

AN-1179 Multiplexing a 4-Digit LED Drive and a 4x4 Keypad Scanner

This application note demonstrates a practical solution for four-digit 7-segment LED display control and a 4x4 keypad scanner. It will also demonstrate sending and receiving data via the I2C bus using one GreenPAK™ IC.

To connect the four-digit LED display and a 4x4 keypad to a microcontroller (MCU), one typically needs twenty I/O pins. However, using the [GreenPAK SLG46538V](#) IC allows us to save MCU hardware resources by using only three I/O pins. The main components used in the design are the asynchronous state machine (ASM), bidirectional I/O pins, the I2C serial communication block, and the combination function macrocells.

Theory of Operation

The basic idea of this app note is to use the bidirectional pins of the SLG46538V IC. This makes it possible to use the same ports simultaneously to connect to the cathodes of the digital LED display and to the sixteen keys, arranged in four rows and four columns. The LED display control and the keyboard scanning control occur at different times in multiplexing mode. The basic scheme is shown in Figure 1, and Figure 2 shows a timing diagram of the multiplexing mode.

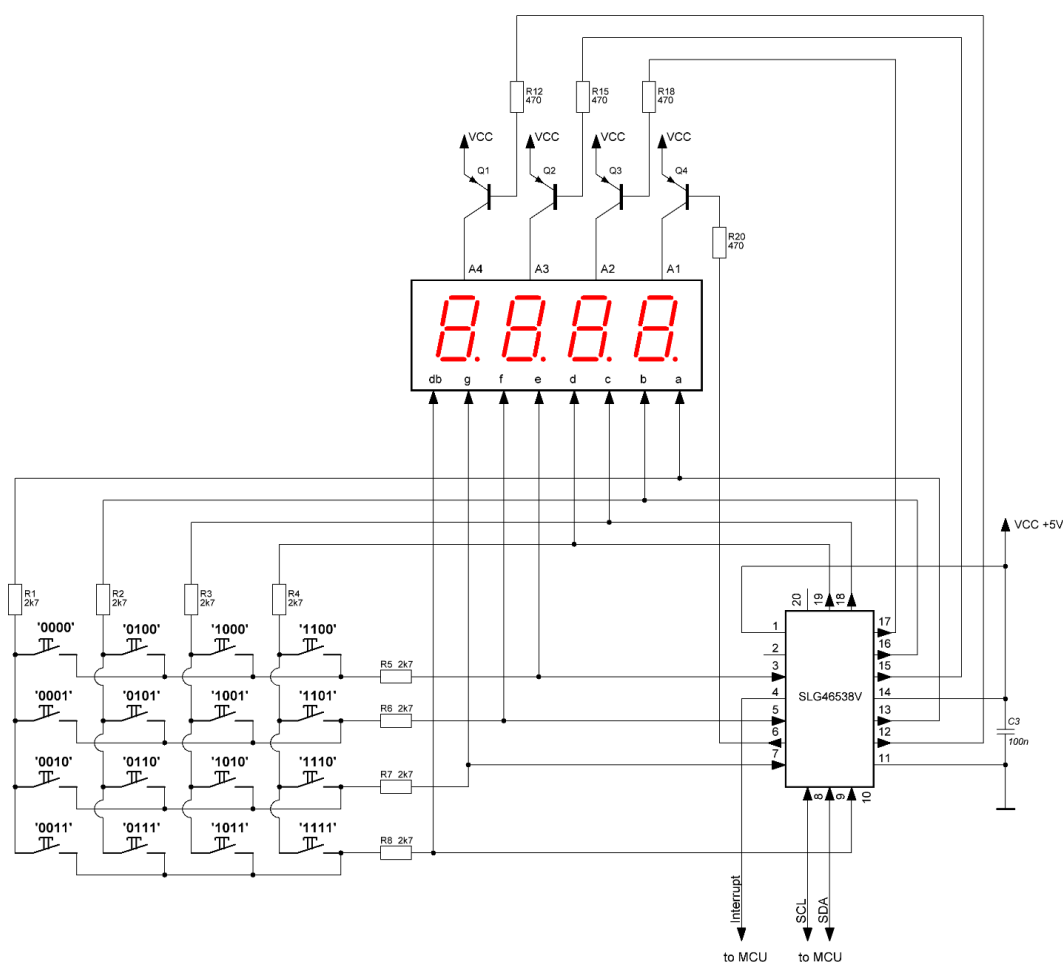


Figure 1. External connections of the SLG46538V GreenPAK IC

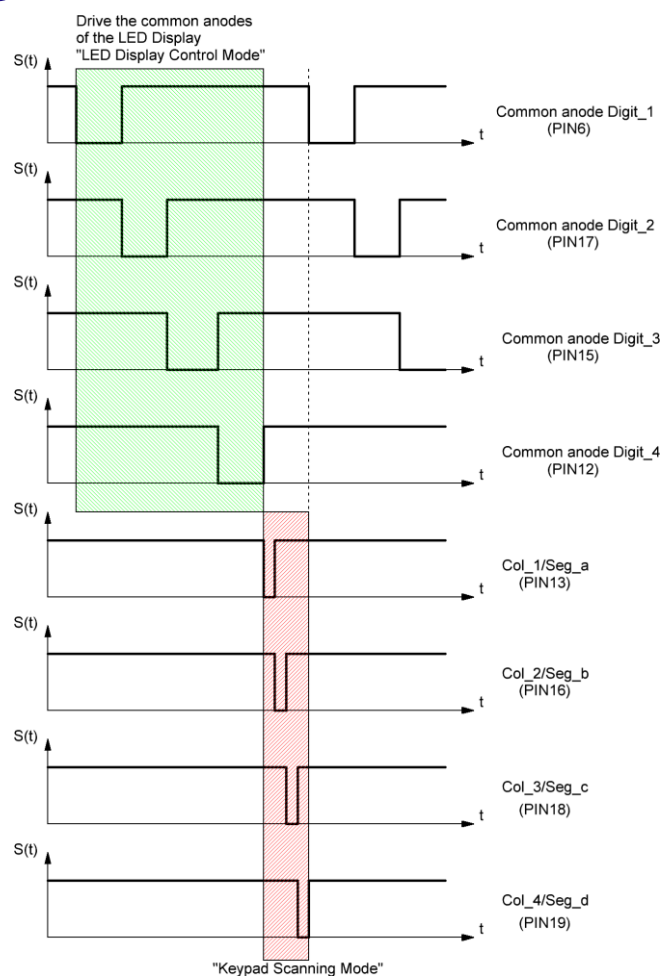
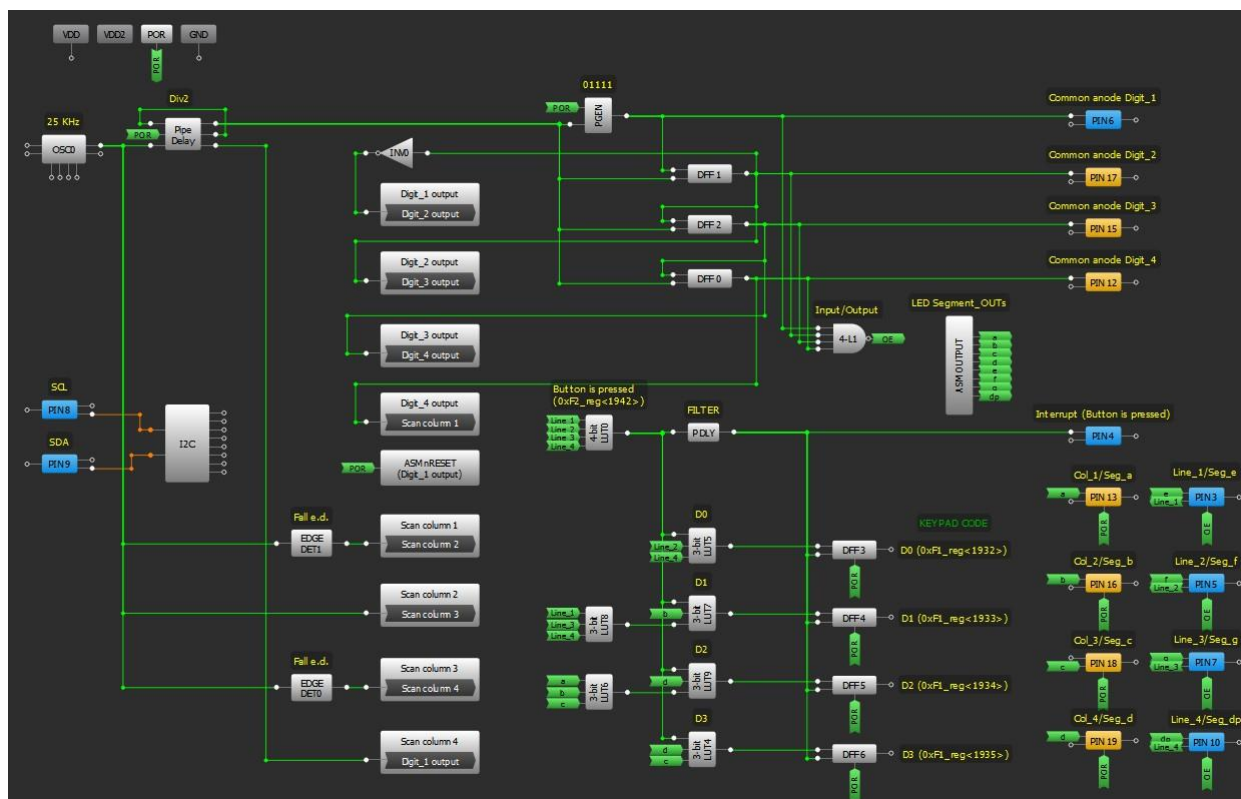


