7-Segment LED Control with GreenPAK
SLG46537

The application note gives step-by-step guidelines for creating a 7-Segment LED Control with GreenPAK using a SLG46537V device. A unique set of components of the SLG46537 allows the creation of such a system.

The application note comes complete with design files which can be found in the Reference section.

Contents

Figures

Tables

1. Terms and Definitions

ASM	Asynchronous State Machine
DFF	D flip-flop
LED	Light-Emitting Diode

2. References

For related documents and software, please visit:

GreenPAK Programmable Mixed-Signal Products | Renesas

Download our free Go Configure Software Hub [1] to open the .gp5 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 provides a complete library of application notes [4] featuring design examples as well as explanations of features and blocks within the Renesas IC.

[1] Go Configure Software Hub, Software Download and User Guide, Renesas Electronics

[2] AN-1097 7-segmebnt LED Control with GreenPAK.gp, GreenPAK Design File, Renesas Electronics

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

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

3. Introduction

This app note will explain how to control 7- segment LED displays using two new components in GreenPAK: the I2C block and the asynchronous state machine (ASM). It incorporates I2C I/O Controller techniques discussed in AN-1090.

4. 7-Segment Control Overview

A 7-segment display has 7 distinct sections that can be powered on individually. The display can show digits 0-9 depending on which segments are lit. It can also display letters A-F for use in hexadecimal applications. These displays typically include a segment for the Decimal Place (dp) and an enable pin (see **Error! Reference source not found.**).

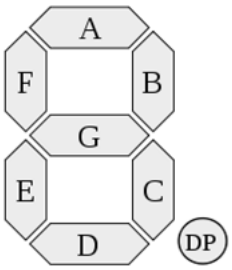

Figure 1. Generic Segment Names

In this App Note we will be driving a 2-Digit Common-Anode LED 7-Segment Display (see Figure 2) with the pinout shown in Figure 3. Since this is a Common-Anode display, its enable pins are active HIGH and its segments are active LOW, meaning if pin EN1 is HIGH and the pin F1 is LOW, the F1 segment will be turned on. The same result could be achieved with a Common-Cathode LED 7-Segment with inverted logic.

