LCD display pins, including Segment Pins (SEGXX), Common Pins (COMXX), and Power Pins (VLX), specifically for the 100-pin packaged RA4L1 device.

These pins serve as the interface between the MCU and the LCD module and are routed through a pin header on the board’s LCD connector. The proper mapping of these connections is essential for ensuring accurate segment control and stable LCD operation. Understanding the pin distribution across the MCU ports is crucial for configuring the software and hardware correctly, especially when handling multiplexed segment displays.

LCD controller/ driver	**Number of segment pins (SEG): 52 (48)[*1] Number of common pins (COM): 8**															
Multiplexed I/O port	**b15**	**b14**	**b13**	**b12**	**b11**	**b10**	**b9**	**b8**	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT 0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PORT 1	SEG 31	SEG 30	SEG 29	SEG 3/COM 7	SEG 2/COM 6	SEG 1/COM 5	SEG 0/COM 4	SEG 28	SEG 38	SEG 39	SEG 40	SEG 41	SEG 42	SEG 43	SEG 44	SEG 45
PORT 2	—	—	—	—	SEG 19	SEG 20	SEG 21	COM 1	CAPH[*2]	CAPL[*2]	COM 0	SEG 18	—	—	—	—
PORT 3	—	—	—	—	—	—	—	—	SEG 22	SEG 23	SEG 24	COM 2	COM 3	SEG 25	SEG 26	SEG 27
PORT 4	SEG 13	SEG 14	SEG 15	SEG 16	VL1[*3]	VL2[*3]	VL4[*3]	VL3[*3]	SEG 17	SEG 10	SEG 9	SEG 8	SEG 7	SEG 6	SEG 5	SEG 4
PORT 5	—	—	—	—	—	—	—	—	—	—	SEG 51	SEG 50	SEG 49	SEG 48	SEG 47	SEG 46
PORT 6	—	—	—	—	—	SEG 34	SEG 33	SEG 32	—	—	—	—	—	SEG 35	SEG 36	SEG 37
PORT 7	—	—	—	—	—	—	—	SEG 12	—	—	—	—	—	—	—	SEG 11
PORT 8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Note 1. ( ) indicates the number of signal output pins when 8-time slice is selected.
Note 2. CAPH and CAPL are capacitor connection pins for the LCD controller/driver.
Note 3. VL1, VL2, VL3, and VL4 are the power supply pins for driving the LCD.

Figure 2. SLCD display function pins for RA4L1 100-pin part MCU

Figure 2. LCD module segment layout diagram shows the LCD display's layout diagram. The LCD module pins 1 through 48, which can be directly connected to the MCU for segment control, are shown. These pins serve as the interface through which the MCU drives the LCD segments by applying appropriate signals. The LCD module's pins are connected to the MCU board via a dedicated module connector, facilitating seamless electrical connectivity between the MCU and the display. The proper configuration of these connections is essential for accurate segment control, ensuring clear and stable display performance.

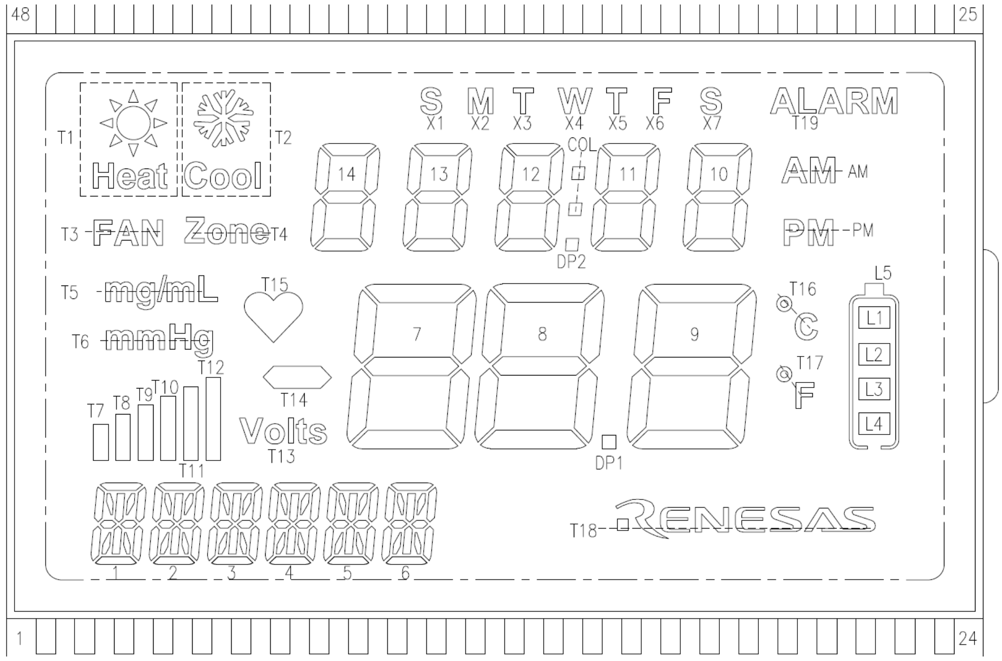


Figure 3. LCD module segment layout diagram

While creating the user application, the user needs to know which segment pin from the MCU drives the LCD segments. Also, since the LCD segments are multiplexed, the user needs to create an appropriate segment driving code value to turn on the Digit/Icon, as shown in Figure 3. LCD module pin to MCU port and pin mapping for EK-RA4L1 Board. The table helps the user to create a binary encoded value for each segment and to create a pattern for the application code.

Users are required to verify that these pins are laid out in the same order given in the schematics to start with. This can also be verified using the pins tab of the configurator to individually verify each pin configured and used as a segment pin connecting the LCD module.

LCD Module Pin	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
MCU Port Pin (RA4L1)	P205	P208	P304	P303	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC
LCD SEGMENT Registers(Hex)																
LCD SEGMENT																
COM1	COM1	_	_	_	1H	1A	2H	2A	3H	3A	4H	4A	5H	5A	6H	6A
COM2	_	COM2	_	_	1I	1J	2I	2J	3I	3J	4I	4J	5I	5J	6I	6J
COM3	_	_	COM3	_	1N	1M	2N	2M	3N	3M	4N	4M	5N	5M	6N	6M
COM4	_	_	_	COM4	1D	1L	2D	2L	3D	3L	4D	4L	5D	5L	6D	6L
LCD Module Pin	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
MCU Port Pin(RA4L1)	P115	P608	P209	P113	P211	P210	P600	P107	P602	P402	P403	P404	P405	P414	P406	P700
LCD SEGMENT Registers(Hex)	0x1F	0x20	0x15	0x1D	0x13	0x14	0x25	0x26	0x23	0X06	0X07	0X08	0X09	0X0E	0X0A	0X0B
LCD SEGMENT	31	32	21	29	19	20	37	38	35	6	7	8	9	14	10	11
COM1	7F	7A	8F	8A	9F	9A	L5	L1	AM	10D	X5	11D	X4	DP2	12D	X1
COM2	7G	7B	8G	8B	9G	9B	PM	L2	T19	10C	10E	11C	11E	COL	12C	12E
COM3	7E	7C	8E	8C	9E	9C	T16	L3	X7	10B	10G	11B	11G	X3	12B	12G
COM4	T14	7D	T18	8D	DP1	9D	T17	L4	X6	10A	10F	11A	11F	X2	12A	12F
LCD Module Pin	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
MCU Port Pin(RA4L1)	P708	P415	P407	P204	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC
LCD SEGMENT Registers(Hex)	0X0C	0X0D	0x11	0x12												
LCD SEGMENT	12	13	17	18												
COM1	13D	T1	14D	T2	T15	T13	T4	T3	T5	T6	T12	T11	T10	T9	T8	T7
COM2	13C	13E	14C	14E	6B	6F	5B	5F	4B	4F	3B	3F	2B	2F	1B	1F
COM3	13B	13G	14B	14G	6K	6G	5K	5G	4K	4G	3K	3G	2K	2G	1K	1G
COM4	13A	13F	14A	14F	6C	6E	5C	5E	4C	4E	3C	3E	2C	2E	1C	1E

Figure 4. LCD module pin to MCU port and pin mapping for EK-RA4L1 Board

2.4.3 Generating Segment Values for a Particular Digit

Figure 3. LCD module pin to MCU port and pin mapping for EK-RA4L1 Board illustrates the connections between the LCD module pins and the corresponding PORT pins, along with their associated segment numbers and registers. For example, according to the schematic design, LCD module pin 17 is connected to Port 1, Pin 15. The register responsible for driving this pin and writing the segment value is 0x400D001F. In this design, four segments are multiplexed to a single pin, specifically for digit 7 (segments E, F, G, and the Minus Icon (T14)).

For example, to turn on the digit 7 (LCD segment 31 and 32) with a value of “-3”, segments E and F need to be turned OFF, and segment G needs to be turned along with segment T14. Additionally, segments (A, B, C, D) of digit 7 need to be turned ON as well by using the segment register 0x400D0020. The resulting value is written “0x0F” to segment register 0x400D0020 and “0xC0” to register 0x400D001F.

More details on the specific registers and segment registers and the corresponding display area for different pin count packages can be found in the MCU User’s Manual. A sample snapshot of the table is given in Figure 4. Relationship between LCD Display registers and segment/common outputs.

