Application Note

Quadrature Encoder Counter with SPI Bus Interface

AN-CM-277

Abstract

This application note describes an SLG46140V design that implements a 16-bit up/down counter with quadrature encoder inputs. The counter will keep track of the current position of a rotary encoder, and a host microcontroller or PC can read out the position using standard Serial Peripheral Interface (SPI) bus. The GreenPAK device relieves the host of realtime input requirements and allows for easy connection of multiple encoders.

Contents

Figures

Tables

1 Terms and Definitions

IC	Integrated circuit
FSM	Finite state machine
MISO	Master input slave output
Quadrature	Pair of signals that have 90-degree phase difference
SPI	Serial peripheral interface

2 References

For related documents and software, please visit:

GreenPAK™ Programmable Mixed-Signal Products | Renesas

Download our free GreenPAK Designer software [1] to open the .gp files [2] and view the proposed circuit design. Use the GreenPAK development tools [3] to freeze the design into your own customized IC in a matter of minutes. Renesas Electronics provides a complete library of Application Notes [4] featuring design examples as well as explanations of features and blocks within the Renesas IC.

[1] GreenPAK Designer Software, Software Download and User Guide, Renesas Electronics.

[2] AN-CM-277 Quadrature Encoder Counter with SPI Bus Interface.gp, GreenPAK Design File, Renesas Electronics.

[3] GreenPAK Development Tools, GreenPAK Development Tools Webpage, Renesas Electronics.

[4] GreenPAK Application Notes, GreenPAK Application Notes Webpage, Renesas Electronics.

3   Introduction

Rotary encoders are widely used to sense the orientation of shafts and machine parts, and as user interface input devices. Most rotary encoders output a pair of quadrature signals, which encode the shaft movement as a series of pulses.

To keep track of the encoder position, every pulse must be counted. Often, interrupt pins are used for this purpose, with the counting implemented in software. Many microcontrollers include specialized peripherals to read quadrature signals. However, interrupt based counting limits the maximum pulse rate and takes up CPU time, while the specialized peripherals limit the number of encoders that can be connected to a single microcontroller. Furthermore, on non-realtime platforms such as PCs neither option is available.

This application note demonstrates how a low-cost SLG46140V GreenPAK device can implement a quadrature encoder interface with a 16 bit up/down counter and SPI bus interface. The current position of the encoder can be read out by the host application at the desired interval, while the auxiliary device keeps track of each input pulse. Furthermore, the SPI bus allows chaining multiple devices to interface a flexible number of encoders to a single processor.

4   High-Level Design

This application note design has three input signals from the encoder: A, B and Z. Signals A and B are the quadrature signals which switch state alternately in 00, 10, 11, 01 gray code sequence. Signal Z is an optional encoder zero index input, which resets the counter to the zero position.
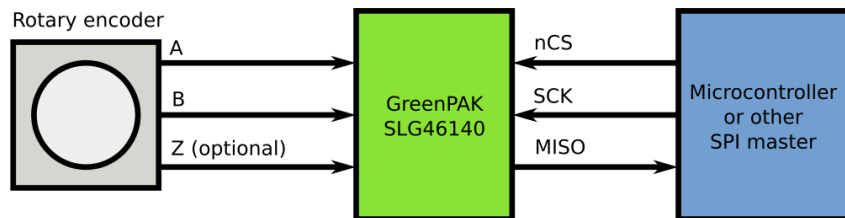
Figure 1: System Connections with One Encoder Device

The described SPI interface consists of the nCS chip select signal, a SCK serial clock, and the MISO (master-in slave-out) serial output. Chip select is active low, and the bus can be used by other SPI devices when nCS is high. The SPI transfer format used in this application note is CPOL=0, CPHA=1, which means that SCK is in low state when idle and data is sampled on the falling edge.

