SLG46824/6 MTP RA MCU Programming Example

This document describes how to use the Renesas RA MCU e2 studio C-code to program the SLG46824/6 devices. Firmware designers can analyze that code to create a modified version that is compatible with their unique microcontroller. This document also contains links to design files (see References section).

Contents

Figures

1. Introduction

This application note describes how to program the SLG46824/6 GreenPAK™ Multiple-Time Programmable (MTP) device using the Renesas RA MCU e2 studio C-code.

Most GreenPAK devices are One-Time Programmable (OTP), meaning that once their Non-Volatile Memory bank (NVM) is written, it cannot be overwritten. GreenPAKs with the MTP feature (for example SLG46824 and SLG46826) have a different type of NVM memory bank that can be programmed more than once.

Renesas developed a program for its RA MCUs that allows users to program the NVM memory of the MTP GreenPAK with just a few simple steps. In this application note, the SLG46826 is used as the GreenPAK model and FPB-RA4E1 as the MCU. Renesas offers a Quick-Connect board called the QCIOT-SLG46826POCZ which facilitates rapid prototyping of the SLG46826V. This board was used to program the SLG46826 for the given example in this document.

A sample code is provided for the FPB-RA4E1 MCU using an open-source platform based on C/C++. Designers should extrapolate the techniques used in the RA MCU code for their specific platform.

For specific information regarding I^2C signal specifications, I^2C addressing, and memory spaces, refer to the GreenPAK In-System Programming Guide provided on the SLG46826 product page. This application note provides a simple implementation of this programming guide.

1.1 Terminology

Term	Definition
EEPROM	Electrically erasable programmable read-only memory.
I2C	Inter-integrated circuit.
MTP	Multiple-time programmable.
NVM	Non-volatile memory.
OTP	One-time programmable.

2. Exporting GreenPAK NVM Data from a GreenPAK Design File

A very simple GreenPAK design will be constructed to illustrate how to export the NVM data. The design shown in Figure 1 is a window comparator. Open the sample schematic file provided or create your design and save it to your local machine.

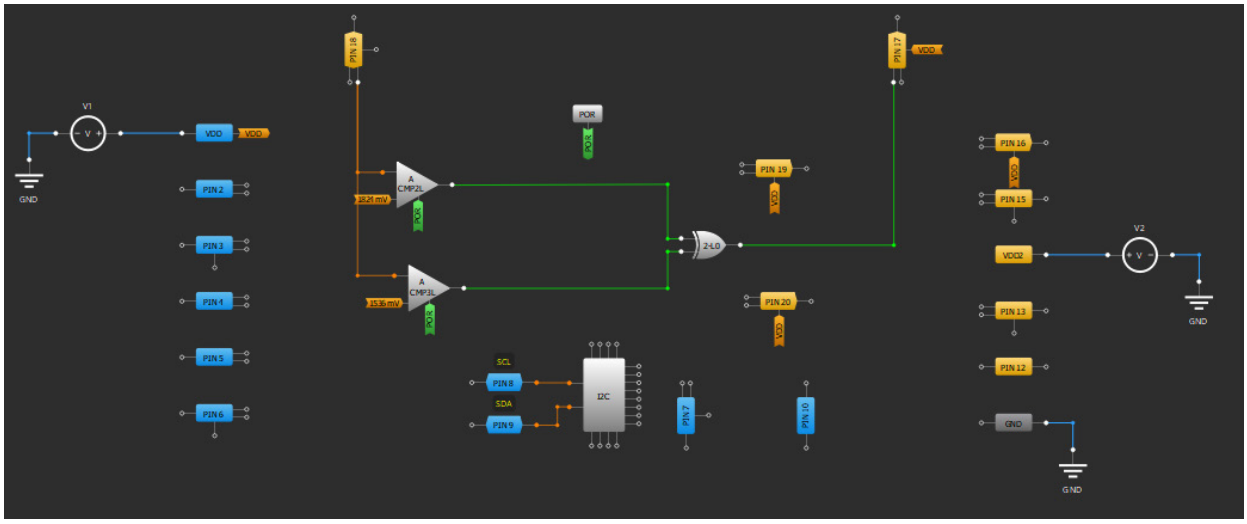

Figure 1. Simple GreenPAK Design in a SLG46826

To export the information from this design, select **File** → **Export** → **Export NVM** (see Figure 2).