Register name	Address	b7	b6	b5	b4	b3	b2	b1	b0	100-pin	72-pin	64-pin LQFP	64-pin BGA
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0				
SEG0	0x400D_4100	SEG0 (B-pattern area)				SEG0 (A-pattern area)				A	A	A	A
SEG1	0x400D_4101	SEG1 (B-pattern area)				SEG1 (A-pattern area)				A	A	A	A
SEG2	0x400D_4102	SEG2 (B-pattern area)				SEG2 (A-pattern area)				A	A	A	A
SEG3	0x400D_4103	SEG3 (B-pattern area)				SEG3 (A-pattern area)				A	A	A	A
SEG4	0x400D_4104	SEG4 (B-pattern area)				SEG4 (A-pattern area)				A	A	A	A
SEG5	0x400D_4105	SEG5 (B-pattern area)				SEG5 (A-pattern area)				A	A	A	A

Figure 5. Relationship between LCD display registers and segment/common outputs

Below, Figure 5. Sample Segment Encoding with 7-Segment Display shows the binary representation of the segments to write the digits with 0-9 along with the DP (Decimal point). A similar table is useful for the application developers as part of the code development. The supporting code as part of this app note has the required data (in the form of #define and Enums) in the slcd.c/h file to turn ON/OFF segments and icons

Note: Care must be taken while identifying the binary value for the segments based on how it is mapped. It is confusing if proper care is not taken, for example, in Figure 3. LCD module pin to MCU port and pin mapping for EK-RA4L1 Board, Segments 17 and 18 are mapped differently compared to 31 and 32. COM1 controls segment A in Segment 32, while COM4 controls segment A in Segment 17.

Digits/segments	dp	g	f	e	d	c	b	a
0	0	0	1	1	1	1	1	1
1	0	0	0	0	0	1	1	0
2	0	1	0	1	1	0	1	1
3	0	1	0	0	1	1	1	1
4	0	1	1	0	0	1	1	0
5	0	1	1	0	1	1	0	1
6	0	1	1	1	1	1	0	1
7	0	0	0	0	0	1	1	1
8	0	1	1	1	1	1	1	1
9	0	1	1	0	1	1	1	1
8.	1	1	0	1	1	0	1	0

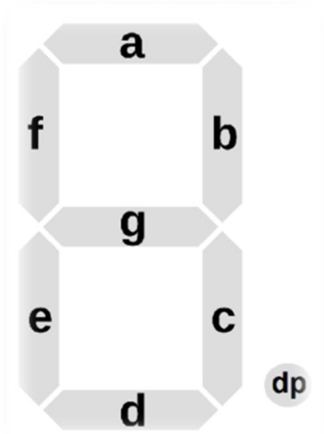


Figure 6. Sample segment encoding with 7-segment display

2.5 Software Drivers

RA FSP drivers for segment LCDs simplify the process of communicating with the display hardware by abstracting low-level register configurations. Instead of directly managing intricate hardware interactions, developers can use higher-level APIs provided by FSP to configure and control the LCD with ease.

These APIs handle tasks such as initializing the LCD, setting display modes, controlling segments, and updating content without requiring developers to manually manipulate hardware registers. This approach not only reduces development time but also minimizes errors related to low-level hardware programming.

FSP provides a set of APIs at the application level that allow developers to interact with the segment LCD module. These APIs enable essential operations like enabling or disabling the display, configuring contrast settings, and updating display segments, thereby offering a flexible and user-friendly interface for integrating segment LCD functionality into embedded applications.

FSP provided SLCD API	Description
R_SLCDC_Open()	Opens the LCD module and configures the controller for different modes chosen by the user as part of the configurator.
R_SLCDC_Write()	Write a sequence of display data to the segment data registers.
R_SLCDC_Modify()	Modifies a single segment register based on a mask and the desired data
R_SLCDC_Start()	Starts output of LCD signals
R_SLCDC_Stop()	Stops output of LCD signals
R_SLCDC_SetContrast()	Sets contrast to the specified level
R_SLCDC_SetDisplayArea()	Sets output to Waveform A, Waveform B, or blinking output
R_SLCDC_Close()	Closes the SLCDC driver

2.5.1 Segment LCD Programming

Segment LCD programming involves controlling individual segments of an LCD to display characters, symbols, or numeric data. Unlike graphical LCDs that use a pixel grid, segment LCDs rely on predefined segments that are turned ON or OFF to create readable outputs. To achieve this, developers must configure and program the RA4L1 using the FSP configuration tool and the SLCD driver.

The RA4L1 MCU provides a range of configuration options to control the LCD module effectively. These options allow developers to define the behavior of the LCD in terms of power efficiency, contrast, refresh rate, and display clarity.

For the EK-RA4L1 evaluation kit, the onboard segment LCD module operates using four COM (common) pins. The BSP within the FSP automatically handles pin configurations based on the hardware design, ensuring correct routing of signals to the LCD.

2.5.1.1 SLCD Driver Initialization and Configuration

To begin segment LCD operation, the SLCD driver must be initialized. This is done using the R_SLCDC_Open() API. When this function is called, the driver configures the LCD according to the parameters set in the FSP configurator. These parameters include:

1. LCD Display Bias Method: Defines the drive scheme used to apply voltages to LCD segments for uniform brightness and clarity.
2. Time Slice Selection: Controls how the display is refreshed in a time-multiplexed manner.
3. Display Waveform Selection: Determines the waveform used to drive LCD segments, impacting power efficiency and display stability.
4. LCD Drive Voltage Generator Settings: Configures how the voltage for driving the LCD is generated, which affects contrast and power consumption.
5. LCD Clock Configuration: Sets the timing source for the LCD operation, impacting refresh rate and stability.
6. Voltage Boost Level Adjustment: Adjusts the contrast level of the display by increasing or decreasing the drive voltage.

2.5.1.2 Starting and Stopping the LCD Display

Once the LCD is configured, the APIs R_SLCDC_Start() and R_SLCDC_Stop() can be used to turn the LCD display on or off. This provides control over power consumption and display activation based on application needs.

2.5.1.3 Controlling Display Output Mode

The SLCD driver also includes an API called R_SLCDC_SetDisplayArea(), which allows developers to configure how the LCD segments behave. This API provides different output modes:

- Waveform A – A standard waveform for segment control.

- Waveform B – An alternative waveform that may improve display performance under different conditions.
- Blinking Mode – Enables segment blinking, which is useful for drawing attention to certain data or alerts.

2.5.2 Writing Data to Segment LCDs

To control a segment LCD, data is written to segment registers of the segment LCD controller directly. The controller interprets this data and activates the corresponding segments to create the desired display. The data must be structured in a format that matches the LCD segment addressing scheme. The segment registers start at address 0x400D_4100 and extend to 0x400D4133. The FSP driver provides API **R_SLCDC_Write()** to write the data to these segments.

2.5.3 Controlling Individual Segments

Control of the individual segments of an LCD is done by a specific pin or register. Toggling the corresponding data bits or updating the corresponding register states turns individual segments on or off. This requires careful mapping between the MCU's output pins or register bits and the physical segments of the LCD. Care must be taken not to modify the different segments that are driven by the same register. Proper software design at the application level ensures that the correct segments are illuminated to display the intended characters or symbols. **R_SLCDC_Modify()** is used with proper mask bits to modify the particular segments.

2.6 Configuring the RA4L1 LCD Controller for the Sample Application

2.6.1 Hardware Setup

It is important to be familiar with the EK-RA4L1 MCU board being used, including the pin configurations, power supply requirements, and peripheral connections. Understanding how the segment LCD controller is integrated into the MCU hardware and how signals are routed ensures proper communication between the MCU and the LCD module.

2.6.2 Pin Settings

The Pin Configuration tab allows you to assign roles to the available I/O pins on your target RA MCU for communication with system peripherals. For the SLCDC module, pin assignments must be configured according to the hardware schematics. On the EK-RA4L1 board, the BSP provides default pin settings based on the board design. However, when developing applications, it is recommended to review these settings to ensure they match the hardware schematics.
Note: When designing a custom board, the pin configurations in the custom BSP must be properly managed to ensure correct operation.

2.6.3 Clock Settings

The Clocks configuration tab sets up the internal and external clocking on the MCU for the target application. Clock sources and frequency divider settings can be selected for each of the clocks on the MCU. For full details on the Clock Generation Circuit (CGC), refer to the RA4L1 Group’s Hardware Users’ Manual.

2.6.4 BSP Property Settings

EK-RA4L1 BSP comes with default settings. Users can change these settings based on requirements. Some of them can be Stack settings, Heap settings, etc.