Figure 2. Timing diagram of multiplexing mode

In LED display control mode, the four digits of the display light up, one by one, in about two milliseconds each. Then, during the next ~ 2 ms, the keys are scanned in keypad scanning mode. The whole cycle continues for about 9.7 ms, after which the process is repeated. Transistors Q1-Q4 should be PNP type, with a collector current of at least 200 mA (e.g. PMBT5401).

The data that should be displayed on the digital display and the data that was obtained by scanning the keyboard is transmitted via the I2C bus. The GreenPAK IC generates an interrupt while scanning the keyboard if any key is pressed. The data obtained by scanning is updated at regular intervals when the chip generates an interrupt.

A standard 7-segment LED display can show numbers from 0 to 9, some letters of the alphabet, and a decimal point. The display has a common anode for each digit. In order to light up any segment, a logic LOW level signal has to be transmitted to its cathode.



In order to implement the multiplexed display mode in this design, we used the GreenPAK's ASM (asynchronous state machine) block. The output signal comes from the RC Oscillator Block with frequency:

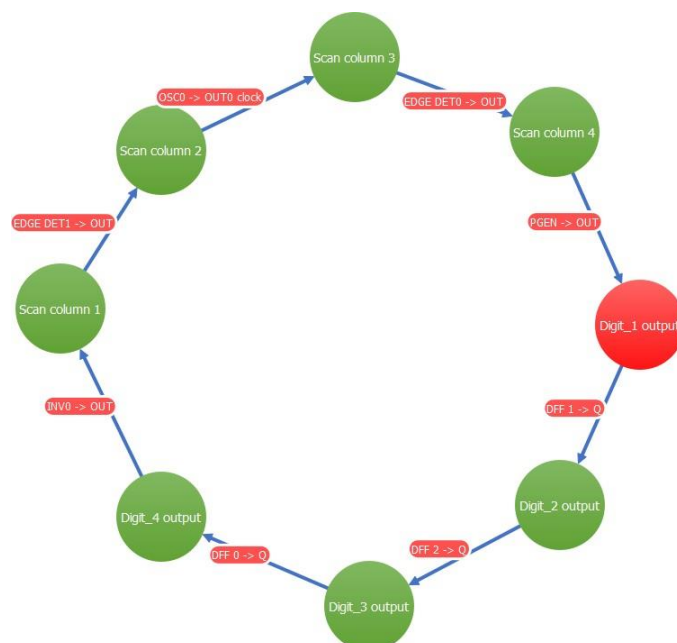


Figure 4. The sequence of switching states in ASM

The microcontroller communicates with the chip via a 2-wire bus, the I2C Serial Communication Block. The MCU initiates the exchange of data with the following command sequence:

Start bit → *Control byte '0x00'(write)* → *Address byte* → *Data byte* →...→ *Stop bit*

In order to show data on the display, it must be written in the following registers of the Connection Matrix Output RAM:

Digits	Address Byte, Hex	Register Bit	Signal Function
Digit 1(Common Anode 1)	0xD1	reg<1679:1672>	RAM 8 outputs for ASM-Digit_1 output
Digit 2(Common Anode 2)	0xD2	reg<1687:1680>	RAM 8 outputs for ASM-Digit_2 output
Digit 3(Common Anode 3)	0xD3	reg<1695:1688>	RAM 8 outputs for ASM-Digit_3 output
Digit 4(Common Anode 4)	0xD4	reg<1703:1696>	RAM 8 outputs for ASM-Digit_4 output

Table 1. The registers for data storage to indicate on LED display

These data are stored in the registers and displayed until they are overwritten via I2C again, or the IC is rebooted. In the latter case, these registers will resume the default value ('0xFF', stored in the NVM).

RAM								
State name	Connection Matrix Output RAM							
	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
Scan colum..	1	1	1	1	1	1	1	0
Digit_1 ou..	1	1	1	1	1	1	1	1
Digit_2 ou..	1	1	1	1	1	1	1	1
Digit_3 ou..	1	1	1	1	1	1	1	1
Digit_4 ou..	1	1	1	1	1	1	1	1
Scan colum..	1	1	1	1	1	1	0	1
Scan colum..	1	1	1	1	1	0	1	1
Scan colum..	1	1	1	1	0	1	1	1

Bulk operations

All to 0 Set

States

ASM

- ☒ State 0 (Scan column 1)
- ☒ State 1 (Digit_1 output)
- ☒ State 2 (Digit_2 output)
- ☒ State 3 (Digit_3 output)
- ☒ State 4 (Digit_4 output)
- ☒ State 5 (Scan column 2)
- ☒ State 6 (Scan column 3)
- ☒ State 7 (Scan column 4)

Figure 5. Connection Matrix Output RAM default configuration

Figure 6 shows example data for transmission via the I2C bus, to be output to the LED display.



LED Digit	Data, BIN	I2C Data, HEX																
	<table><tr><td>dp</td><td>g</td><td>f</td><td>e</td><td>d</td><td>c</td><td>b</td><td>a</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	dp	g	f	e	d	c	b	a	1	0	0	1	0	0	1	0	0x92
dp	g	f	e	d	c	b	a											
1	0	0	1	0	0	1	0											
	<table><tr><td>dp</td><td>g</td><td>f</td><td>e</td><td>d</td><td>c</td><td>b</td><td>a</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	dp	g	f	e	d	c	b	a	1	0	1	1	0	0	0	0	0xB0
dp	g	f	e	d	c	b	a											
1	0	1	1	0	0	0	0											

Figure 6. Example of LED display data registers from the connection matrix output RAM

In LED display control mode, all outputs of the SLG46538V IC (except those that provide data exchange via the I2C bus) are configured as digital push pull outputs.

