7-segment Display using PmodSSD

SLG47910

Abstract

This application shows how to display a 2-digit counter on the 7-segment display using a PmodSSD. This application note comes complete with design files which can be found in the References section.

Contents

1. Terms and Definitions

FPGA	Field Programmable Gate Array
FPGA Editor	Main FPGA design and simulation window
Go Configure Software Hub	Main window for device selection
ForgeFPGA Window	Main FPGA project window for debug and IO programming

2. References

For related documents and software, please visit: https://www.renesas.com/. Download our free ForgeFPGA™ Workshop software [1] to open the .ffpga design file [2] and view the proposed circuit design.

[1] ForgeFPGA Workshop Software, Software Download and User Guide

[2] AN-FG-010 7-Segment Display using PMODSSD.ffpga, ForgeFPGA Design File

[3] SLG47910, Datasheet, Renesas Electronics

[4] PmodSSD Reference Guide, Digilent

3. Introduction

In this Application note we intend to create a counter that counts from 0 to 99 in 100 seconds. Each count is displayed for one second. The design consists of three Verilog modules. The counter outputs are displayed on the seven-segment PmodSSD utilizing the in-built IP block. Let us discuss each module in detail and understand how they are all connected to each other.

We have three sub-modules called counter_1s, dynamic_indication and timer_FSM. Each block has been created for a specific purpose and all these three sub-modules are connected to each other to form the top module. In Figure 1, the connections between the three Verilog block are shown. The user can cross check the mentioned signal and wire names with the Verilog Code.