2.7 FSP Integrated SLCDC Module(r_slcdc) Configuration

2.7.1 Configuring the RA4L1 SLCD Module

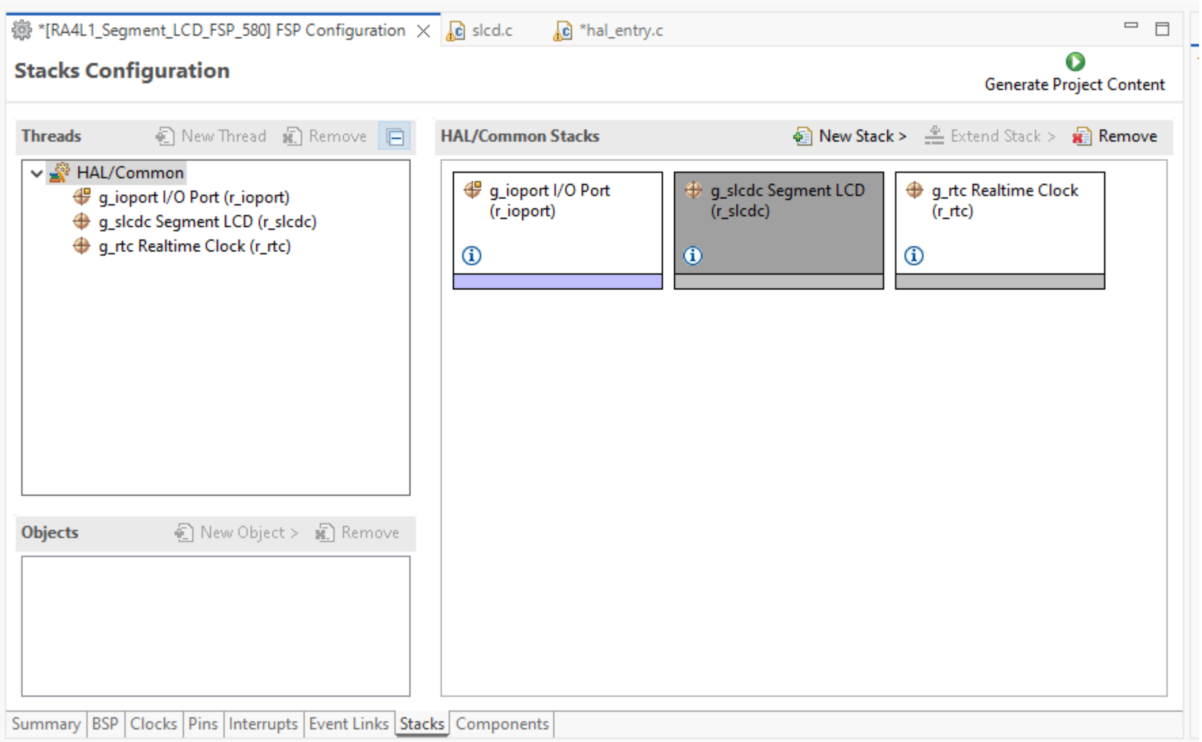


Figure 7. Adding and configuring the SLCDC module for the application

After adding the r_slcdc module to the project using the configurator, the module needs to be configured based on the hardware and application requirements. This can be achieved using the properties tab of the module stack, as shown in Figure 7. Configuring SLCDC Module property settings.

2.7.2 SLCD module property settings

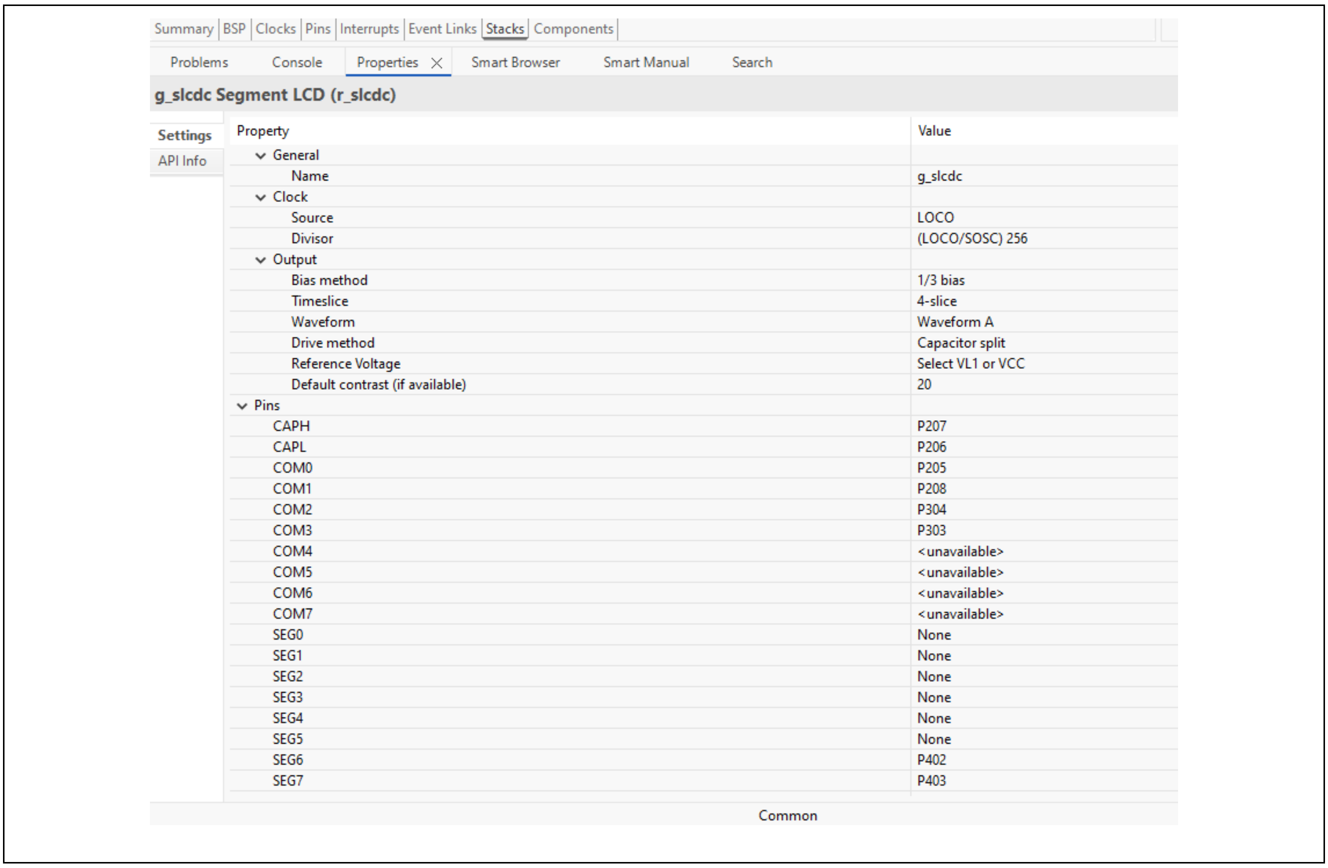


Figure 8. Configuring SLCDC module property settings

Under the property settings for the r_slcdc module, default settings are listed. Users can change these settings based on the Hardware and application requirements. The below sections give details on each property and its selection.

2.7.2.1 Clock Source Selection

The FSP configurator provides different clock sources, such as LOCO, MOCO, SOSC, and MOSC, for the LCD clock. Users can configure the desired clock and its frequency based on the divisor setting in the configurator.

2.7.2.2 Bias Methods Selection

Segment LCDs require an AC voltage drive to prevent the liquid crystal material from deteriorating over time. Biasing helps distribute the voltage levels effectively, reducing power consumption while maintaining good contrast.

Bias mode is defined by the ratio of applied voltage levels in a multiplexed LCD. The number of bias levels depends on the LCD's multiplexing scheme. Based on the Hardware design and LCD module selected, the bias mode can be configured using the FSP configurator. RA4L1 provides 1/2, 1/3, and 1/4 bias methods for using the FSP configurator.

2.7.2.3 Time Slice Methods Selection

The number of time slices (N) is directly related to the multiplex ratio (1:N), where N is the number of common (COM) lines. Higher multiplexing allows for more display segments while using fewer pins.

Table 1. Time Slice Methods Selection

Multiplex Mode	Time Slices (N)	Description
Static (1:1 Multiplex)	1 Time Slice	Each segment has a dedicated pin. No multiplexing is used in this case.
1/2 Duty Cycle (1:2 Multiplex)	2 Time Slices	Two COM lines share segment control, reducing driver count.
1/3 Duty Cycle (1:3 Multiplex)	3 Time Slices	Three COM lines are used. Requires 1/3 biasing for proper contrast.
1/4 Duty Cycle (1:4 Multiplex)	4 Time Slices	Four COM lines are used. Requires 1/3 or 1/4 biasing. Used in small and medium displays.
1/6 Duty Cycle (1:6 Multiplex)	6 Time Slices	Six COM lines are used. Requires 1/3 or 1/4 biasing. Used in small and medium displays.
1/8 Duty Cycle (1:8 Multiplex)	8 Time Slices	Eight COM lines are used. Used in complex LCD panels.

2.7.2.4 Waveform Selection

When working with a segment LCD controller, waveform A and waveform B refer to different drive schemes used to control the liquid crystal display. These waveforms define how voltage is applied to LCD segments and common terminals to properly drive the display without causing DC bias issues.

RA4L1 provides waveform A and waveform B as the option. This again depends on the HW and SW design being used in the LCD application.

Table 2. Waveform Selection

Feature	Waveform A (Frame Inversion)	Waveform B (Line Inversion)
Driving Scheme	Alternates per frame	Alternates per line
Flicker Reduction	Better	Slightly less effective
Power Consumption	Moderate	Lower
Usage	Low-multiplex LCDs	High-multiplex LCDs

2.7.2.5 Driver Method Selection

When configuring an LCD Controller Driver in RA4L1, selecting the appropriate LCD drive method is crucial for optimizing power consumption, contrast, and display quality. The power supply voltages for the LCD driver can be produced through:

- Capacitor Split Method (Capacitive Voltage Division).
- Internal Voltage Boosting (Charge Pump/Voltage Multiplier).
- External Resistance Division (Resistor Ladder Bias).