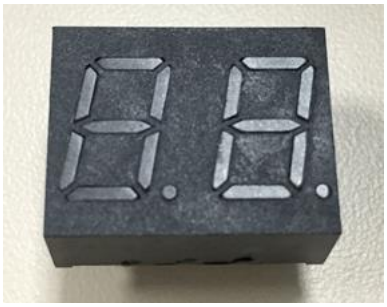

Figure 2. 2-Digit 7-Segment LED Display

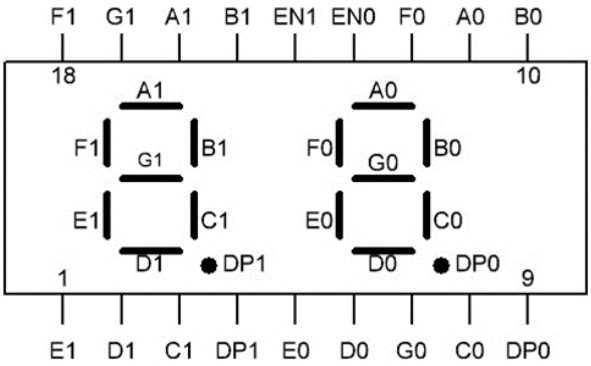

Figure 3. 2-Digit LED 7-Segment Display

5. GreenPAK Configuration

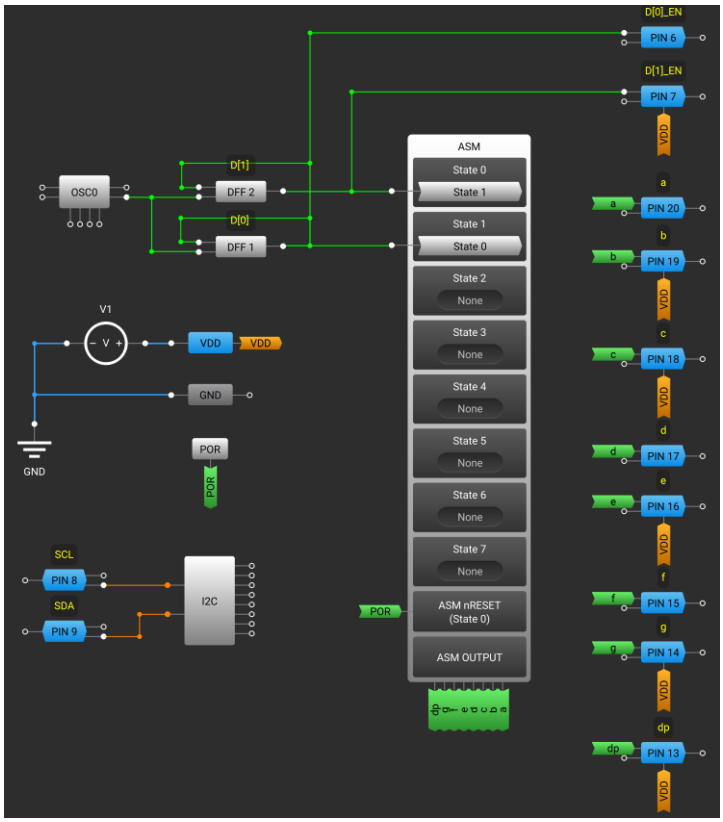

Figure 4. 2-Digit GreenPAK Block Diagram

DFF1 and DFF2 are used to create alternating enable signals which also toggle between ASM states D[0] and D[1] (see Figure 4). When the output of DFF1 is high, the ASM is in state D[0], D[0]_EN is high, and Digit 0 is enabled via the EN0 pin on the LED display. When the output of DFF2 is high, the ASM is in state D[1] and Digit 1 is enabled (see Figure 5).

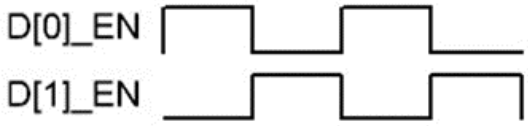

Figure 5. 2-Digit Enable Signals

The LED segment control bits for Digit 1 and Digit 0 are stored in State D[1] and State D[0] of the ASM. If you open the ASM Editor, you will see that the state machine is extremely simple, as shown in Figure 6. 2- Digit ASM Configuration. The segment control signals a, b, c, d, e, f, g, and dp are connected to their corresponding pins for both Digit 1 and Digit 0 as shown in Figure 7. Although both digits are connected to the segment control signals, only one digit enable pin is asserted at a time.