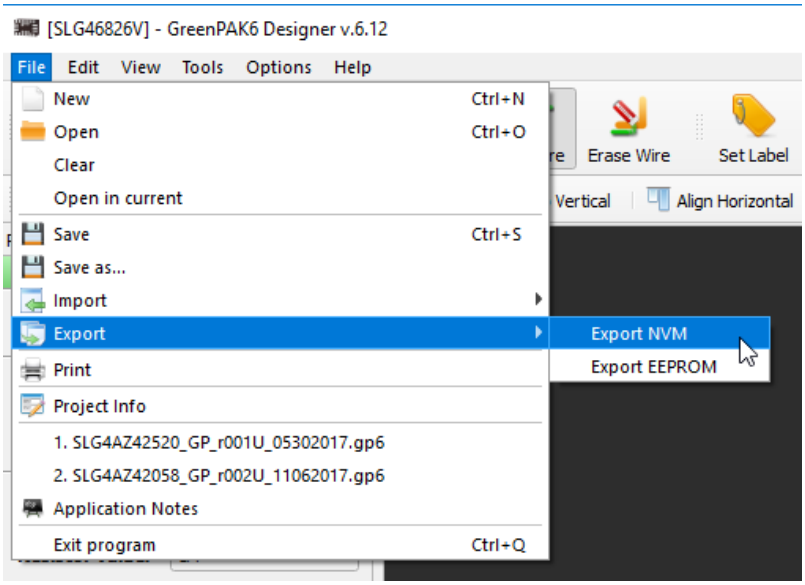

Figure 2. Exporting the NVM Data

Select the “Intel HEX Files (*.hex)” as the file type and save the file name as **NVM_data.hex** (see Figure 3).

Important: Name the file as **NVM_data.hex** only. No other name is acceptable.

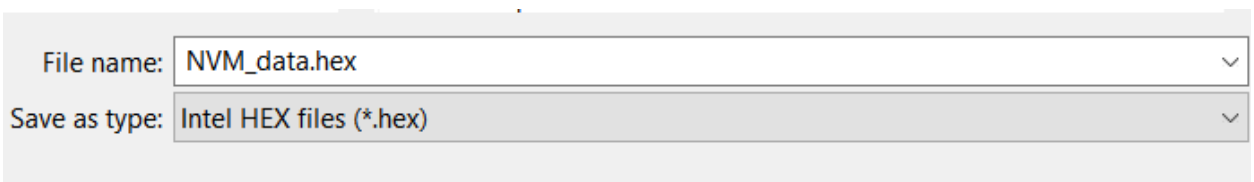

Figure 3. Save as NVM_data.hex File

3. Flash the NVM Data to the MTP GreenPAK

To import the sample project shared with this application code, follow the steps below.

1. Open e2 studio.

   Note: Ensure to use the latest e2 studio (v24.4.0 or later); **FSP version v5.2.0** is the required minimum.

2. Select **File → Import → General → Existing project into workspace**. Click on the radio button *Select archive file*. Click on *Browse* to locate the **.zip** file (see Figure 4).

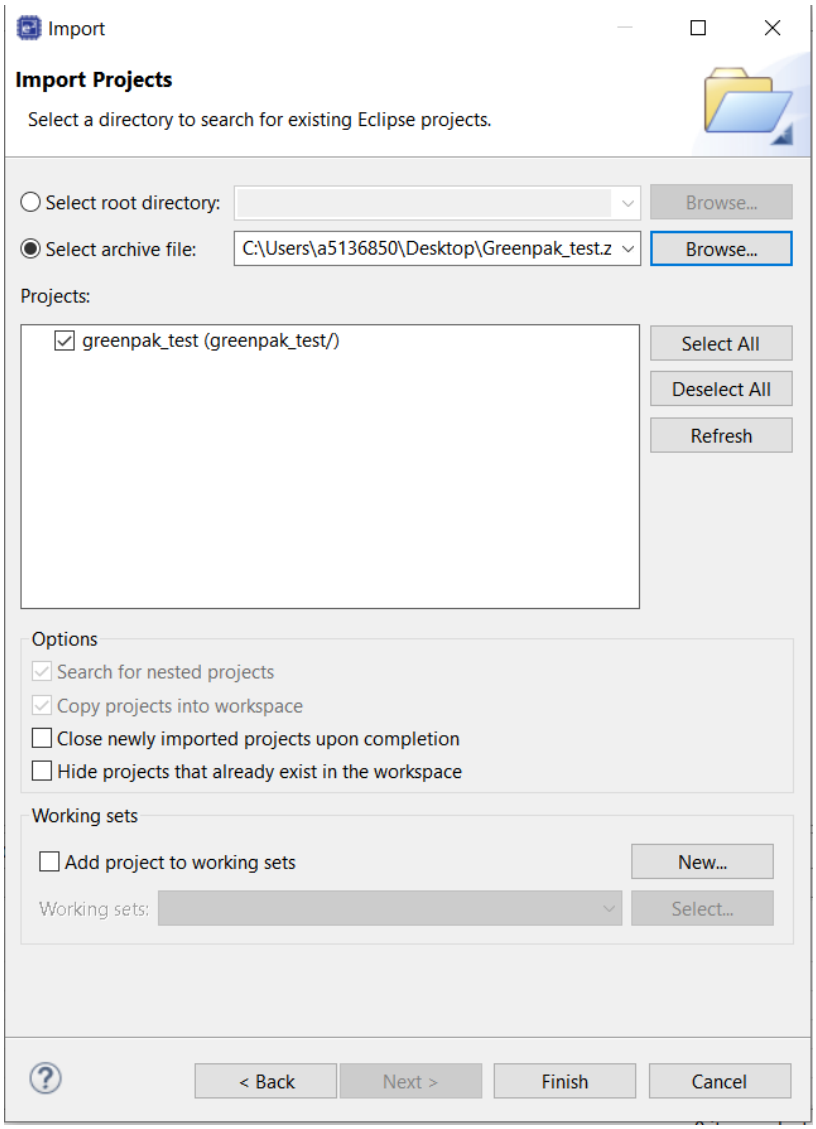

Figure 4. Import Projects Window

3. Open the workspace in the local machine by navigating to the **greenpak_test/scr** folder location and right-clicking on *SYSTEM → Show In → System Explorer*.