Figure 2: System Connections with Multiple Encoders

Quadrature Encoder Counter with SPI Bus Interface

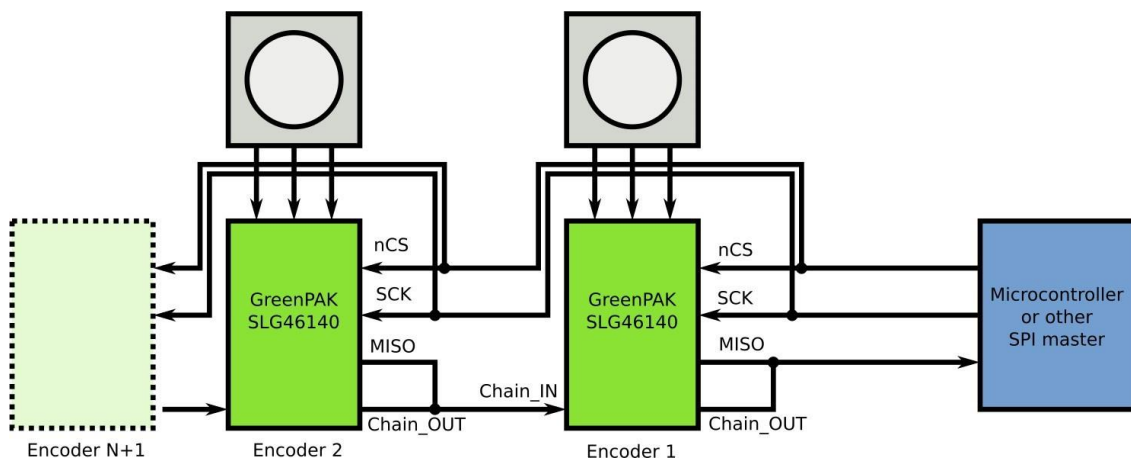Multiple devices can be chained on the same bus by using Chain_IN and Chain_OUT signals, which internally delay the data for 16 SCK clock cycles. Thus, the first 16 bits received by the host will be from the first encoder, the next 16 bits from the second encoder and so on. All encoders will be sampled at the same time, on the first SCK clock edge.

The implementation inside the GreenPAK device consists of three parts:

1. Quadrature Encoder Input Block: Converts quadrature signals to KEEP / UP signal pair for the counter, and synchronizes the signals to internal clock.
2. 16-bit Up/Down Counter: Two chained 8-bit counters count the encoder pulses and provide parallel data to SPI block.
3. SPI Bus Interface: SPI interface latches the parallel data and outputs it serially when nCS is active.

The quadrature input block and the counter run from the internal ring oscillator clock, while the SPI interface is clocked externally by the SPI master. The following sections contain a detailed description of each component.

4.1   Quadrature Encoder Input Block

Signals from the encoder initially go to a pair of delay blocks. These work as both glitch filters and synchronizers, ensuring that their output signal only changes at the clock rising edge. The delay value is set to minimum, giving a 2-clock cycle delay. Optionally, for noisy signals, the delay can be increased to filter out any glitch pulses from the inputs, but this filtering will also limit the maximum pulse rate.

Output from the delay blocks is passed to a pair of D-flipflops, which retain the signal state from the previous clock cycle. The current state and the previous state are then compared with a 4-input XNOR gate. If either of the signals has changed, the XNOR gate will output a 0 value for KEEP signal, causing the counter block to count.

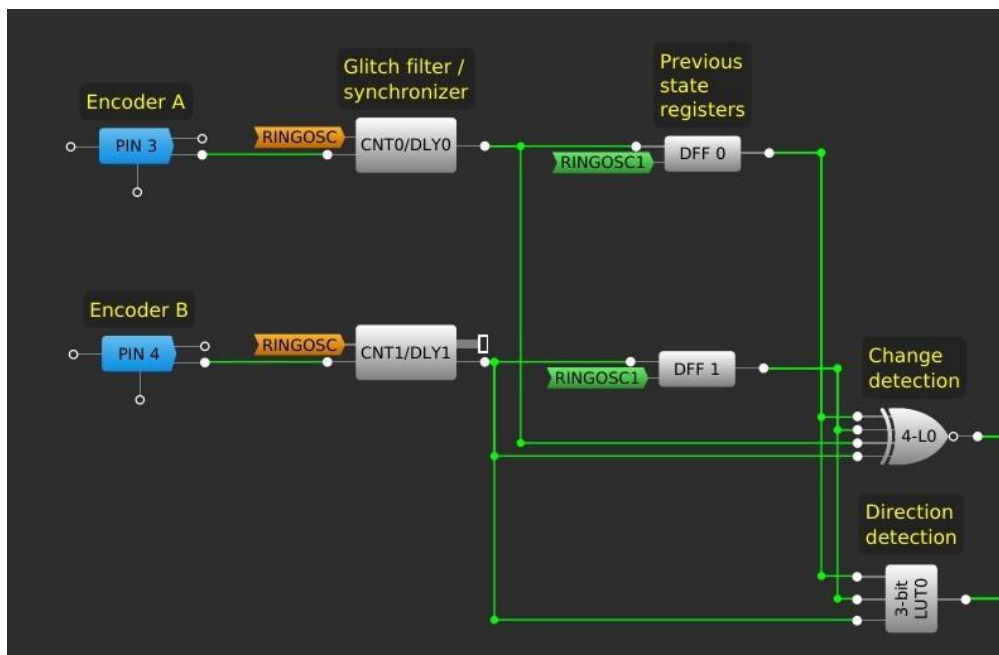**Figure 3: Quadrature Signal Input Block**