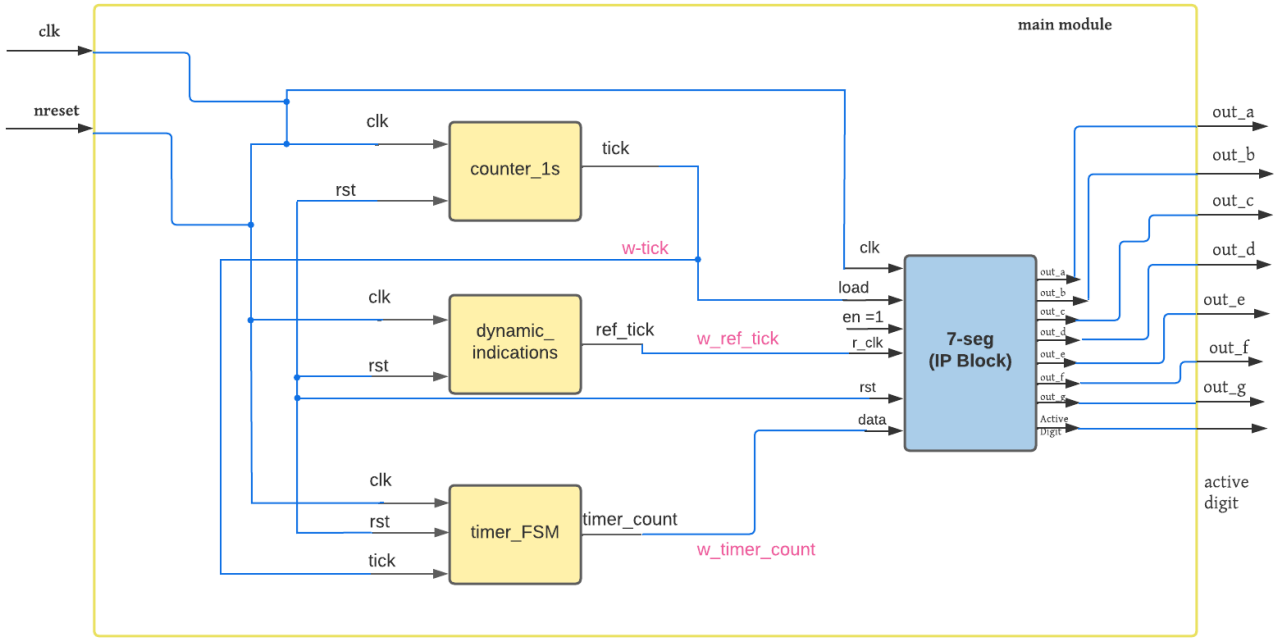

Figure 1: Sub-module connections

1. **Counter_1s**: This sub-module has two inputs- clk & rst and one output - tick. This tick goes HIGH when the counter counts and goes LOW when the count = 0. This tick triggers the loading of the data from timer_FSM submodule to the 7-Segment display module.
2. **Dynamic_Indications**: This sub-module has 2 inputs - clk and rst and one output - ref_tick. This submodule controls the dynamic indication of the 7-segment display by controlling the refresh clock of the 7segment module.
3. **Timer_FSM**: This sub-module has three inputs - clk, rst and tick(from the counter_1s) and one output - timer_count. The tick input controls the state of the timer_FSM and this in turn controls that data entering the 7-segment display module
4. **7-Segment**: The 7-segment display controller is used for displaying numbers and symbols on seven segment display. In this application note, we are displaying the 7-segment numbers on a PmodSSD connected externally on the Evaluation Board. The data on this 7-segment is loaded by connecting the 7segment IP block with the above-mentioned sub-modules. The seven output signals of this module connect to the PmodSSD to display the active numbers.
5. **Seven_seg_counter (top)**: The top module connects all the sub-modules together and runs as a single Verilog Code. It uses the onboard oscillator as the clock. It has two inputs - clk & rst and it has 8 output signals - out_a to out_g and active digit.

4. PmodSSD

For this application note, we are using the PmodSSD. PmodSSD is a peripheral module that can extend the capabilities of the boards. Users can toggle through GPIO signals which digit is currently on at a rate of 50Hz or greater to achieve persistence-of-vision to give the effect of both digits being lit up simultaneously.

Features:

- Two-digit high brightness seven-segment display
- Easily view a counter or timer
- Common Cathode configuration
- Small PCB size for flexible designs 1.0“× 1.7” (2.5 cm × 4.3 cm)
- Two 6-pin Pmod connectors with GPIO interfaces

The PmodSSD utilizes a common cathode configuration to display a variety of LED segment combinations. The ten segment combinations corresponding to digits 0 - 9 are generally the most useful, although other custom combinations can also be created (see Figure 2).

The PmodSSD communicates with the host board via the GPIO protocol. A logic level high signal on a particular anode will light up that respective segment on whichever digit is currently enabled. Users can select a particular digit by driving the Digit Selection pin (C) to a logic high or low voltage. Because only one digit can be lit at a particular time, users that want to use both digits to display a particular value will need to alternately light up the two digits at least every 20 milliseconds (50 Hz). This will correlate to each digit being lit up for 10 milliseconds each before the other segment needs to be “turned on”. Higher refresh rates can be achieved by alternating which digit is currently powered at shorter time intervals.

