AN-1092

I2C Controlled State Machine

Author: Roger Liang, Luke Thomas

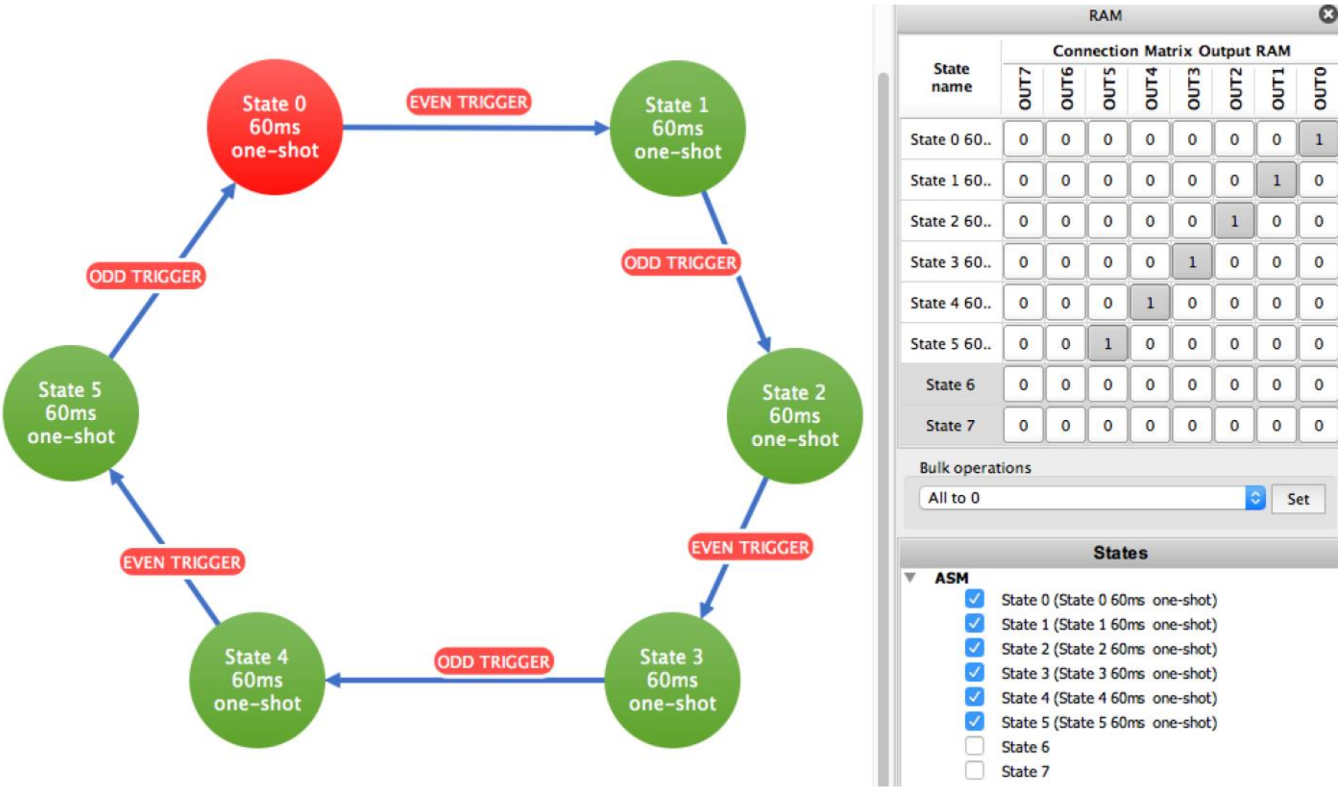
Date: January 16, 2016

Introduction

The GreenPAK5 programmable mixed-signal ASICs family introduces Dialog's first asynchronous state machine (ASM). One of the ASM's biggest benefits is decreasing the complexity of designs. The SLG46531, the first product in the GreenPAK5 family, also comes with a slave I2C macro-cell. The combination of the ASM and the I2C macro-cell gives the system engineer nearly unlimited flexibility in constructing their design. For example, counter timings, comparator thresholds, and look-up-tables can be changed during runtime. This application note shows how to reset and restart the state machine via I2C.