Table 3. Driver Method Selection

Drive Method	Power Consumption	Voltage Stability	Best suited for
Capacitor Split	Low	Moderate	Low-power, simple LCDs
Internal Voltage Boosting	Medium	High	Multiplexed LCDs with high contrast
External Resistance Division	High	Very High	High-multiplex, large displays

Using the FSP configurator, driver methods can be selected based on the design being used in the hardware and application.

2.7.2.6 Reference Voltage Selection

The power supply voltages for the LCD driver can be produced through the external resistance division method, internal voltage boosting method, or capacitor split method. Refer to section 46.6 of the UM for more details.

(1) External Resistance Division Method

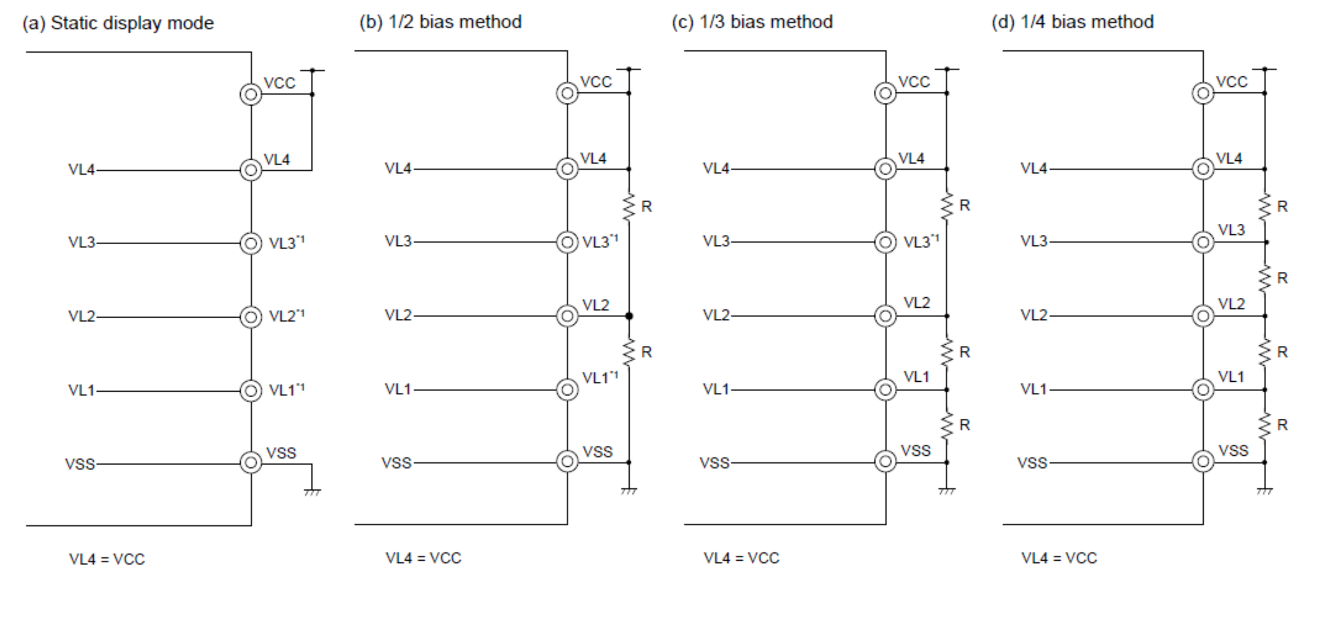
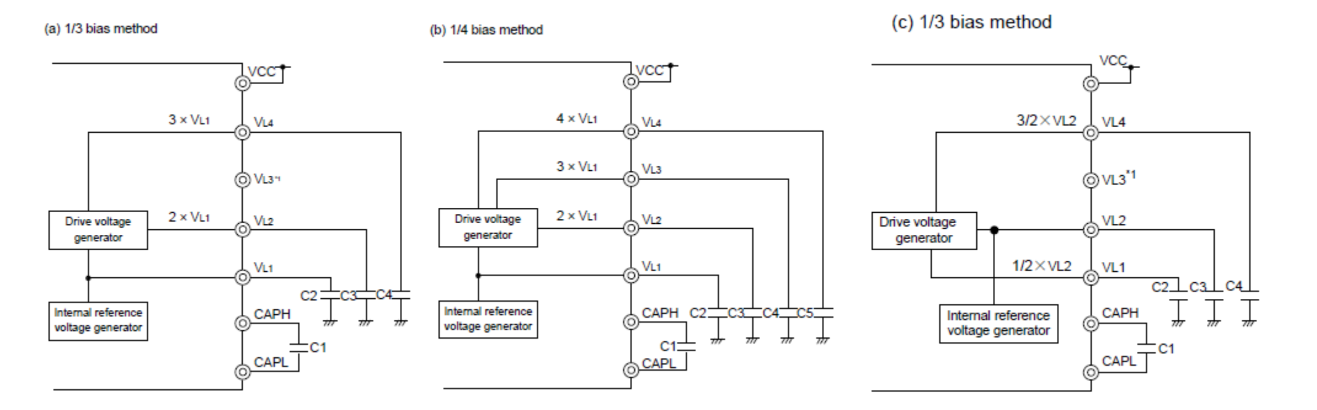


Figure 9. LCD drive power connections using the external resistance division method

(2) Internal Voltage Boosting Method

The MCU includes an internal voltage boost circuit for generating LCD drive power supplies. This circuit, along with external capacitors (0.47 μF ± 30%), is used to produce the LCD drive voltage. For the internal voltage boosting method, only the 1/3 or 1/4 bias mode can be set for VL1 reference mode, while only the 1/3 bias mode can be set for VL2 reference mode. Since the internal voltage boost circuit operates as an independent power supply separate from the main unit, it can maintain a constant voltage regardless of fluctuations in VCC.



Note: Use a capacitor with as little leakage as possible. In addition, make C1 a non-polar capacitor.
Note 1. VL3 can be used as a port.

Figure 10. LCD drive power connections using VL1 reference mode of internal voltage boosting

LCD drive voltage pin	1/3 bias method		1/4 bias method
	VL1 reference	VL2 reference	VL1 reference
VL4	3 × VL1	3/2 × VL2	4 × VL1
VL3	—	—	3 × VL1
VL2	2 × VL1	LCD reference voltage	2 × VL1
VL1	LCD reference voltage	1/2 × VL2	LCD reference voltage

(3) Capacitor Split Method

The MCU features an internal voltage reduction circuit for generating LCD drive power supplies. This circuit, in combination with external capacitors (0.47 μF ± 30%), is used to produce the LCD drive voltage. In the capacitor split method, only the 1/3 bias mode can be set for both VCC reference mode and VL4 reference mode. Unlike the external resistance division method, the capacitor split method eliminates the need for continuous current flow, thereby reducing power consumption.

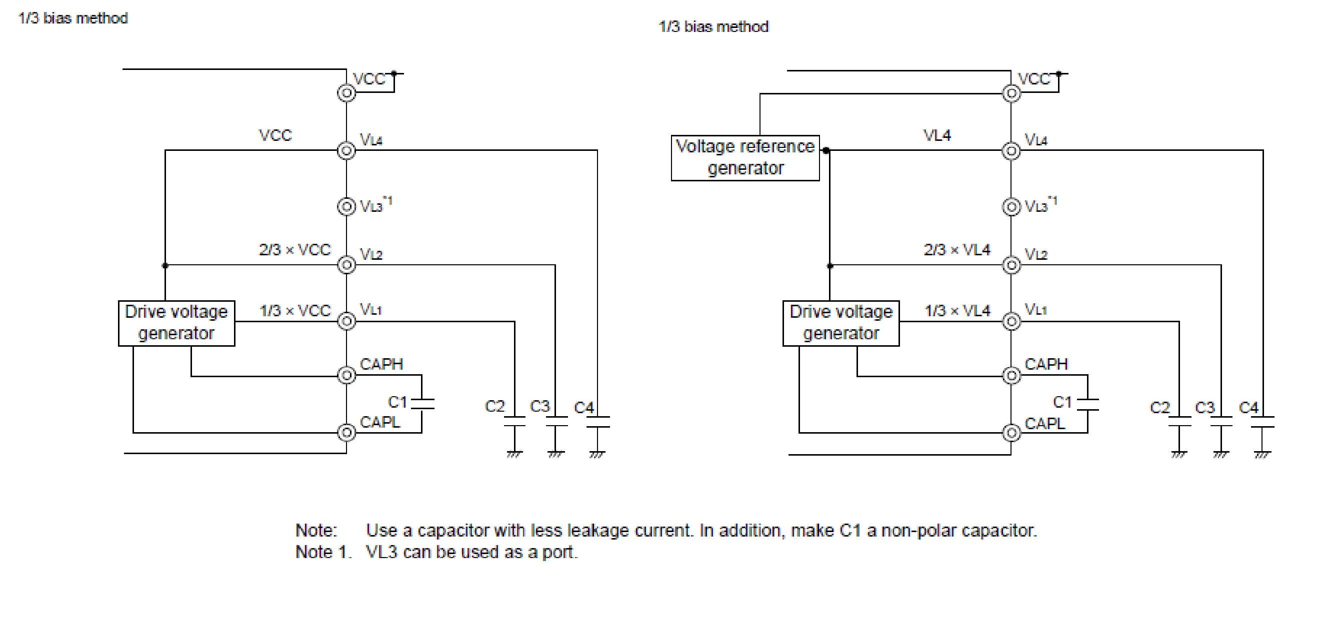


Figure 11. LCD drive power connections using VCC and VL4 reference mode of capacitor split method

LCD drive voltage pin	1/3 bias method	
	VCC reference	VL4 reference
VL4	VCC	LCD reference voltage
VL3	—	—
VL2	2/3 × VL4	2/3 × VL4
VL1	1/3 × VL4	1/3 × VL4

2.7.2.7 Contrast Level Selection

Contrast adjustment in a segment LCD controller determines the visibility of segments by controlling the voltage difference between the common (COM) electrodes and segment (SEG) electrodes. Proper contrast adjustment ensures clear, sharp, and flicker-free display operation.