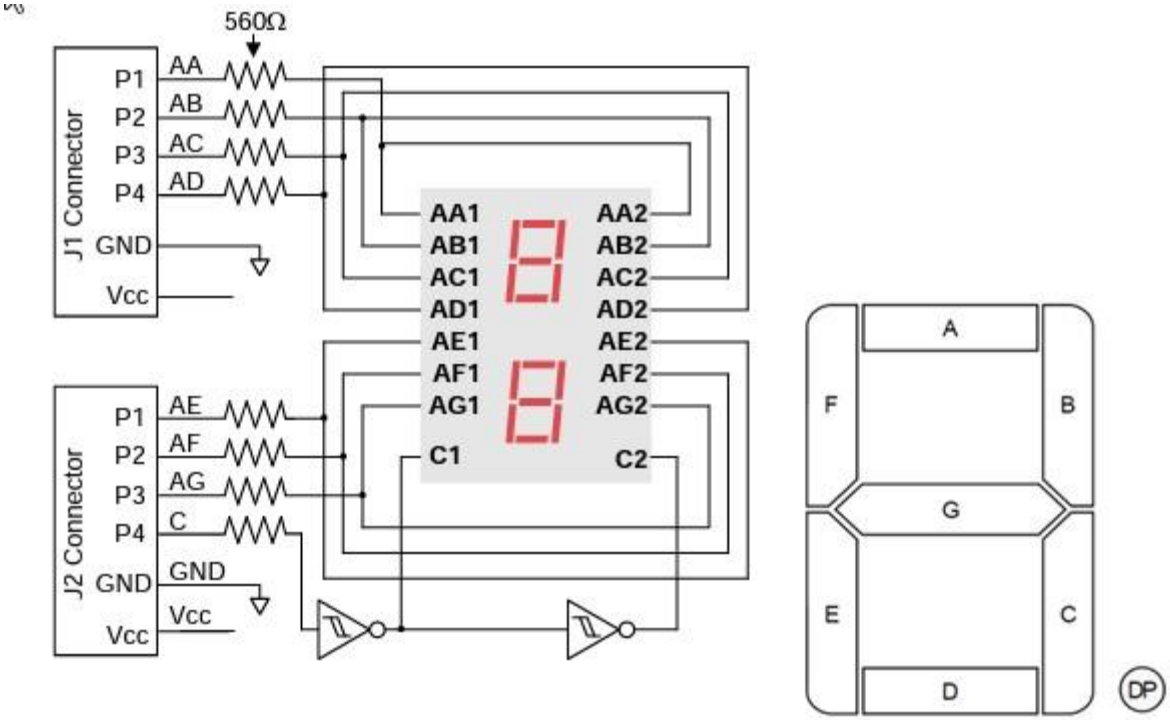

Figure 2: Seven-segment Display Connection Diagram

Table 1: Pinout Description Table

Header J1			Header J2		
Pin	**Signal**	**Description**	**Pin**	**Signal**	**Description**
1	AA	Segment A	1	AE	Segment E
2	AB	Segment B	2	AF	Segment F
3	AC	Segment C	3	AG	Segment G
4	AD	Segment D	4	C	Digit Selection Pin
5	GND	Power Supply Ground	5	GND	Power Supply Ground
6	VCC	Positive Power Supply	6	VCC	Positive Power Supply

5. Components

- ForgeFPGA SLG47910
- Latest Revision of ForgeFPGA Workshop software
- SLG47910 Evaluation Board
- PmodSSD

6. Verilog Code

Shown below is the (*top*) module called seven_seg_counter. It is available for download AN-FG-010 7-Segment Display using PMODSSD.ffpga

```
(*top*) module sevenseg (
  (* iopad_external_pin *) input nreset,
  (* iopad_external_pin, clkbuf_inhibit *) input clk,
  (* iopad_external_pin *) output osc_en,
  (* iopad_external_pin *) output out_a,
  (* iopad_external_pin *) output out_b,
  (* iopad_external_pin *) output out_c,
  (* iopad_external_pin *) output out_d,
  (* iopad_external_pin *) output out_e,
  (* iopad_external_pin *) output out_f,
  (* iopad_external_pin *) output out_g,
  (* iopad_external_pin *) output active_digit,
  (* iopad_external_pin *) output out_a_oe,
  (* iopad_external_pin *) output out_b_oe,
  (* iopad_external_pin *) output out_c_oe,
  (* iopad_external_pin *) output out_d_oe,
  (* iopad_external_pin *) output out_e_oe,
  (* iopad_external_pin *) output out_f_oe,
  (* iopad_external_pin *) output out_g_oe,
  (* iopad_external_pin *) output active_digit_oe
  );

  wire [7:0] w_timer_count;
  wire w_ref_tick;
  wire w_tick;
```

```
  wire rst;

  assign rst = !nreset;
  assign osc_en = 1'b1;

//oe
      assign out_a_oe = 1; assign out_b_oe = 1;
      assign out_c_oe = 1; assign out_d_oe = 1;
      assign out_e_oe = 1; assign out_f_oe = 1;
      assign out_g_oe = 1; assign active_digit_oe = 1;

  counter_1s counter_1s_wrapp(
   .clk (clk),
   .rst (rst),
   .tick (w_tick)
  );

  dynamic_indication dyn_ind_wrapp(
   .rst (rst),
   .clk (clk),
   .ref_tick (w_ref_tick)
  );

  timer_FSM timer_FSM_wrapp (
   .clk (clk),
   .rst (rst),
   .tick (w_tick),
   .timer_count (w_timer_count)
  );

 seven_segment_disp #(
  .SEL_CA (1)
 )seven_segment_disp_wrapp (
  .clk (clk),
  .load (w_tick),
  .en (1'b1),
  .rst(rst),
  .refresh_clock(w_ref_tick),
  .data ({2'b00,w_timer_count}),
  .active_digit(active_digit),
  .out_a(out_a),
  .out_b(out_b),
  .out_c(out_c),
  .out_d(out_d),
  .out_e(out_e),
  .out_f(out_f),
  .out_g(out_g)
  );


endmodule
```