Building a simple state machine

We first build a simple ASM: a repeating loop of one-shot pulses of equal widths on 6 separate outputs. There are a number of ways to approach this design; the simplest way is to build a looping ASM. We start by clicking the ASM Editor button in the designer software. Figure 1 shows the ASM editor window with a straightforward unidirectional loop through 6 states. We use the ASM's 8-bit configurable output to sequence the loop of one-shot pulses. Shown in figure 1's RAM window, the ASM outputs are configured in a cascaded series of "1's".



Connection Matrix Output RAM

State name	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
State 0 60..	0	0	0	0	0	0	0	1
State 1 60..	0	0	0	0	0	0	1	0
State 2 60..	0	0	0	0	0	1	0	0
State 3 60..	0	0	0	0	1	0	0	0
State 4 60..	0	0	0	1	0	0	0	0
State 5 60..	0	0	1	0	0	0	0	0
State 6	0	0	0	0	0	0	0	0
State 7	0	0	0	0	0	0	0	0

Bulk operations: All to 0 | Set

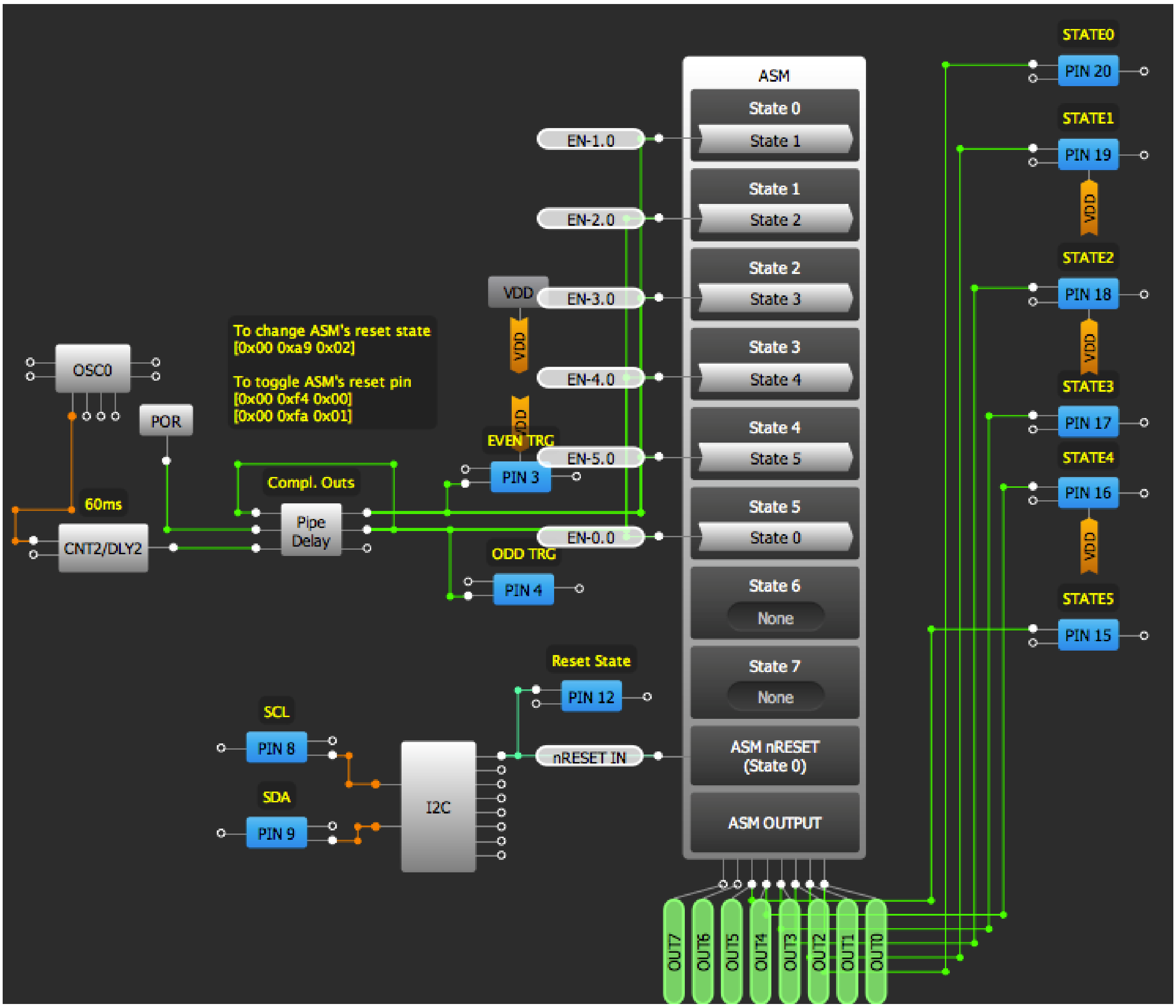