- The contrast is controlled by the bias voltage applied across the liquid crystal material.
- Higher voltage → Darker Segments (More contrast).
- Lower voltage → Lighter Segments (Less contrast).

Table 46.11 List of combinations of contrast adjustment when {MDSET2,MDSET[1:0]} = 001 (VL1 voltage reference)

VLCD [4:0]					VL1 voltage reference	VL4 voltage	
						1/3 bias method	1/4 bias method
0	0	1	0	0	1.01 V	3.03 V	4.04 V
0	0	1	0	1	1.04 V	3.12 V	4.16 V
0	0	1	1	0	1.07 V	3.21 V	4.28 V
0	0	1	1	1	1.11 V	3.33 V	4.44 V
0	1	0	0	0	1.14 V	3.42 V	4.56 V
0	1	0	0	1	1.17 V	3.51 V	4.68 V
0	1	0	1	0	1.21 V	3.63 V	4.84 V
0	1	0	1	1	1.24 V	3.72 V	4.96 V
0	1	1	0	0	1.27 V	3.81 V	5.08 V
0	1	1	0	1	1.31 V	3.93 V	5.24 V
0	1	1	1	0	1.34 V	4.02 V	Setting prohibited
0	1	1	1	1	1.37 V	4.11 V	
1	0	0	0	0	1.40 V	4.20 V	
1	0	0	0	1	1.44 V	4.32 V	
1	0	0	1	0	1.47 V	4.41 V	
1	0	0	1	1	1.50 V	4.50 V	
1	0	1	0	0	1.54 V	4.62 V	
1	0	1	0	1	1.57 V	4.71 V	
1	0	1	1	0	1.60 V	4.80 V	
1	0	1	1	1	1.64 V	4.92 V	
1	1	0	0	0	1.67 V	5.01 V	
1	1	0	0	1	1.70 V	5.10 V	
1	1	0	1	0	1.74 V	5.22 V	

Note: Other settings are prohibited.

Figure 12. Contrast level settings and their voltages

3. The Example Application

3.1 Overview of RA4L1 LCD Controller as a Thermostat

The example application project provided with this application note demonstrates the capabilities of the RA4L1 microcontroller's built-in LCD controller by simulating a thermostat system. This application shows how the MCU interacts with an LCD module to display real-time temperature readings, control heating and cooling indicators, and show additional information such as time and days of the week.

Temperature Sensing and Display

The RA4L1 MCU includes an integrated temperature sensor that continuously monitors the ambient temperature. The measured temperature value is processed and displayed on the LCD module’s big digit section. The LCD controller dynamically updates the MCU die temperature reading in real time as the sensor detects variations in the surrounding environment of the MCU die.

Heating and Cooling Indicators

To simulate thermostat functionality, the application incorporates heating and cooling indicators based on predefined temperature thresholds:

- **Cooling Mode Activation**: If the temperature reading exceeds 80°F, the "Cool" icon on the LCD is turned ON, indicating that cooling is required.
- **Heating Mode Activation**: If the temperature drops below 60°F, the "Heat" icon on the LCD is turned ON, signaling that heating is needed.

This behavior demonstrates how the RA4L1 MCU can control and display elements dynamically, making it suitable for real-world thermostat applications.

Real-Time Clock (RTC) Display

In addition to temperature monitoring, the application also leverages the RA4L1's real-time clock (RTC) functionality. The current time is displayed on the LCD module, ensuring users can keep track of time while monitoring temperature changes. Furthermore, the day of the week is displayed alongside the time, enhancing the utility of the application.

3.2 Software Overview

Application Flow and Functionality Overview

The thermostat application, built using BareMetal code, is designed to interact with various hardware modules on the RA4L1 MCU, such as the LCD, ADC, and RTC, to simulate a real-time temperature and time display. The system operates without the overhead of an operating system, directly interacting with the hardware to perform essential tasks for thermostat control. Below is an in-depth explanation of how the application functions:

(1) System Initialization on Power-Up

Upon power-up, the following modules are initialized sequentially:

- **LCD Module**: The LCD controller is initialized, and the contrast level is adjusted to ensure optimal display visibility. Specific icons such as the Renesas logo and Battery level indicator are displayed to give users a clear indication of the system's status.
- **ADC**: The ADC module is initialized to read temperature data from the on-chip temperature sensor, which is used to monitor the ambient temperature and make decisions based on predefined thresholds.
- **RTC**: The RTC module is initialized, and a user-configured RTC time value is configured into the application. This value is written in the RTC registers using the appropriate FSP API. This provides the system with a starting time, which is used to track the current time and display it on the LCD.

(2) User-Configured Temperature Thresholds

The application allows the user to set upper and lower temperature limits. These values define the temperature ranges at which the thermostat should trigger certain actions:

- **Upper temperature limit**: When the temperature exceeds this threshold (For example, 80°F), the Cool icon is displayed on the LCD to indicate that cooling is required.
- **Lower temperature limit**: If the temperature falls below this threshold (For example, 60°F), the Heat icon is displayed on the LCD to indicate that heating is needed.

These thresholds are stored as constants or variables in the application code, allowing for easy adjustments based on the user's preferences.

(3) Periodic Checks and Interrupt-Driven Updates

The program monitors changes in both the ADC value (temperature reading) and RTC value (time) using periodic RTC interrupt. These checks are periodic, and the system updates the LCD display based on the state changes of these modules.

- **RTC interrupt**: When the RTC value is updated using interrupt. An interrupt service routine (ISR) is triggered. The ISR takes the raw RTC value, converts it into a human-readable HH:MM (hours and minutes) format, and updates the corresponding fields on the LCD. The day of the week and AM/PM indicators are also updated in real time, ensuring the time displayed is accurate and current.
- **ADC**: The raw ADC value is also read with periodic interrupts and then converted into an actual temperature reading (typically in Celsius or Fahrenheit). This value is displayed on the LCD, allowing the user to track the ambient temperature.

(4) Dynamic Display Updates

The application continuously updates the display based on changes in temperature and time. The LCD shows the following data:

- **Time display**: The current time is updated every time the RTC triggers an interrupt. The HH:MM format is used to display the time, along with the day of the week and AM/PM status, ensuring that users can keep track of both the time and the date briefly.
- **Temperature display**: As the ADC value changes, the temperature reading is converted from the raw ADC value into a readable format (For example, Fahrenheit or Celsius). This updated temperature is displayed prominently on the LCD.
- **Thermostat indicators**: Based on the temperature reading, the system updates the thermostat indicators:
  - If the temperature exceeds the upper threshold, the Cool icon is turned ON to indicate that cooling should be activated.
  - If the temperature falls below the lower threshold, the Heat icon is turned ON to indicate that heating should be activated.