Encoder direction is detected by comparing the current states At and Bt with the previous state Bt-1, as shown in Table 1. The detection logic has been programmed to the 3-bit LUT0 component, which outputs an UP signal to the counters.

Quadrature Encoder Counter with SPI Bus Interface

Table 1: Direction Detection from Signal States

Increasing Angle (UP = 1)				Decreasing Angle (UP = 0)		
At	Bt	Bt-1		At	Bt	Bt-1
0	0	0		0	0	1
0	1	0		0	1	1
1	1	1		1	1	0
1	0	1		1	0	0

Some encoders have an additional zero position sensor, which can be used to find the absolute position of the system after a reboot. A high level on this Z signal will reset the counter to zero. If this functionality is not needed, the signal can be left unconnected as the input pin has been configured with a pulldown resistor.
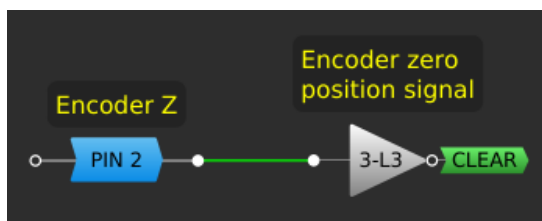
4.2 16-bit Up/Down Counter

Figure 4: Logic for Z Input Signal

The SLG46140V device contains two counter / finite state machine blocks with parallel data connection to the SPI block. Each of these can output an 8-bit count, and by chaining the two counters a 16-bit counter can be made. The counter wraps around from 0x0000 to 0xFFFF in either direction.

By default, the CNT3 block counts from configurable start value either downwards to 0 or upwards to 255, and then restarts at the configured value. Wrap around for downward counting is available by setting the start value for 255, but to get wraparound for upward counting simultaneously, extra logic is needed.
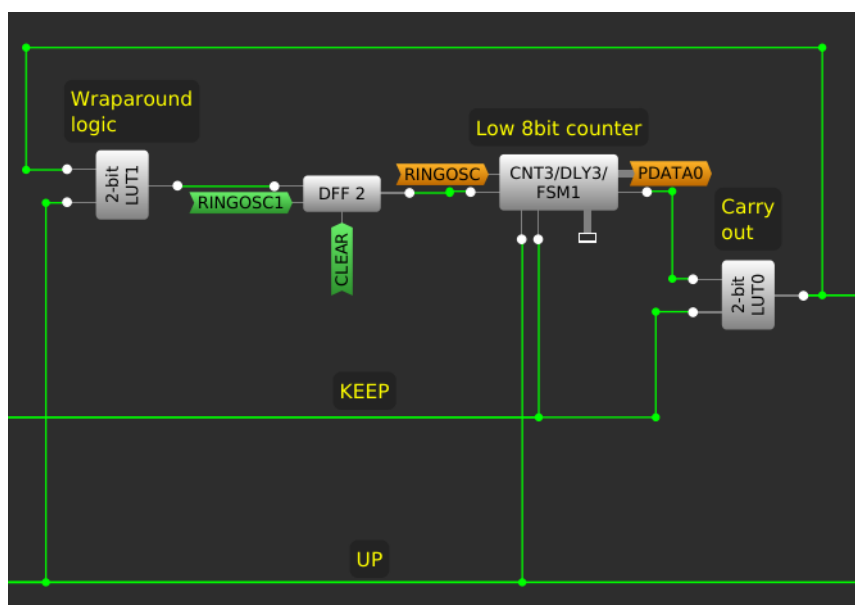