States

- ASM
  - ☑ State 0 (State 0 60ms one-shot)
  - ☑ State 1 (State 1 60ms one-shot)
  - ☑ State 2 (State 2 60ms one-shot)
  - ☑ State 3 (State 3 60ms one-shot)
  - ☑ State 4 (State 4 60ms one-shot)
  - ☑ State 5 (State 5 60ms one-shot)
  - ☐ State 6
  - ☐ State 7

Figure 1. ASM Editor

After we build the state diagram in ASM Editor, we go back to the main designer software window to add the state transition triggers. Figure 2 shows the circuit schematic. Pay special attention to how state transitions are represented on the ASM block.

Each state is illustrated as a gray box; the next states are illustrated as a smaller silver arrow-shaped box inside the gray box. Take State 0 for example: its box contains a State 1 arrow. This means that in order to transition from State 0 to State 1, the input of the State 1 arrow needs to be high while the ASM is in state 0.



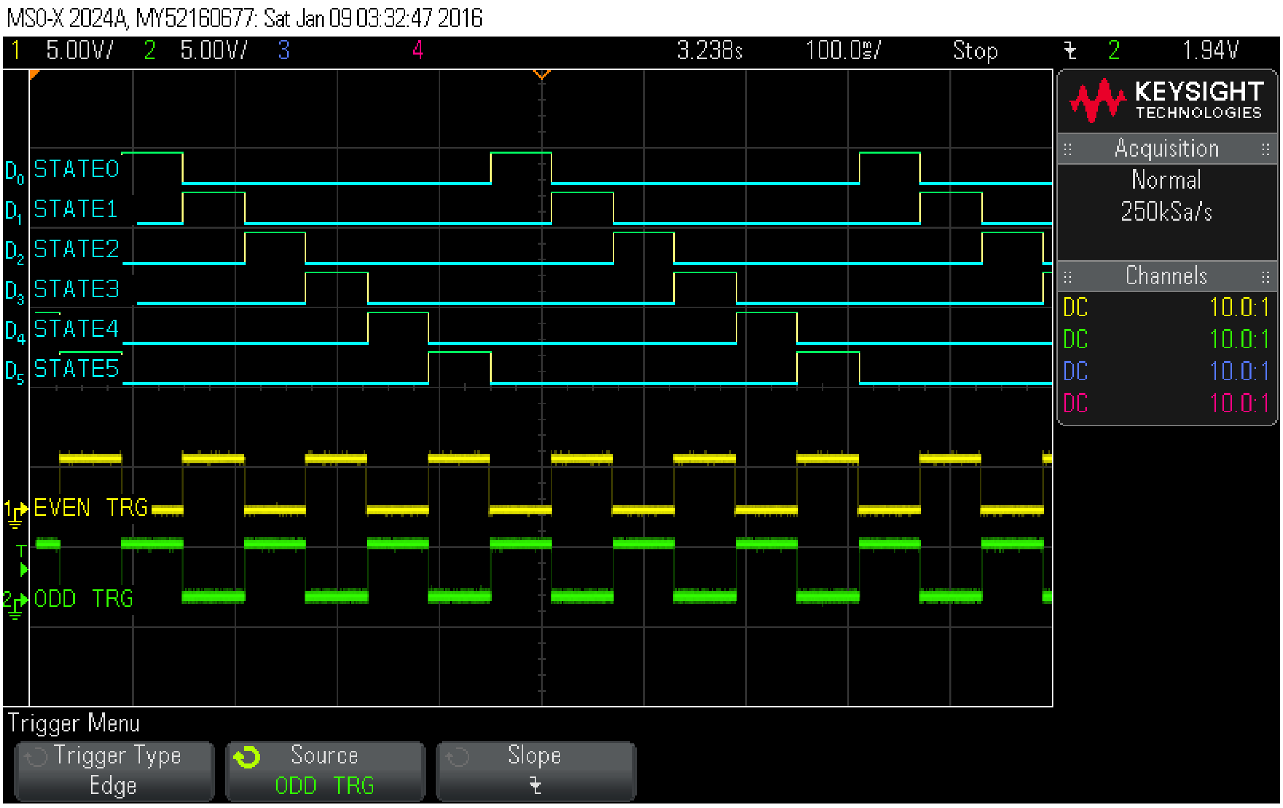
To change ASM's reset state
[0x00 0xa9 0x02]