Implementation - Keypad Scanning

After completing the LED display control mode, the IC goes into the keypad scanning mode. The **DFF0** trigger output is set to a logic HIGH level, and the ASM goes into the "Scan column 1" state. In scanning mode, the transition of the **ASM** from state to state is done by the **OSC0**, **EDGE DET0** and **EDGE DET1** macro-cells. These edge detectors form a short pulse on the falling edge of the **OSC0** signal output, and the transition frequency between the states of the machine in this mode is four times greater than in the display mode. In keypad scanning mode, the **4-bit LUT1** sets the output to a logic LOW level, which switches the **PIN3**, **PIN5**, **PIN7**, and **PIN10** pins into digital input mode. These PINs are configured with 10 kΩ pull-up resistors.

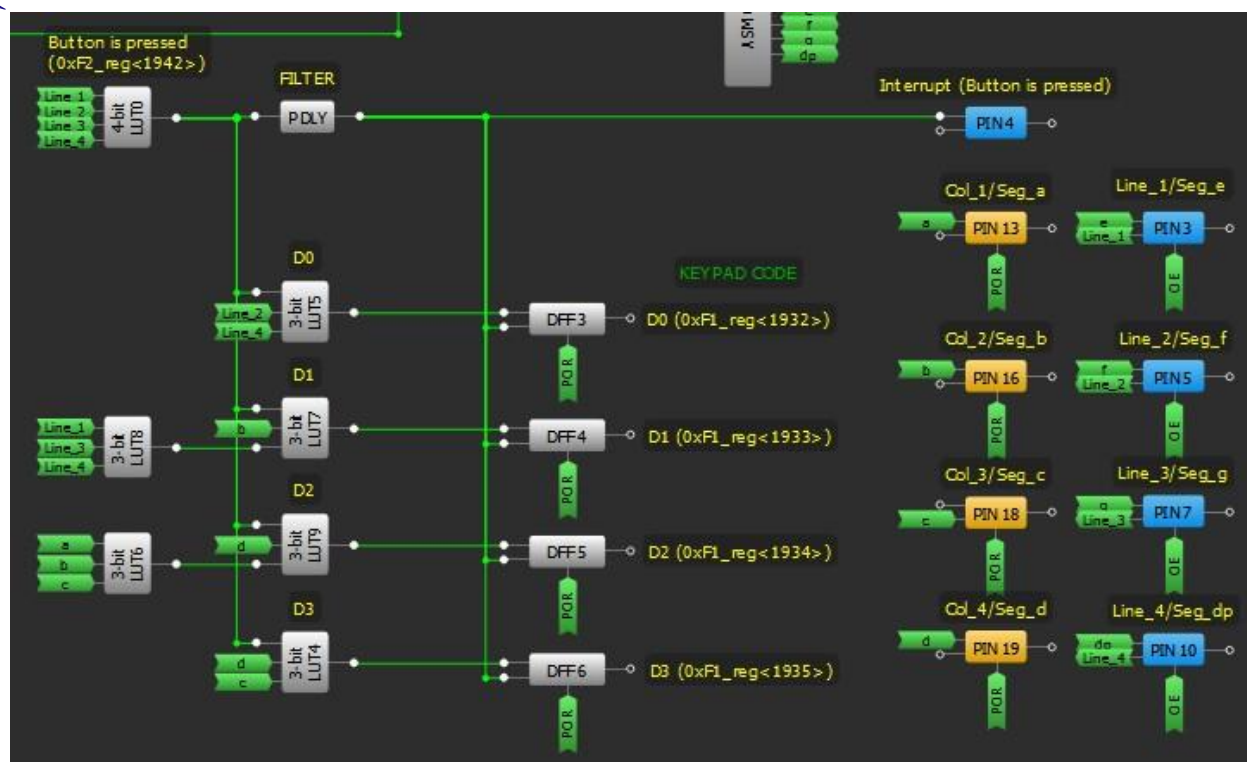


Figure 7. Keypad scanner circuit

The ASM alternates between each pin (PIN13, PIN16, PIN18, PIN19) to send logic LOW level pulses. These PINs are connected to the four columns of the keyboard. The other four PINs (**PIN3**, **PIN5**, **PIN7**, and **PIN10**) are connected to the four rows of the keyboard. When any keypad button is pressed during the scanning process, a logic LOW level of the output pins is transferred to the corresponding input pins. At this time, the **4-bit LUTO** output gets a logic HIGH level.