Figure 5: Carry Out and Wraparound Logic for Lower 8-bit Counter

Quadrature Encoder Counter with SPI Bus Interface

The counter's OUT signal activates when the counter is at its extreme value. When KEEP=0 and OUT=1, LUT0 outputs a carry pulse to the next counter to increment its count. Simultaneously LUT1 activates the reset signal if UP=1. DFF2 samples the reset signal so that the counter resets synchronously on the clock rising edge.

CLEAR signal from the encoder Z input is connected to asynchronous set input of DFF2, which causes an immediate reset of the counter. The counter resumes counting on the next rising clock edge.
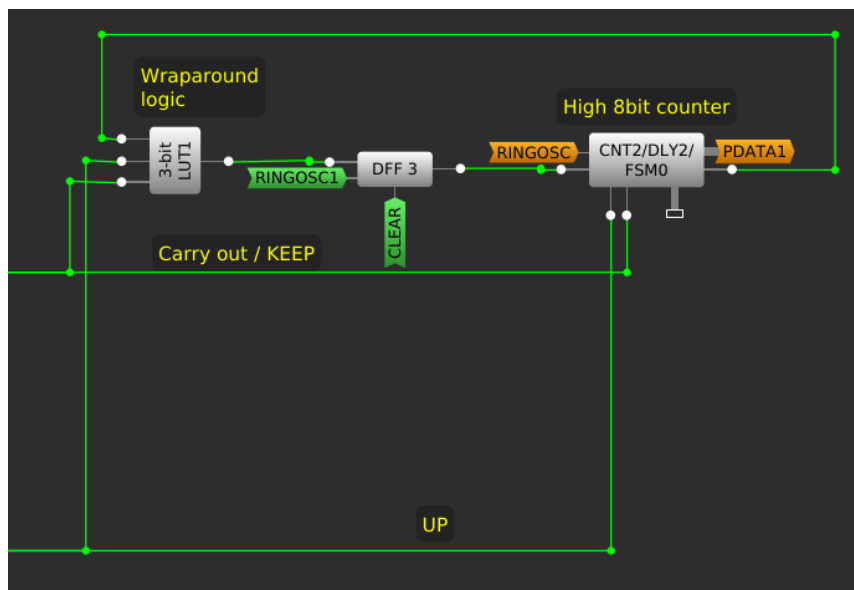

Figure 6: Wraparound Logic for High 8-bit Counter

The second counter stage is connected similarly, because a separate carry out signal is not needed from this stage; the three signals are all connected to a single LUT component. It resets the counter to zero when UP=1, KEEP=0 and OUT=1.

4.3   SPI Bus Interface

The parallel to serial converter interface is available as a ready-made functional block in the SLG46140 and the SLG46620. The block latches the parallel data from the two counters on the first SCK edge after nCS goes low, and then shifts out the data bit by bit.

The externally generated SCK signal is not synchronous with the internal clock used by the counters, so the two blocks need to be synchronized in some way to reliably transfer data between them. The SPI block offers a built-in selection for gating the FSM clock for two clock cycles when data transfer occurs. In this application gating the FSM clock is undesirable, as it could cause pulses to be lost while the clock was disabled. Instead, DFF5 is used to synchronize the external SCK signal to internal ring oscillator clock. This ensures that the SCK clock edge as seen by SPI block occurs at ring oscillator clock edge, when the FSM data is stable.

In this application note, the basic SPI interface is extended by data chaining logic. This allows connecting multiple encoders easily on the same SPI bus, and sampling the counter value of them all simultaneously.

The chaining function is based on Pipe Delay functional block. This block has a chain of 16 flip-flops, which will shift in and shift out one bit on each rising edge of the clock signal. The MISO signal from previous device in the chain is connected to the input, and the output goes to the next device.