To toggle ASM's reset pin
[0x00 0xf4 0x00]
[0x00 0xfa 0x01]

Figure 2. I2C controlled ASM schematic

CNT2 is used to generate a 60ms timer that is used as the one-shot pulse width for all the states. Since the ASM's inputs are level triggered and not edge triggered, we cannot simply use CNT2's output as the trigger for all state transitions. To do so would cause almost instantaneous transitions from state to state instead of waiting 60ms between transitions. To address this, the Pipe Delay macro-cell is used to generate 2 complementary outputs with 50% duty cycle and a 120ms period. While one signal is used to transition from even to odd numbered states, the other is used to transition from odd to even numbered states.

Figure 3 shows the design's state transitions. When the ASM is in state 0, its OUT0 will be high for 60ms, after which the EVEN TRG signal will transition the ASM to state 1. Notice if the same EVEN TRG signal is used to transition the ASM from state 1 to 2, it would happen almost immediately because the ASM's inputs are activated by active high signals, not by rising edges. By using 2 complementary signals called EVEN TRG and ODD TRG, state transitions will consistently happen every 60ms and the ASM will loop indefinitely.



MSO-X 2024A, MY52160677: Sat Jan 09 03:32:47 2016

Acquisition: Normal, 250kSa/s

Trigger Menu — Trigger Type: Edge; Source: ODD TRG; Slope: ⭍

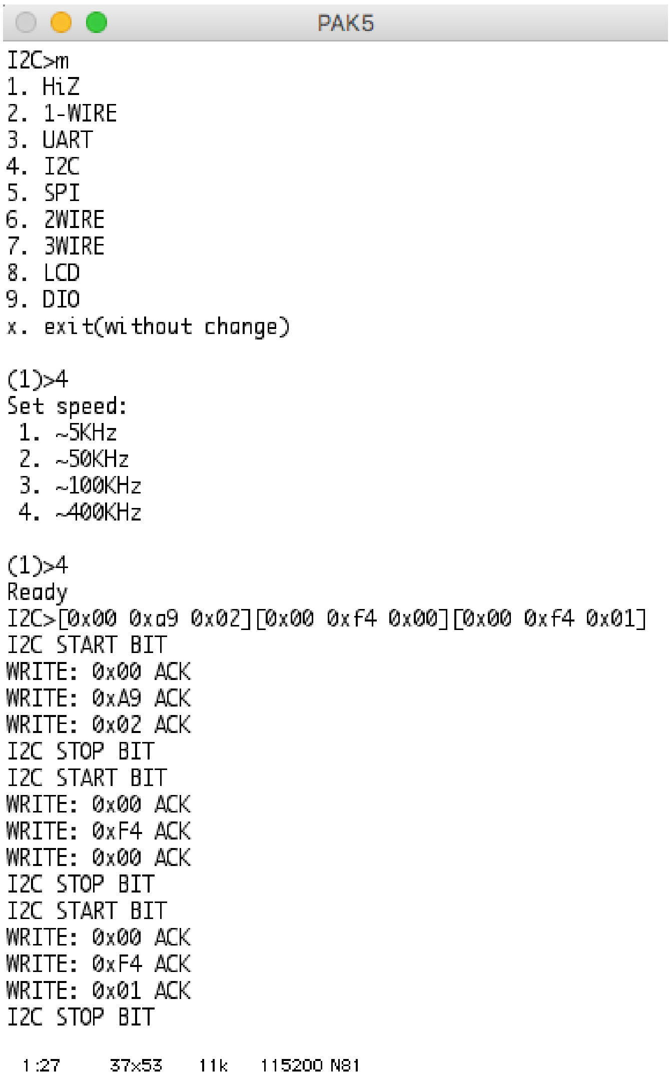
Figure 3. State transition waveform

Using I2C commands to alter state transitions

Since the SLG46531 is an I2C slave device, its state machine can be changed via I2C in real time. Any master I2C device in the system can use the 3 commands below to reset and change states:

1. Change the ASM's default state [0x00 0xa9 0x02]
2. Assert a low signal on ASM's nRESET pin [0x00 0xf4 0x00]
3. Assert a high signal on ASM's nRESET pin [0x00 0xf4 0x01]