This design allows the code to determine only one button press. Each button has a unique 4-bit code (see Figure 1), which is formed by the 3-bit LUTs (**LUT4** - **LUT9**) when the button is pressed. This is then stored at the outputs of **DFF3-DFF6** until the next scanning mode cycle. The code can be obtained by the microcontroller via the I2C bus, which should read the following data registers:

KEYPAD CODE	Address Byte, Hex	Register Bit	Register Bit Definition
D0	0xF1	reg<1932>	LUT3_0 / DFF3 Output
D1	0xF1	reg<1933>	LUT3_1 / DFF4 Output
D3	0xF1	reg<1934>	LUT3_2 / DFF5 Output
D4	0xF1	reg<1935>	LUT3_3 / DFF6 Output

Table 2. The registers for storing data identifying the pressed button

To do this, the MCU must initiate the transfer of data via the I2C bus:

Start bit → Control byte '0x00'(write) → Start bit → Control byte '0x01'(read) → Address byte '0xF1' → Data byte → Stop bit.

Interrupt signals come from **PIN4** to the MCU whenever the keypad scanning cycle is active and any button is pressed. The MCU receives the interrupt and reads the button code (the four most significant bits in the '0xF1' byte).

The algorithm for the analysis of button presses must be implemented in the controller's program, along with the debounce filter.

The total time for the display scanning modes is approximately 9.7 ms, but if necessary this can be increased or decreased. To do this, the '**CLK**' predivider value and '**OUT0**' second divider value in the **OSCO** macrocell must be changed via I2C:

OSC0

Control pin mode: Power down

OSC Power Mode: Force Power On

Clock selector: OSC

EXT CLK Pin selector: PIN 20 (IO17)

Fast start-up: Disable

RC OSC Frequency: 25 kHz

'CLK' predivider by: 1

'OUT0' second divider by: 24

'OUT1' second divider by: 1

Information

Frequency

Clock output configuration:

RC OSC Output	Value
CLK /4	RC OSC Freq. /4
CLK /12	RC OSC Freq. /12
CLK /24	RC OSC Freq. /24
CLK /64	RC OSC Freq. /64
OUT0	RC OSC Freq. /24
OUT1	RC OSC Freq.

Figure 8. OSC0 default settings

Address Byte	Register Bit	Signal Function	Register Bit Definition
A7	reg<1340:1339>	OSC Clock Pre-divider for 25kHz/2MHz	00: Div1 01: Div2 10: Div4 11: Div8
A8	reg<1349:1347>	Internal OSC 25kHz/2MHz Frequency Divider Control for matrix input <27> (OUT0)	000: OSC/1 001: OSC/2 010: OSC/3 011: OSC/4 100: OSC/8 101: OSC/12 110: OSC/24 111: OSC/64

Table 3. The registers for the OSC0 macrocell dividers setting

Keep in mind that after restarting the IC, all settings and data registers return to their default values.

Conclusion

Using the SLG46538V GreenPAK IC and its resources (I2C, ASM, Bi-directional PINs, etc.), it is easy to build a device for displaying data on four 7-segment LED displays with a 4x4 keyboard scanner using only three MCU I/O pins. This is especially effective for the manufacturing of simple products in large quantities, because the free I/O pins of the MCU can be used for other tasks.

This solution can be used to develop devices such as an electronic code door lock, signaling devices, simple programmable power supplies, or clocks and timers. In addition, this device can be realized as a "shield" board for Arduino or other platforms.

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

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.