Because the MISO signal is directly connected to pin 12 in hardware, Chain_OUT will have to be externally connected in parallel with it. The output enable logic ensures that only one of the pins is active at a time.

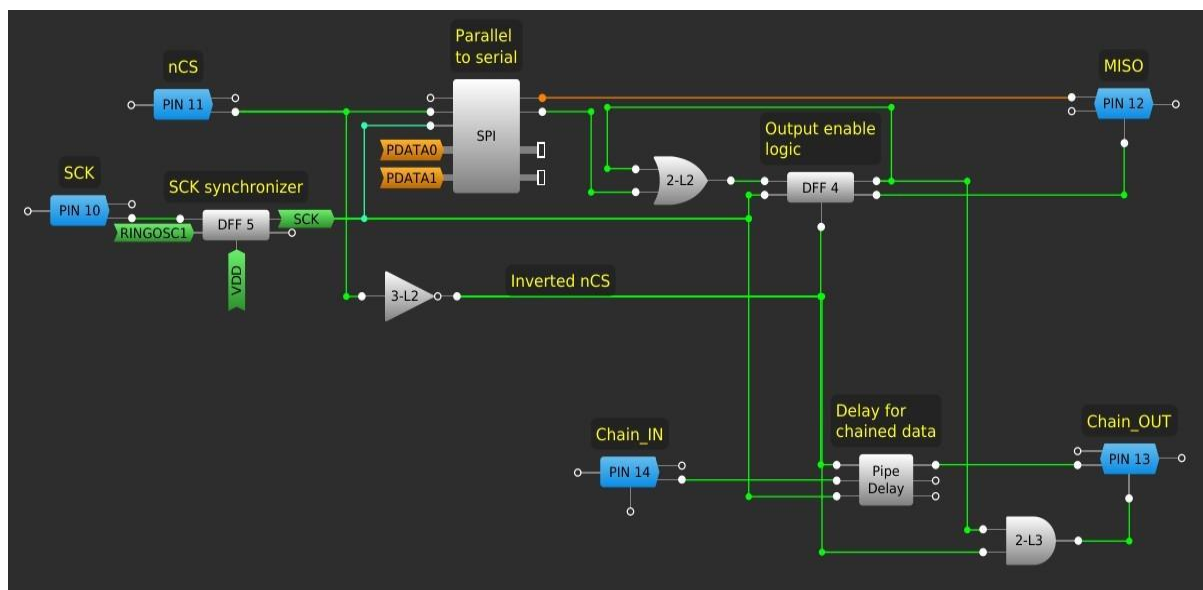## Quadrature Encoder Counter with SPI Bus Interface

Figure 7: SPI Interface and Data Chain Logic

For the first 16 clock cycles after nCS goes low, each device keeps MISO enabled and clock out its counter value. Simultaneously, the Pipe Delay block stores the signal received from the N-1th device in the chain.

After 16 clock cycles, the SPI INTR output goes high to indicate end of transmission. LUT2 OR-gate then causes DFF4 flipflop to go high, disabling PIN12 and enabling PIN13. The Pipe Delay block shifts out the N-1th value, while simultaneously shifting in N-2th value. This continues until all values have been read and the host raises an nCS signal, which resets DFF4 back to its initial state.
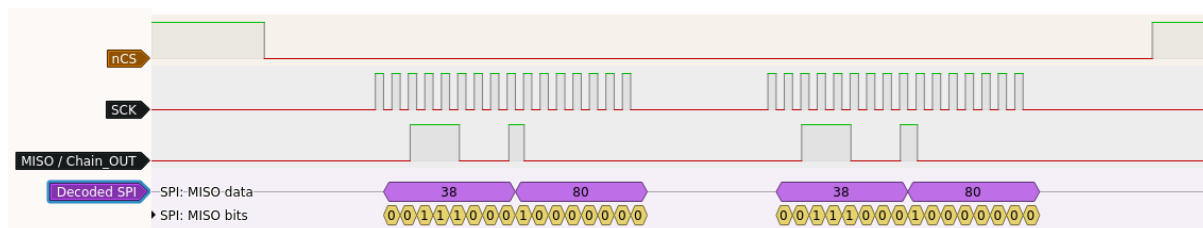

Figure 8: SPI Signal Capture with Two Devices

Figure 8 shows an example of SPI communication where first device sends its own 16 bit data value, and then forwards the 16 bit value from the next device in the chain.