7. Floorplan: CLB Utilization

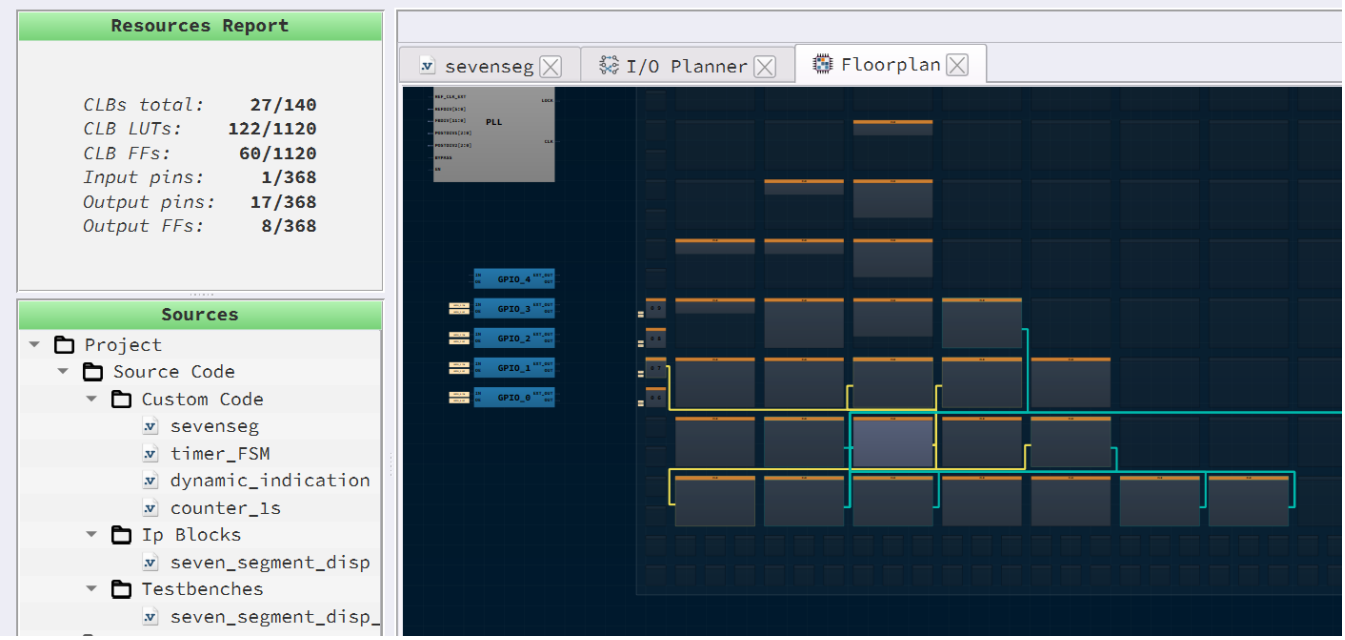

Figure 3: Floorplan & CLB Utilization

The resources that are utilized when designing the 7-segment display can be seen under the Floorplan and the Resources Tab in the Toolbar on the top of the window. The Floorplan highlights the CLB's that are been utilized in this design and when the user clicks the any CLB box, the software shows the wires(blue/yellow) connecting multiple CLB blocks.

8. File Structure & Resources

With the latest updated software revision v6.41, user can see the structure of the multiple Verilog files in the project and understand the resources that are used.