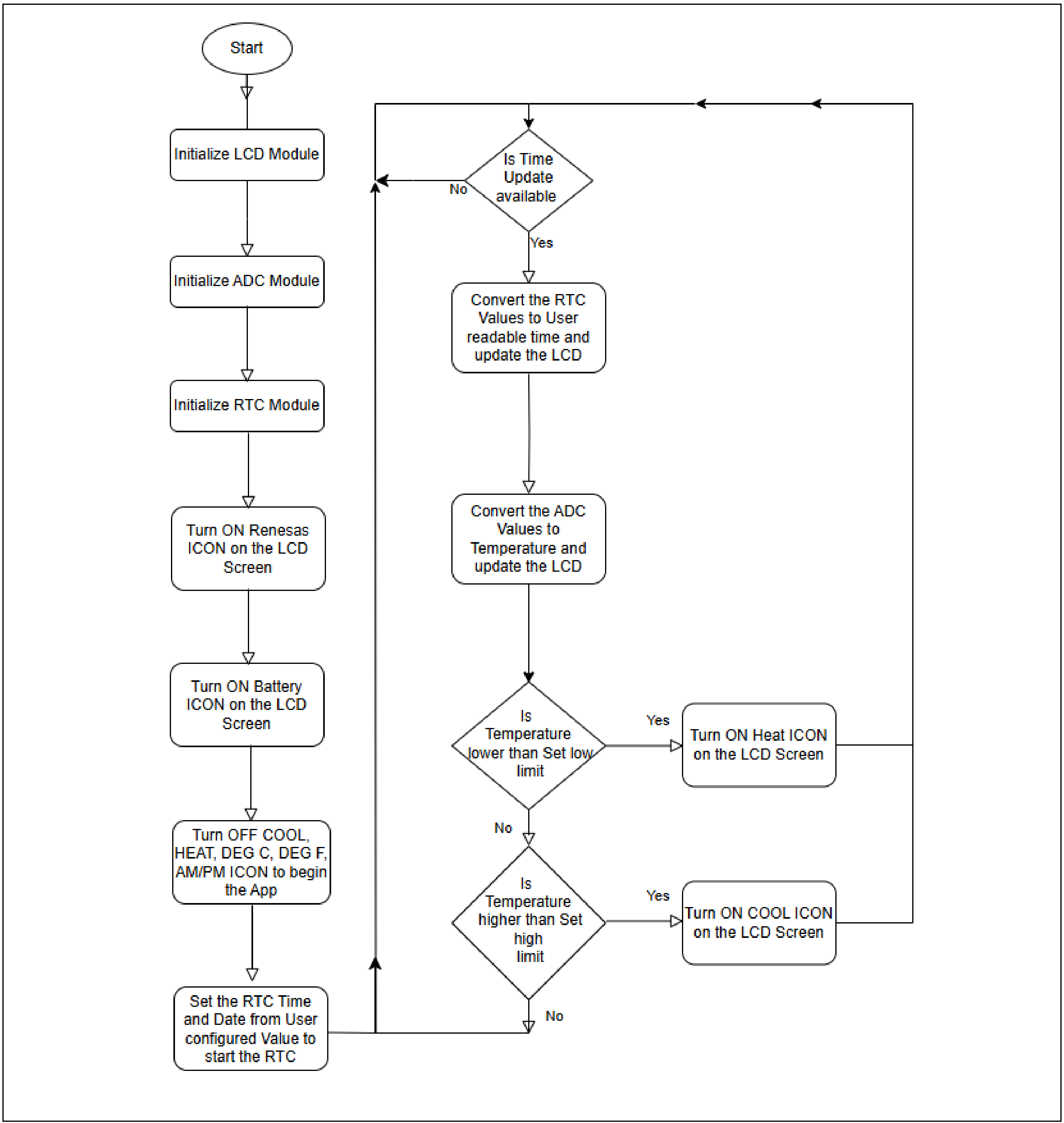


Figure 13. Example Application flow diagram

4. Running the Example Application Project

4.1.1 Connecting the MCU to the LCD

Connect the J1 connector of the LCD module to the J2 connector of the EK-RA4L1 (pin 0 of the LCD module is connected to pin 0 of the board). The LCD module extends outside the board, as shown in Figure 13. Mounted LCD module on the EK-RA4L1 Board.

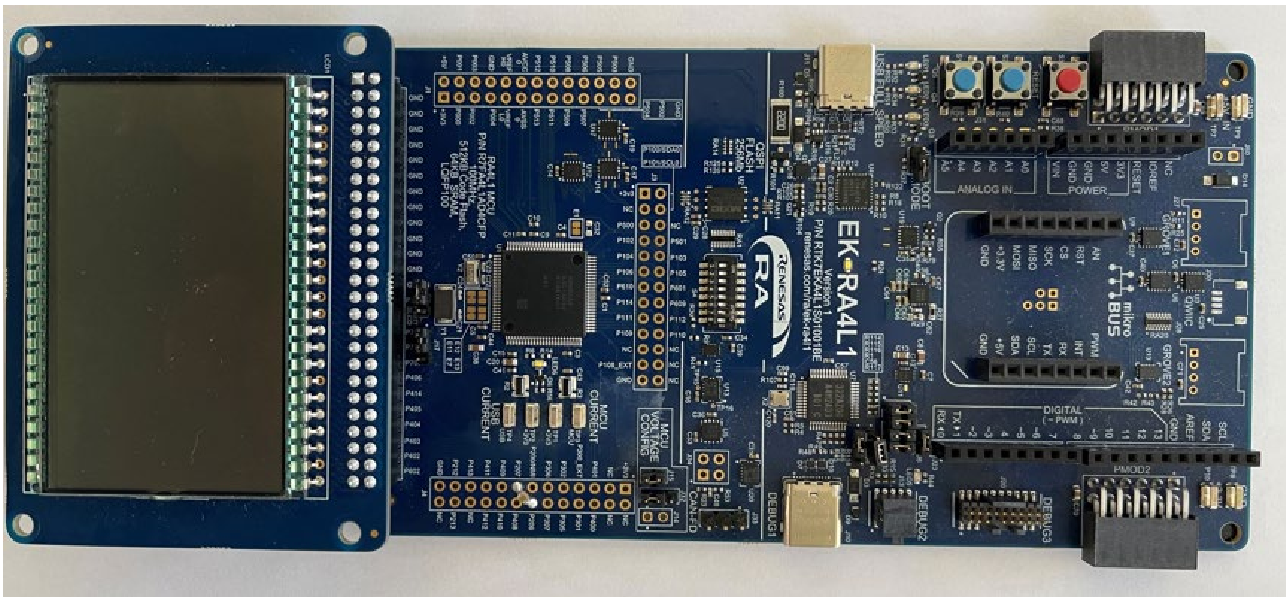


Figure 14. Mounted LCD module on the EK-RA4L1 board

Table 4. Jumper Settings

Jumper	Connection
J7	Open
J17	Connect 1-2

The location of J7 and J17 can be seen above the LCD connector, as shown in Figure 14. Locating J7 and J17 Jumpers on the EK-RA4L1 Board.

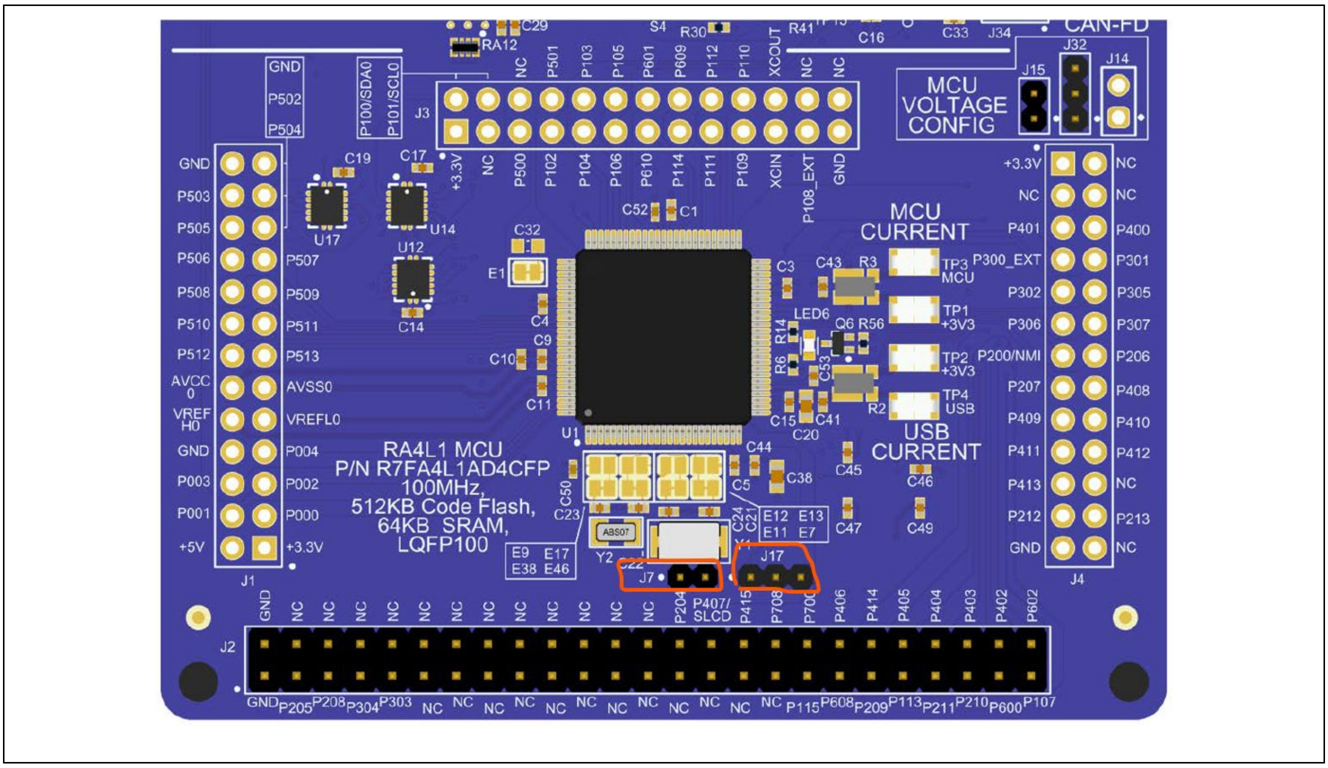


Figure 15. Locating J7 and J17 Jumpers on the EK-RA4L1 board

4.1.2 Import, Build, and Run

1. Ensure the application project folder, located under *r11an0xxx/ RA4L1_Segment_LCD* is downloaded to your host machine.
2. Connect the EK-RA4L1 USB-C debug port (J10 - DEBUG1) to your host machine with the USB-C cable.
3. Open an instance of e² studio IDE.
4. In the workspace launcher, browse to the workspace location of your choice and select it.
5. In e² studio, navigate to File → Import.
6. In the Import dialog box, select General → Existing projects into workspace → Next.
7. Click select archive file and use the browse button to point to the location of the RA4L1_Segment_LCD.zip file.
8. Make sure the option to copy projects into the workspace is selected. Click Finish.
9. Open the *configuration.xml* file in the Project Explorer view and click Generate Project Content.
10. Build the project.
11. Debug and run the project.
12. After running the application, open the J-Link RTT Viewer to see the output of time and temperature displayed on the LCD Display. RTT viewer displays the user level informational and error messages as part of the initialization and execution. Follow the steps below to open and run the RTT viewer.
    a. Open the Segger J-Link RTT Viewer.
    b. File → Connect pops up a configuration window as shown in Figure 15. RTT Viewer configuration.
    c. Specify the connection to J-Link as “**USB**”.
    d. Specify the Target device as “**R7FA4L1AD”** (RA4L1).
    e. Target Interface and Speed as **SWD** and **4000Khz.**
    f. RTT Control Block as **Address** and enter the address obtained from the map file as shown in Figure 16. RTT Control Block address selection from map file and press OK.

g. You will see the RTT Output on the RTT terminal.