For example if we want to send the ASM into state 2, we first set the ASM's default state to "state 2". Then we reset and pause the ASM to default state by asserting a low signal on its nRESET pin. This will send the ASM to state 2 regardless of its current state. Finally we release the low signal on nRESET so the ASM restarts to run from state 2. Figure 4 shows the 3 commands in a terminal window. Bus Pirate is used as the I2C master to send these commands to the SLG46531. For details on how to communicate to SLG46531 via Bus Pirate, please refer to AN-1091. The first write command, [0x00 0xa9 0x02], sets the default state to 2. Please refer to SLG46531's datasheet to verify that the 3 least significant bits of register address 0xa9 are used to set the default state. The second and third commands, [0x00 0xf4 0x00][0x00 0xf4 0x01], reset the ASM to its default state and let the ASM resume operation from that default state. Register address 0xf4's LSB is the ASM nRESET pin.



```
I2C>m
1. HiZ
2. 1-WIRE
3. UART
4. I2C
5. SPI
6. 2WIRE
7. 3WIRE
8. LCD
9. DIO
x. exit(without change)

(1)>4
Set speed:
 1. ~5KHz
 2. ~50KHz
 3. ~100KHz
 4. ~400KHz

(1)>4
Ready
I2C>[0x00 0xa9 0x02][0x00 0xf4 0x00][0x00 0xf4 0x01]
I2C START BIT
WRITE: 0x00 ACK
WRITE: 0xA9 ACK
WRITE: 0x02 ACK
I2C STOP BIT
I2C START BIT
WRITE: 0x00 ACK
WRITE: 0xF4 ACK
WRITE: 0x00 ACK
I2C STOP BIT
I2C START BIT
WRITE: 0x00 ACK
WRITE: 0xF4 ACK
WRITE: 0x01 ACK
I2C STOP BIT

 1:27    37x53    11k    115200 N81
```

Figure 4. I2C commands in terminal

Figure 5 shows the timing waveform where I2C is used to reset the ASM to state 2. The ASM's nRESET pin is shown in yellow and the I2C SCL and SDA are shown in green and blue respectively. Figure 6 shows the same timing waveform but zoomed into the three I2C write commands.

Notice the first I2C command, [0x00 0xa9 0x02], has no bearing on ASM current state because it only sets the default state to “state 2”.

After the second command, [0x00 0xf4 0x00], is sent, the ASM jumps from its current state to State2. The third command, [0x00 0xf4 0x01], releases the ASM from its default state so it can continue to transition to the next state.