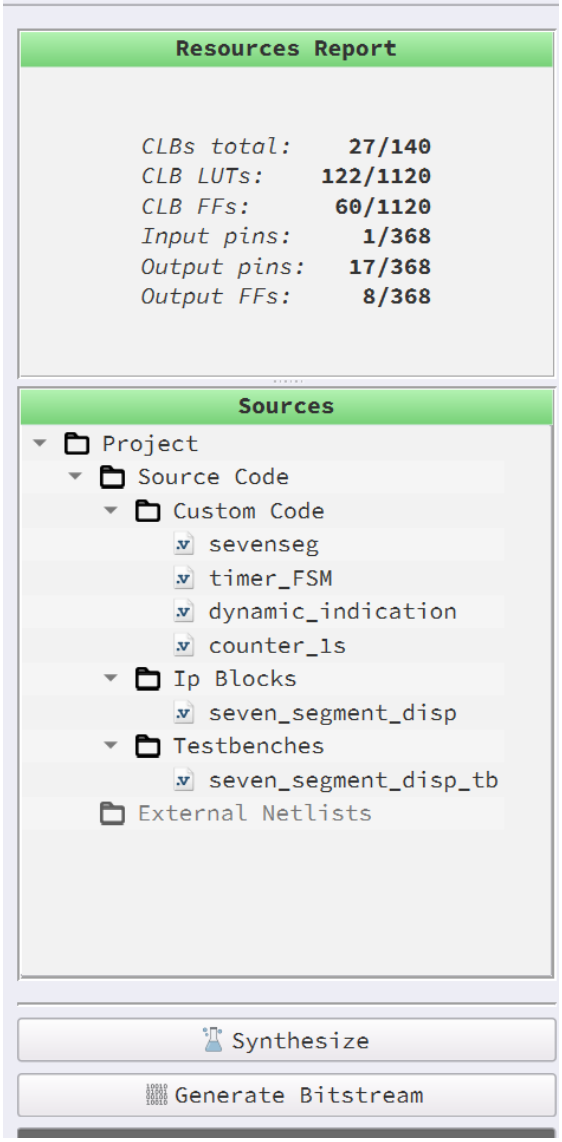

Figure 4: File Structure and Resources Used

9. Design Steps

1. Launch the latest version of the Go Configure Software Hub. Select the SLG47910V device and the ForgeFPGA Workshop software will load.
2. Download the design example 7-Segment display using PmodSSD.ffpga. If you are not familiar with the ForgeFPGA Workshop software, review the Four-Bit Counter application notes that covers the basic design steps.
3. Open the 7-Segment display using PmodSSD.ffpga file after downloading
4. Open the FPGA editor and review the Verilog code. There is a main code with the module name seven_seg_counter, which is the top module defining the whole design. There are 3 sub-modules designed in this example along with an IP Block. All these sub-modules are integrated together to form the top module as function as intended (see Figure 4 for structure).
5. Open the IO planner tab on the FPGA editor and review the pin assignment (Figure 5).

POSITION	FUNCTION	PORT
IOB tile[0, 0] coord[0, 9] Output0	[PIN 16] GPIO3_OUT	active_digit
IOB tile[0, 0] coord[0, 9] Output1	[PIN 16] GPIO3_OE	active_digit_oe
CLK tile[0, 0] clk_side=W Input0	OSC_CLK	clk
IOB tile[0, 0] coord[31, 11] Input0	FPGA_CORE_READY	nreset
IOB tile[0, 0] coord[0, 25] Output0	OSC_EN	osc_en
IOB tile[0, 0] coord[31, 27] Output0	[PIN 23] GPIO8_OUT	out_a
IOB tile[0, 0] coord[31, 27] Output1	[PIN 23] GPIO8_OE	out_a_oe
IOB tile[0, 0] coord[31, 26] Output0	[PIN 24] GPIO9_OUT	out_b
IOB tile[0, 0] coord[31, 26] Output1	[PIN 24] GPIO9_OE	out_b_oe
IOB tile[0, 0] coord[31, 25] Output0	[PIN 1] GPIO10_OUT	out_c
IOB tile[0, 0] coord[31, 25] Output1	[PIN 1] GPIO10_OE	out_c_oe
IOB tile[0, 0] coord[31, 24] Output0	[PIN 2] GPIO11_OUT	out_d
IOB tile[0, 0] coord[31, 24] Output1	[PIN 2] GPIO11_OE	out_d_oe
IOB tile[0, 0] coord[0, 6] Output0	[PIN 13] GPIO0_OUT	out_e
IOB tile[0, 0] coord[0, 6] Output1	[PIN 13] GPIO0_OE	out_e_oe
IOB tile[0, 0] coord[0, 7] Output0	[PIN 14] GPIO1_OUT	out_f