The snapshot of the RTT viewer and the LCD display are shown below.

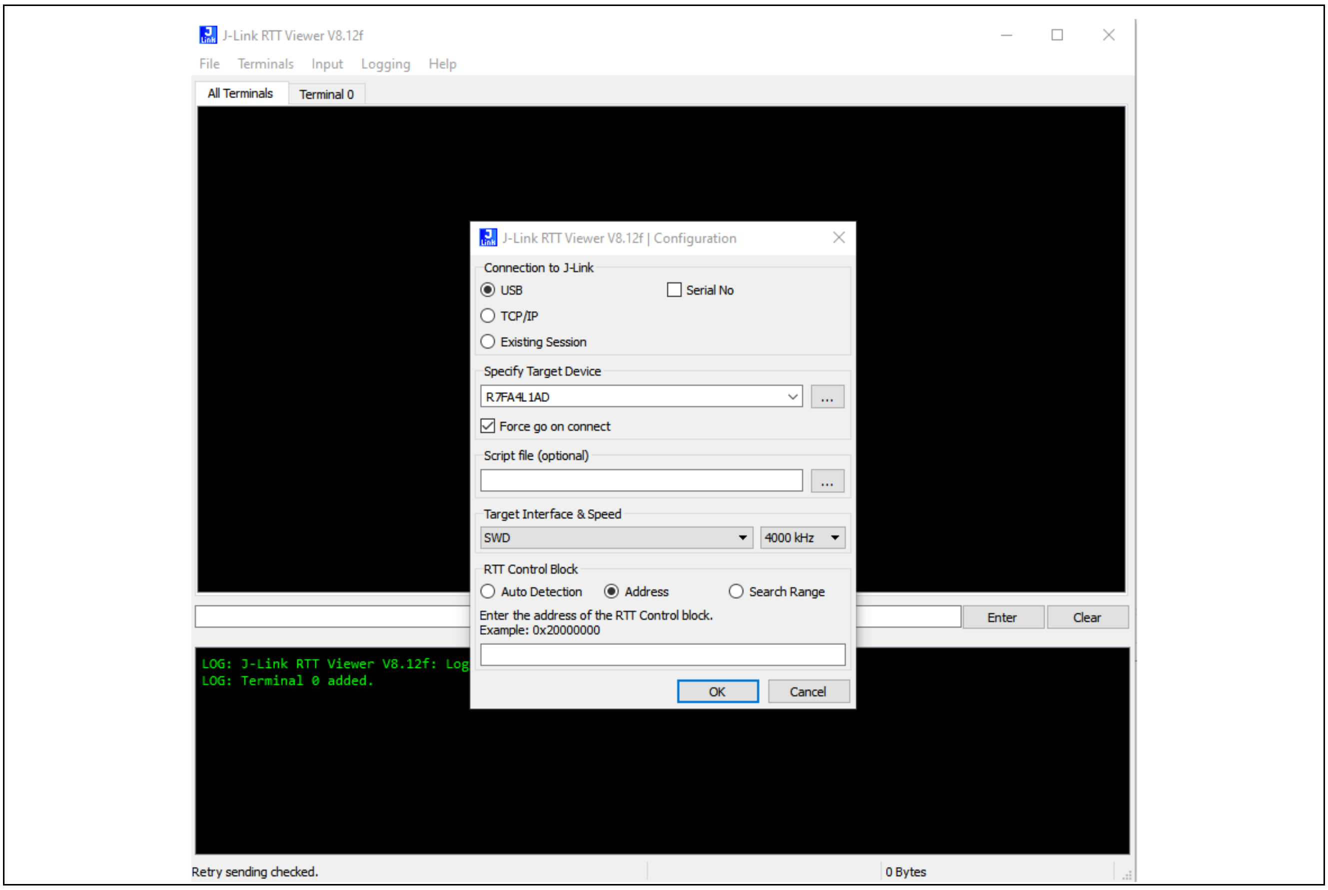


Figure 16. RTT Viewer configuration

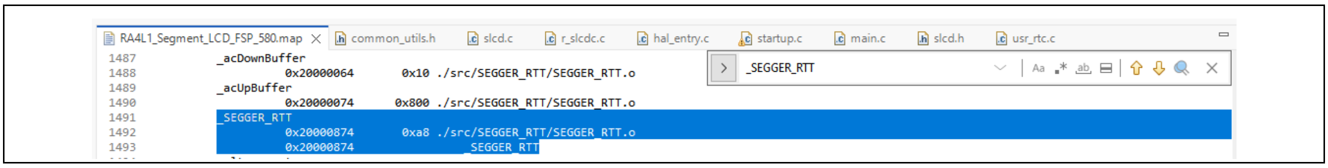


Figure 17. RTT Control Block address selection from map file

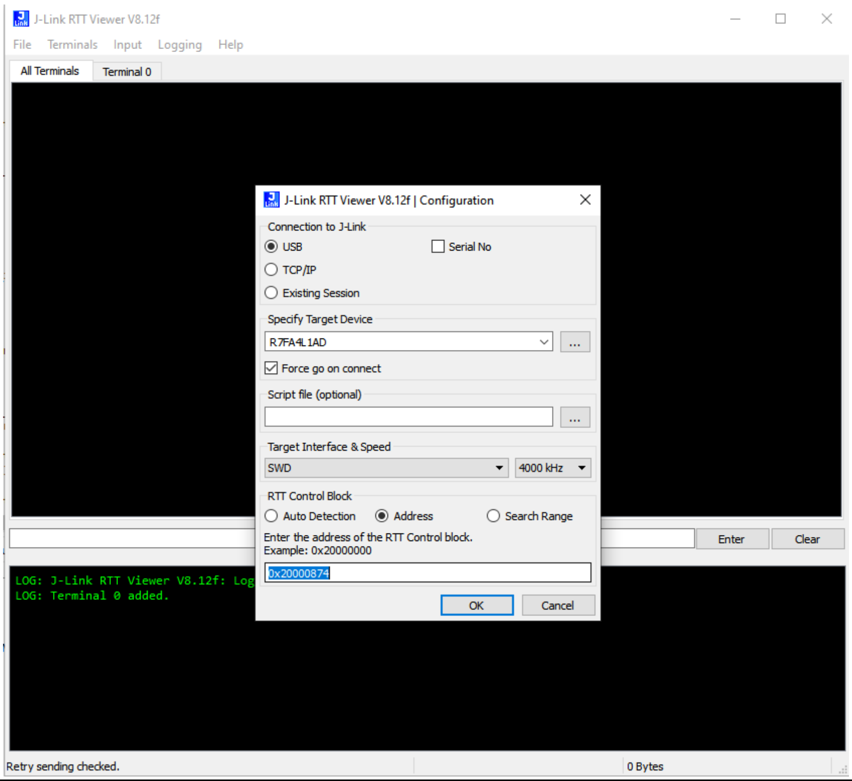


Figure 18. RTT Viewer configuration sample

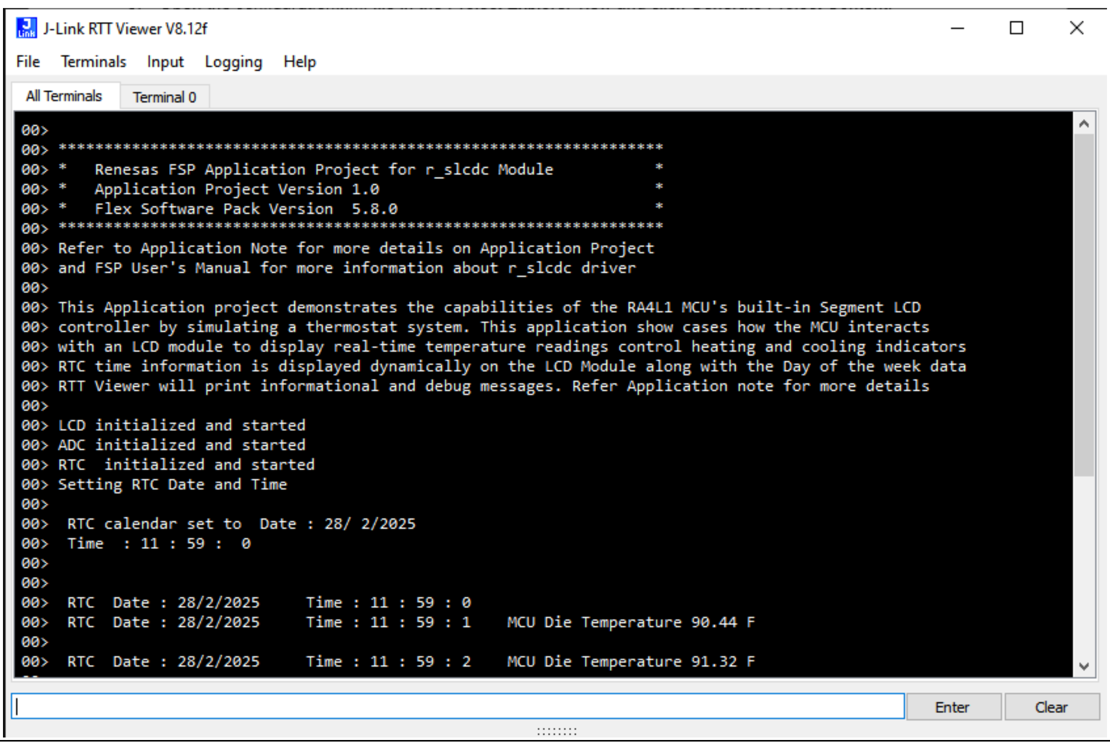


Figure 19. RTT Viewer output

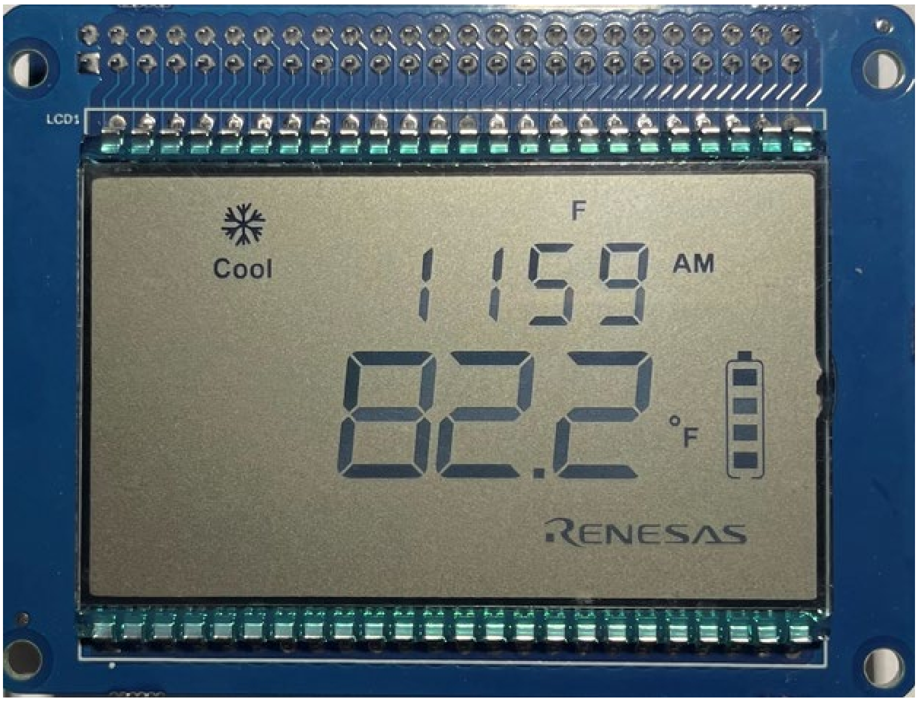


Figure 20. LCD application running on the EK-RA4L1 board with Cooling Icon

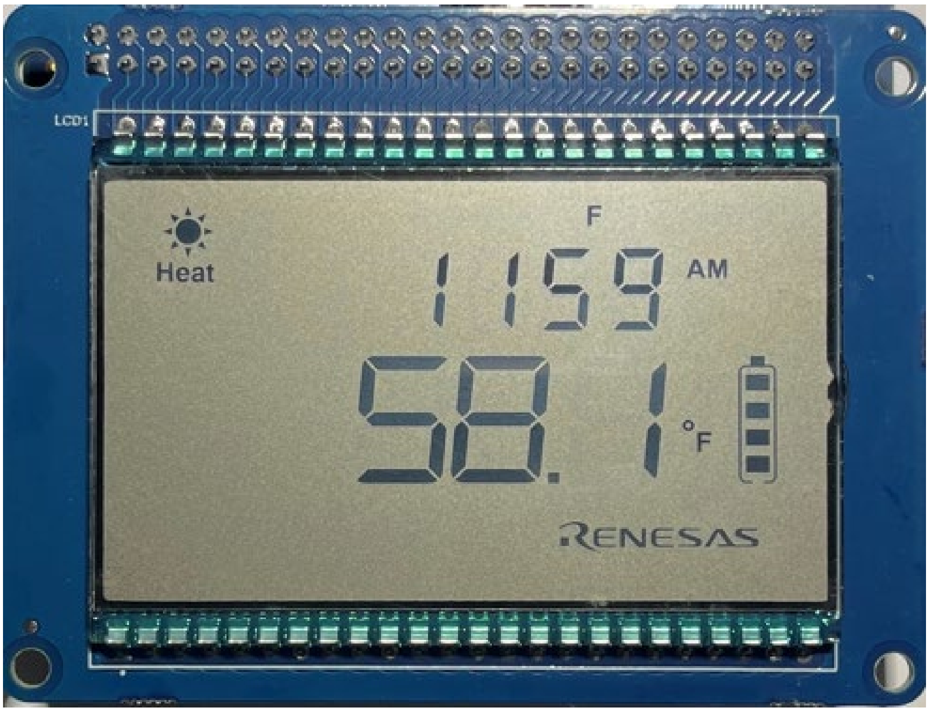


Figure 21. LCD application running on the EK-RA4L1 board with Heating Icon.

5. Debugging and troubleshooting

When working with segment LCDs on the RA4L1 MCU, issues can arise due to incorrect configurations or hardware connections. Below are key areas to check when debugging LCD display problems:

1. Ensure the LCD module is securely connected to the evaluation board or custom PCB.

2. Check for loose or misaligned connectors that might cause some segments not to function.
3. Use a multimeter to verify that signals are properly reaching the LCD pins.
4. Verify that all COM and SEG pins are correctly configured in the FSP configurator.
5. Ensure that the correct pin functions are assigned in the Pin Configuration settings within FSP.
6. Double-check that no other peripherals are using the same pins as the LCD signals.
7. Each LCD segment is mapped to a specific MCU port pin. If a segment is not displaying properly, confirm the segment-to-pin mapping in the hardware reference manual.
8. Ensure that the segment mapping in FSP matches the LCD datasheet specifications for the proper display of characters and symbols.
9. If the display appears too dim or too bright, adjust the contrast level by configuring the voltage boost settings in the SLCD driver.
10. Ensure that the LCD drive reference voltage generator is correctly configured in FSP, as shown in the section 2.6.2.
11. Check if the power supply voltage is within the expected range for proper contrast operation.
12. The selected LCD clock source is stable and enabled.
13. The clock divider settings are appropriate for the display's refresh rate.
14. The configured frame frequency is within the recommended range to avoid flickering.

6. Conclusion

This application note provides a comprehensive guide to working with segment LCDs using RA4L1 MCUs, covering both hardware and software aspects. It detailed the segment LCD controller features, the process of interfacing and driving segment LCDs, and the configuration of the SLCD module using the FSP.