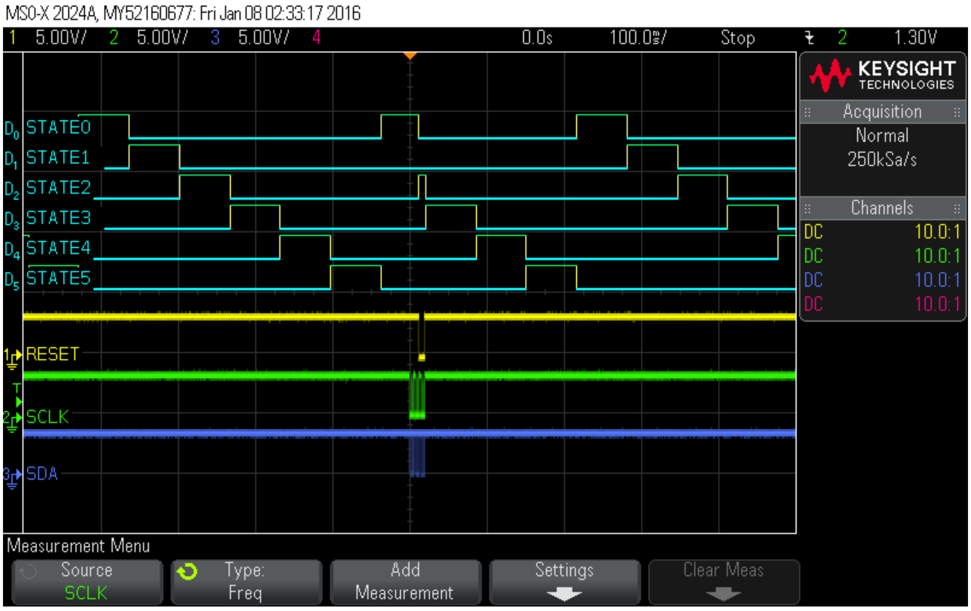


Figure 5. Resetting ASM timing waveform

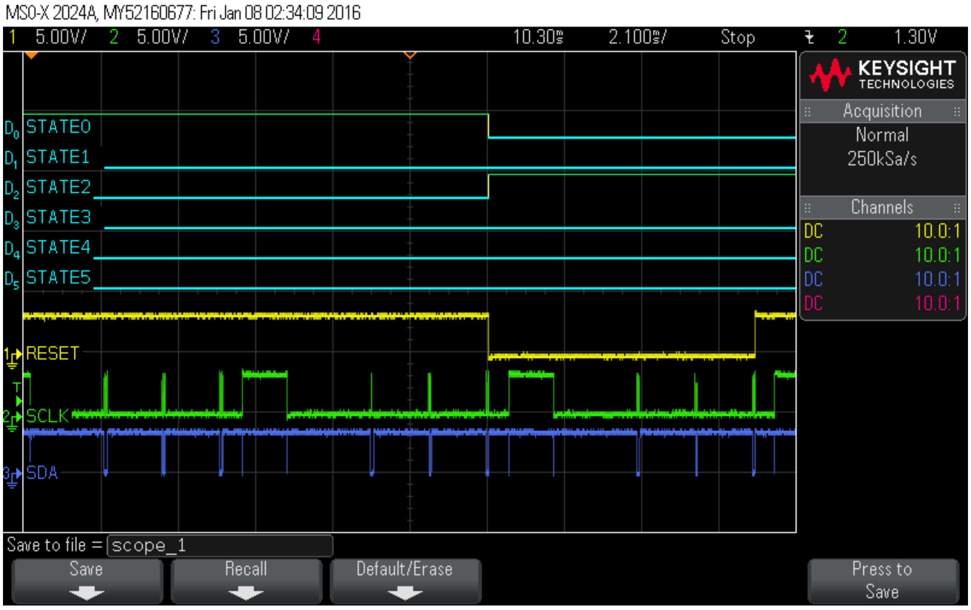


Figure 6. Resetting ASM timing waveform zoomed

Conclusion

This application note shows a simple ASM structure with a set of simple I2C commands. Based on system-level requirements, a more sophisticated state machine with limitless I2C configurability can be constructed.

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)

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-us/.