Figure 5: IO Planner

6. Next select the Synthesize button on the lower left side of the FPGA editor. Select the Generate Bitstream button on the lower left side of the FPGA editor. Check the Logger and Issues tabs to make sure that the bit stream was generated correctly.
7. Now click on the Floorplan tab and see the CLB utilization (Figure 3). Press the Ctrl and the mouse wheel to zoom-in. Confirm that the IOs selected in the IO Planner are shown in the floorplan.
8. Connect the Evaluation Board to your system and now connect PmodSSD to the GPIOs on the board. Once the connection has been established, click on the Debug button on the ForgeFPGA Workshop studio, select platform as ForgeFPGA Evaluation Board and select Emulation. Make sure the VDD = 1.2V and VDDIO = 2.3V (Figure 6)

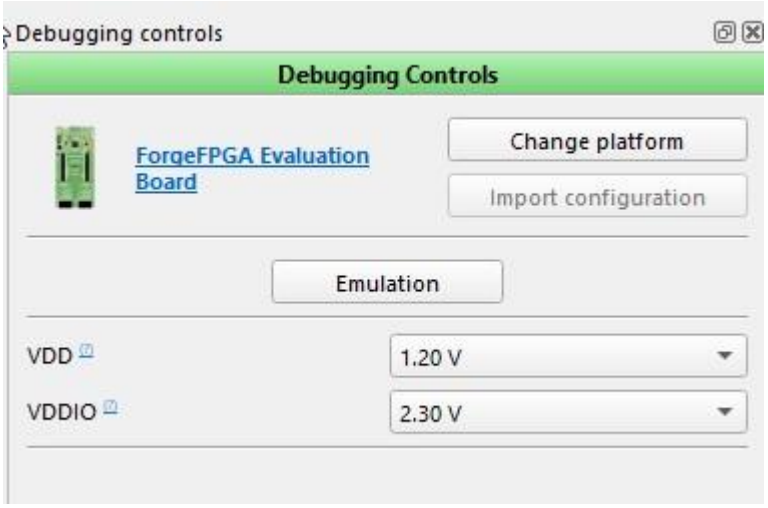

Figure 6: Evaluation Board Settings

9. Once the user clicks the emulation button on the software, the bitstream gets loaded on the SLG47910 part. You can now observe the countdown being reflected in the 2-digits of the PmodSSD 7-segment display (see Figure 7).

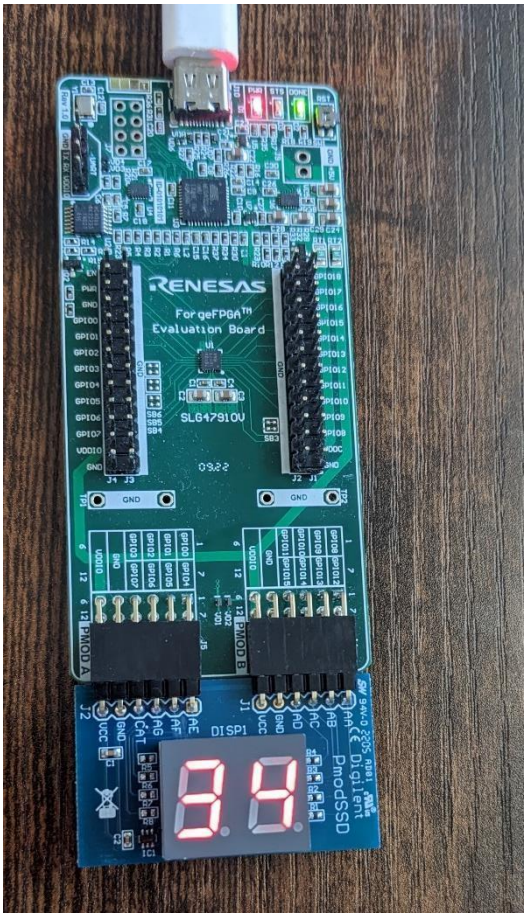

Figure 7: PmodSSD Countdown

10. Conclusion

This application note shows how to connect a PMOD externally to the PMOD slots of the ForgeFPGA Evaluation board and the Verilog code configuration for it. This application note focuses on using a PmodSSD and configuring it to display the countdown from 0-99 in 100 seconds. This testcase is available for download AN-FG-010 7-Segment Display using PMODSSD.ffpga.

11. Revision History

Revision	Date	Description
1.00	Jan 03,2023	Initial release.
2.00	Feb 20,2024	Updated according to BB revision
2.01	Aug 06,2024	Updated as per ForgeFPGA Workshop v6.43

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

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-us/.

Trademarks

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