Additionally, the document explored software drivers, segment control, and data writing techniques, along with a sample thermostat application demonstrating practical implementation. Steps for hardware setup, pin mapping, clock configuration, and property settings were also outlined to ensure proper LCD operation.

Finally, a debugging and troubleshooting section provided solutions to common issues, ensuring a smooth development process. Developers can follow the guidelines in this document to effectively integrate and control segment LCDs in RA4L1-based applications with ease and reliability.

7. Next Steps

- After you run the example application, you can learn more about how the application works and the API calls involved by examining the application source code.
- By following the guidelines provided in the App note, you can efficiently leverage the Segment LCD application and the provided app note from Renesas to implement and configure the SLCDC in your design.
- Visit renesas.com/ra/RA4L1 for more information about the EK-RA4L1 example kit, including its Quick Start Guide, design data, ordering information, and other useful application projects.
- Refer to the following GitHub repository for various FSP modules example projects and application projects to evaluate the RA4L1 MCU(https://github.com/renesas/ra-fsp-examples/)
- Contact the Renesas Sales and support team to get more information on the RA4L1 MCU and support needed as part of the evaluation.

8. Known Issues

This section talks about the known FSP and tool-related issues. More details can be found at this link (https://github.com/renesas/fsp/issues).

9. References

- Renesas FSP User's Manual: https://renesas.github.io/fsp
- Renesas RA MCU User's Manual: ra4l1-group-users-manual-hardware
- SLCD Example Projects on Renesas RA GitHub: https://github.com/renesas/ra-fsp-examples
- EK-RA4L1 Board Schematics: ek-ra4l1-v1-design-package
- EK-RA4L1 Getting Started Guide: ek-ra4l1-v1-quick-start-guide
- EK-RA4L1 Hardware Users Guide: ek-ra4l1-v1-users-manual
- Segment LCD Module Guide: Renesas Starter Kit for LCDAPPV2 User's Manual

10. Website and Support

Visit the following vanity URLs to learn about key elements of the RA family, download components and related documentation, and get support.

RA Product Information	www.renesas.com/ra
RA Product Support Forum	www.renesas.com/ra/forum
RA Flexible Software Package	www.renesas.com/FSP
Renesas Support	www.renesas.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar.04.25	—	First release document

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

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/.