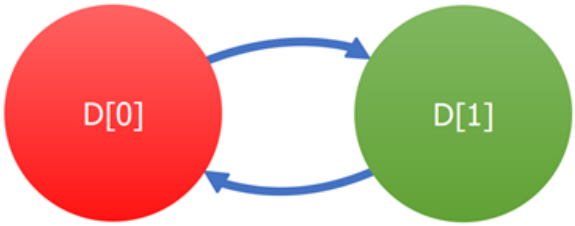

Figure 6. 2-Digit ASM Configuration

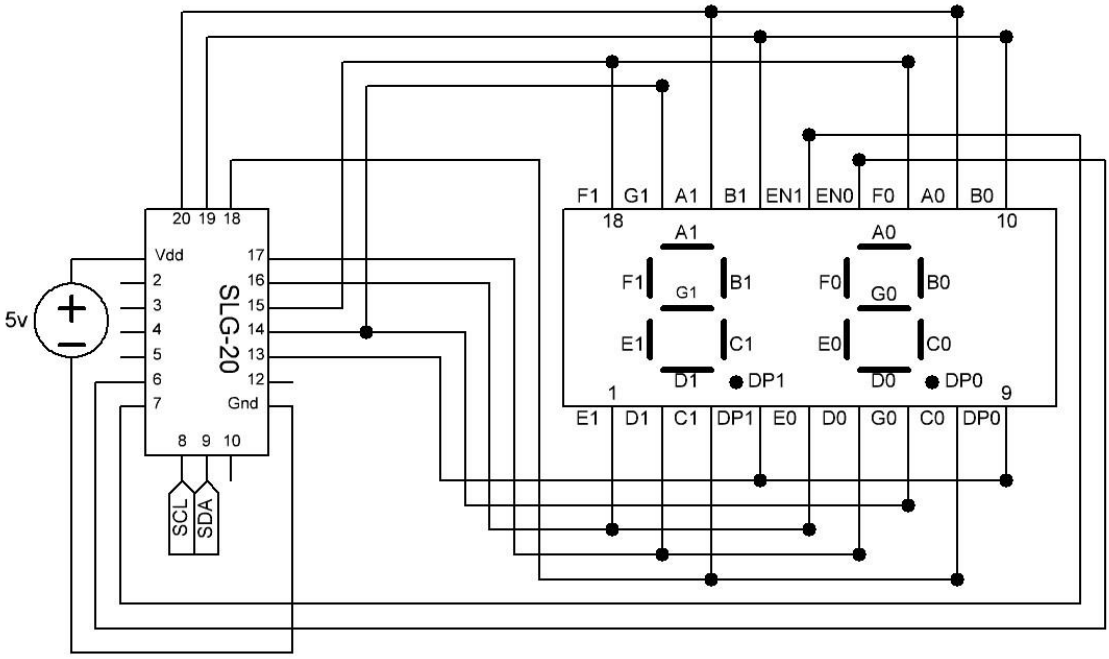

Figure 7. External Connections between GreenPAK and 2-Digit 7-Segment Display

6. 4-Digit GreenPAK Configuration

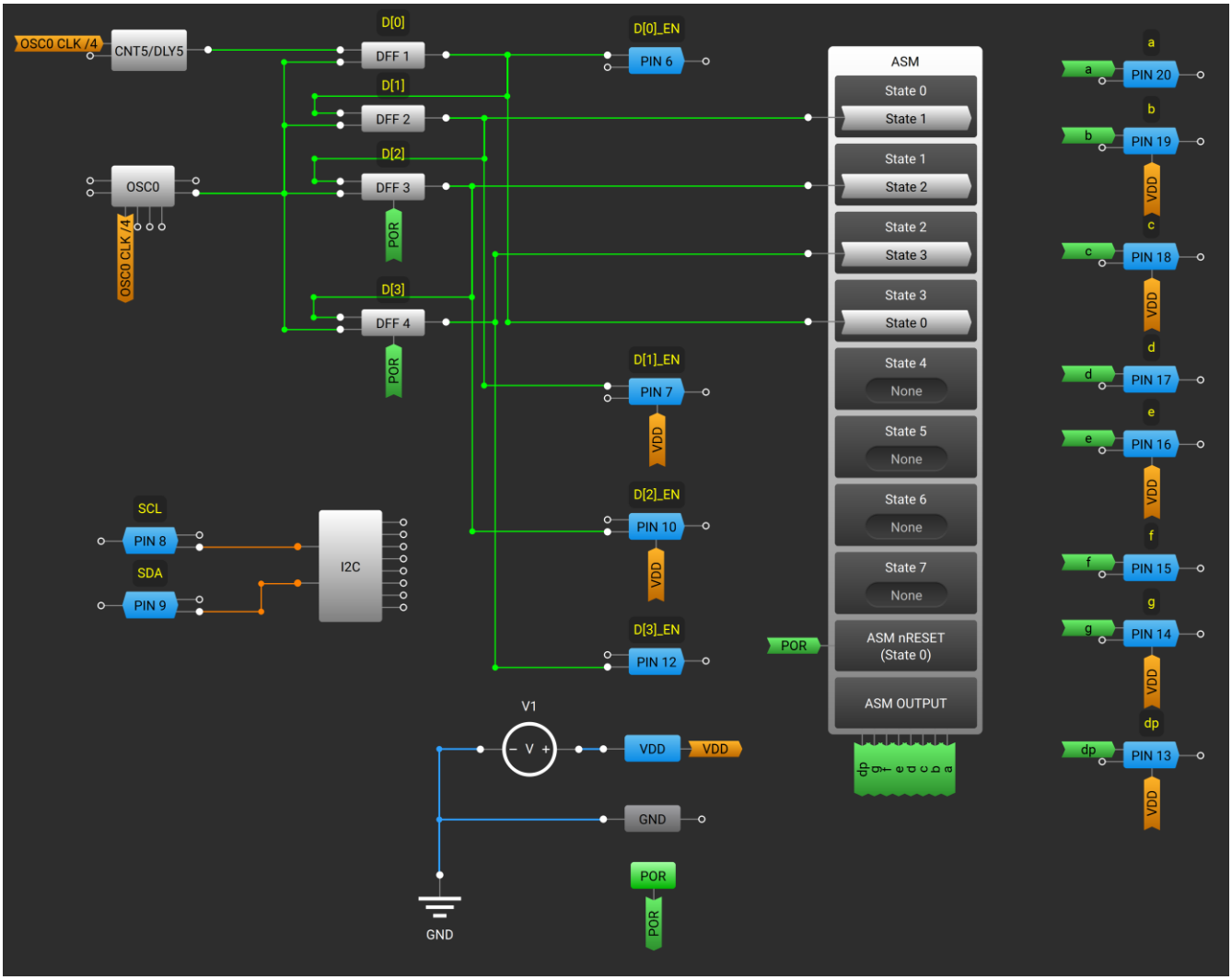

Figure 8. 4-Digit GreenPAK Block Diagram

In order to expand our design to drive a 4- Digit 7-Segment display, we added two more DFFs, two more ASM states, and two more enable pins (see Figure 8). Pin10 is now the enable for Digit 2 (D[2]), and Pin12 is the enable for Digit 3 (D[3]). Instead of toggling between two enable signals, we now cascade through four enable signals so that only one digit is enabled at a time (see Figure 9).

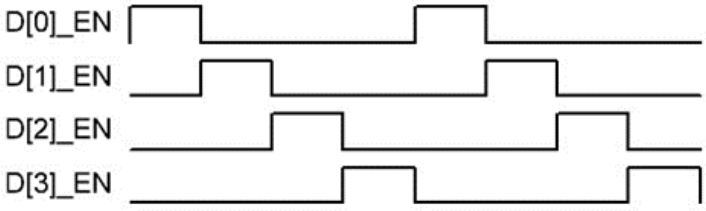

Figure 9. 4-Digit Enable Signals

Figure 10 shows the ASM configuration to transition between 4 digits. Figure 11 shows the external connection routing between the segment control signals and each of their corresponding pins. The routing of each enable signal corresponds directly to its enable pin as illustrated in Figure 11 and Table 1.