4.4   Clock Rate Limitations

The delay and counter blocks are clocked from the internal ring oscillator, which runs at approximately 27 MHz frequency and has selectable divider of 1, 4, 8, or 16. The maximum clock rate depends on the delay of the logic elements and varies with the supply voltage.

The longest logic chain in the design goes from input DLY0/1 blocks through 4-bit LUT0 change detector, 2-bit LUT0 carry out logic and 3-bit LUT1 wraparound logic to the FSM blocks. The delay values for these blocks are defined in the SLG46140 datasheet and are summarized in Table 2. Due to routing delays and part variance, the theoretically possible 6.8 MHz frequency did not work with lowest end of the GreenPAK's input voltage range (V_{DD} = 1.8 V), and 3.4 MHz clock is used instead.

Quadrature Encoder Counter with SPI Bus Interface

Table 2: Block Delays and Maximum Operating Frequency

Block	Delay when V_{DD} = 1.8 V	Delay when V_{DD} = 3.3 V	Delay when V_{DD} = 5.0 V
CNT/DLY	53 ns	21 ns	15 ns
LUT 4-bit	23 ns	10 ns	7 ns
LUT 2-bit	18 ns	7 ns	5 ns
LUT 3-bit	19 ns	7 ns	5 ns
Total	113 ns	45 ns	32 ns
Theoretical max frequency	8.8 MHz	22 MHz	31 MHz
Clock divider	1:8 (3.4 MHz)	1:4 (6.8 MHz)	1:1 (27 MHz)
Min pulse width	888 ns	444 ns	112 ns

The operating frequency will limit minimum pulse width on the quadrature inputs. The input glitch filter pass through pulses that are at least three clock cycles long and filter out anything below two clock cycles.

4.5   Test Results

Functionality was tested in two ways. Manual testing was performed with the quadrature output from AMT203 rotary encoder, to verify counting in both directions worked as expected and followed the encoder movement.

Maximum pulse rates and counting reliability was verified with a microcontroller generated signal source, which emulates quadrature signals. The microcontroller was configured to output 8000 steps at given step frequencies. Pulse width in quadrature signals is twice the step spacing, as only one signal changes state at a time. The step series was repeated several times while data was being read out through SPI bus. A test case was marked as successful only if all steps were correctly counted every time.

Test results are summarized in Table 3. As expected, pulses shorter than two clock periods get rejected by the glitch filter and result in a zero count. When operating close to the minimum pulse width at each voltage, there are some lost steps. This may be caused by asymmetric rise/fall delays, which can slightly shorten some of the pulses as they travel through the logic.

Table 3: Test Results for Different Pulse Rates and Voltages

Step Rate (pulse width)	V_{DD} = 1.8 V	V_{DD} = 3.3 V	V_{DD} = 5.0 V
1.3 MHz (1.50 µs)	OK	OK	OK
2.0 MHz (1.00 µs)	Lost steps	OK	OK
4.0 MHz (0.50 µs)	Zero count	OK	OK
6.0 MHz (0.33 µs)	Zero count	Lost steps	OK
8.0 MHz (0.25 µs)	Zero count	Zero count	OK
12.0 MHz (0.17 µs)	Zero count	Zero count	OK
16.0 MHz (0.13 µs)	Zero count	Zero count	Lost steps

5   Conclusions

A design for interfacing quadrature encoders to a SPI bus has been presented. The design can be used as-is, or the individual components can be repurposed for different applications. The quadrature input section can be modified to output signals directly suitable for different devices, such as stepper drivers. The 16-bit up/down counter block provides a useful tool for measuring many kinds of pulse sources. The chainable SPI interface can be used to simultaneously sample and read from multiple GreenPAK devices, and the data source can come from a counter or an ADC.

Revision History

Revision	Date	Description
1.0	21-Mar-2019	Initial version

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 skilled in the art 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 for development of 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 out of 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 o any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.0 Mar 2020)

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/

Trademarks

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

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/.