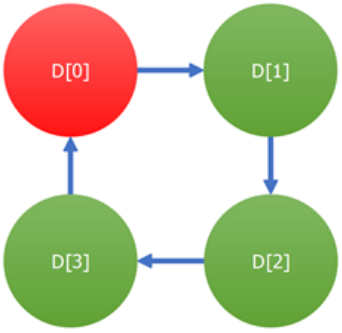

Figure 10. 4-Digit ASM Configuration

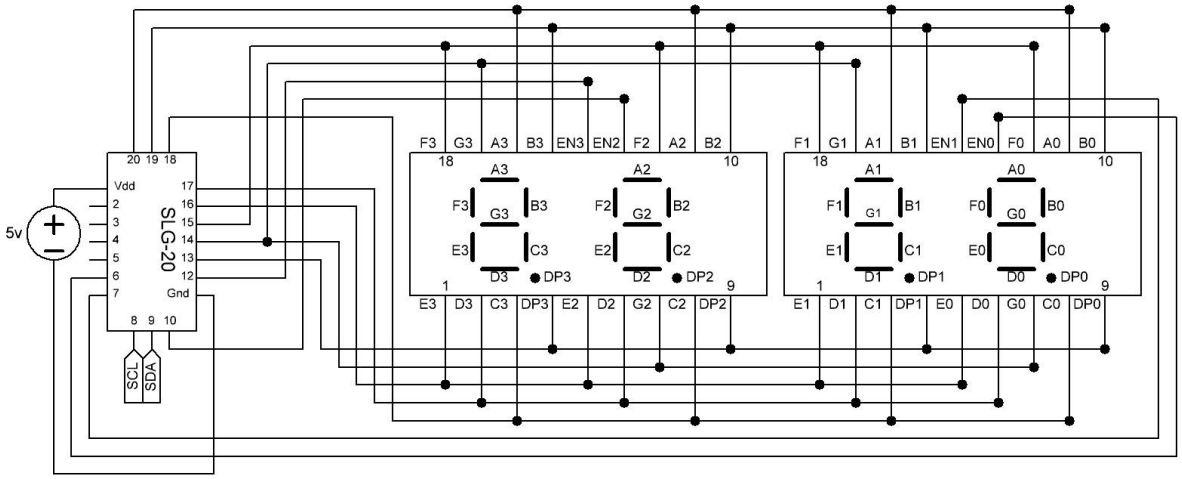

Figure 11. External Connections between GreenPAK and 4-Digit 7-Segment Display

Table 1. Signal Routing

GreenPAK	Signal	LED
6	D[0]_EN	EN0
7	D[1]_EN	EN1
10	D[2]_EN	EN2
12	D[3]_EN	EN3

This design could be expanded further to handle up to 7 digits since we have four more ASM states, three more GPIO pins for enable signals (Pin3, Pin4, and Pin5), and plenty of unused DFFs. However, at that point the duty cycle of each digit would be low enough that it may be difficult to read.

7. I2C Control with an Arduino Uno

In this section, we will use I2C to write directly to the ASM RAM table. Each state (D[3:0]) in the ASM accesses 1 byte of RAM. Each byte has 8 bits which control the 8 segments on the display. (To read more about how to use I2C with GreenPAK5, read AN-1090 or refer to the part's datasheet).

To write to the GreenPAK's registers via I2C, you need 3 bytes:

Control Byte 0x00 Control Code = '0000', Block Address = '000', R/W = '0'

Address Byte 0xD0/0xD1 Register Addresses of ASM RAM for States D[0] & D[1]

Data Byte 0x?? Data to send via I2C

Table 2 shows the I2C Data Byte needed to make the 7-Segment LED Display show HEX 0:F. Notice that the Decimal Point bit is HIGH for every hex value, meaning that the active-LOW decimal point is off. If you want to include a decimal point after your digit, all you need to do is BITWISE AND 0x7F to your I2C Data Byte. For example, to make a digit display '0.', the Data Byte would be: 0xC0 & 0x7F = 0x40.

We wrote a simple Arduino program to send digits to the GreenPAK via I2C to illustrate the control system. This program increments D[3:2] from 'AA' to 'FF', and increments D[1:0] from '0.0' to '9.9'. The code used is included in Appendix A and the Arduino file is included in the app note materials.

Table 2. Hex translation to I2C Data Byte

Hex Value	7-Segment Display	LCD Segment								Byte	I2C Data
		dp	g	f	e	d	c	b	a		
0		1	1	0	0	0	0	0	0	11000000	0xC0
1		1	1	1	1	1	0	0	1	11111001	0xF9
2		1	0	1	0	0	1	0	0	10100100	0xA4
3		1	0	1	1	0	0	0	0	10110000	0xB0
4		1	0	0	1	1	0	0	1	10011001	0x99
5		1	0	0	1	0	0	1	0	10010010	0x92
6		1	0	0	0	0	0	1	0	10000010	0x82
7		1	1	1	1	1	0	0	0	11111000	0xF8
8		1	0	0	0	0	0	0	0	10000000	0x80
9		1	0	0	1	0	0	0	0	10010000	0x90
A		1	0	0	0	1	0	0	0	10001000	0x88
B		1	0	0	0	0	0	1	1	10000011	0x83
C		1	1	0	0	0	1	1	0	11000110	0xC6
D		1	0	1	0	0	0	0	1	10100001	0xA1
E		1	0	0	0	0	1	1	0	10000110	0x86
F		1	0	0	0	1	1	1	0	10001110	0x8E

8. Conclusion

Thanks to new features like I2C and the ASM in GreenPAK, it is a fairly simple task to leverage GreenPAK to drive a series of 7- Segment LED Displays. This design has the flexibility to turn on any combination of LED segments while only requiring two microcontroller pins for I2C: SCL and SDA. With some adjustments, this design could be expanded to drive up to seven 7-segment displays.

9. Revision History

Revision	Date	Description
1.00	Feb 22, 2016	Initial release.
2.00	Apr 10, 2026	The part number has been changed from SLG46531V to SLG46537V.

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.