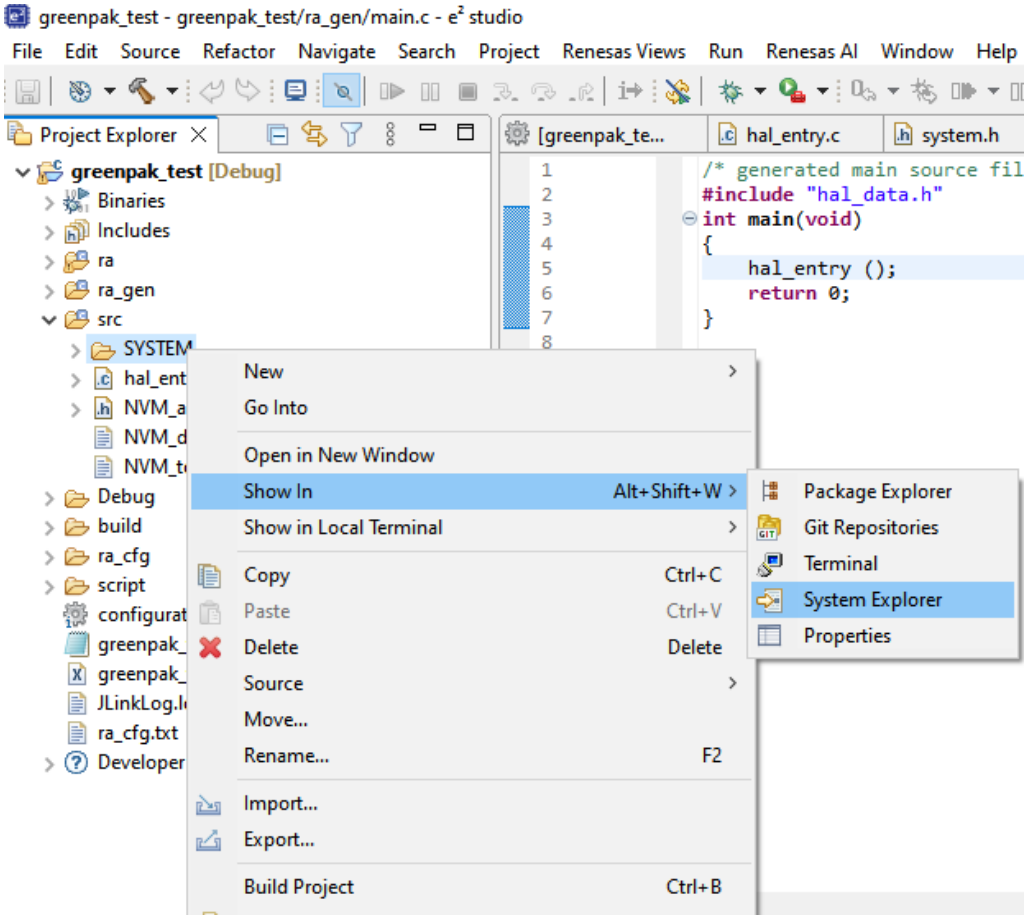

Figure 5. Opening the Workspace in a Local Machine

4. Copy the saved **NVM_data.hex** file in this location.

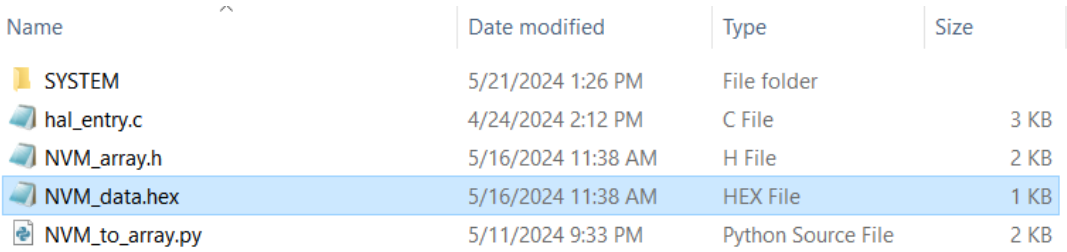

Figure 6. Copying the Exported Hex File

5. Use the hammer icon to build the project on e2 studio. The project should compile with **0 errors and 0 warnings**.

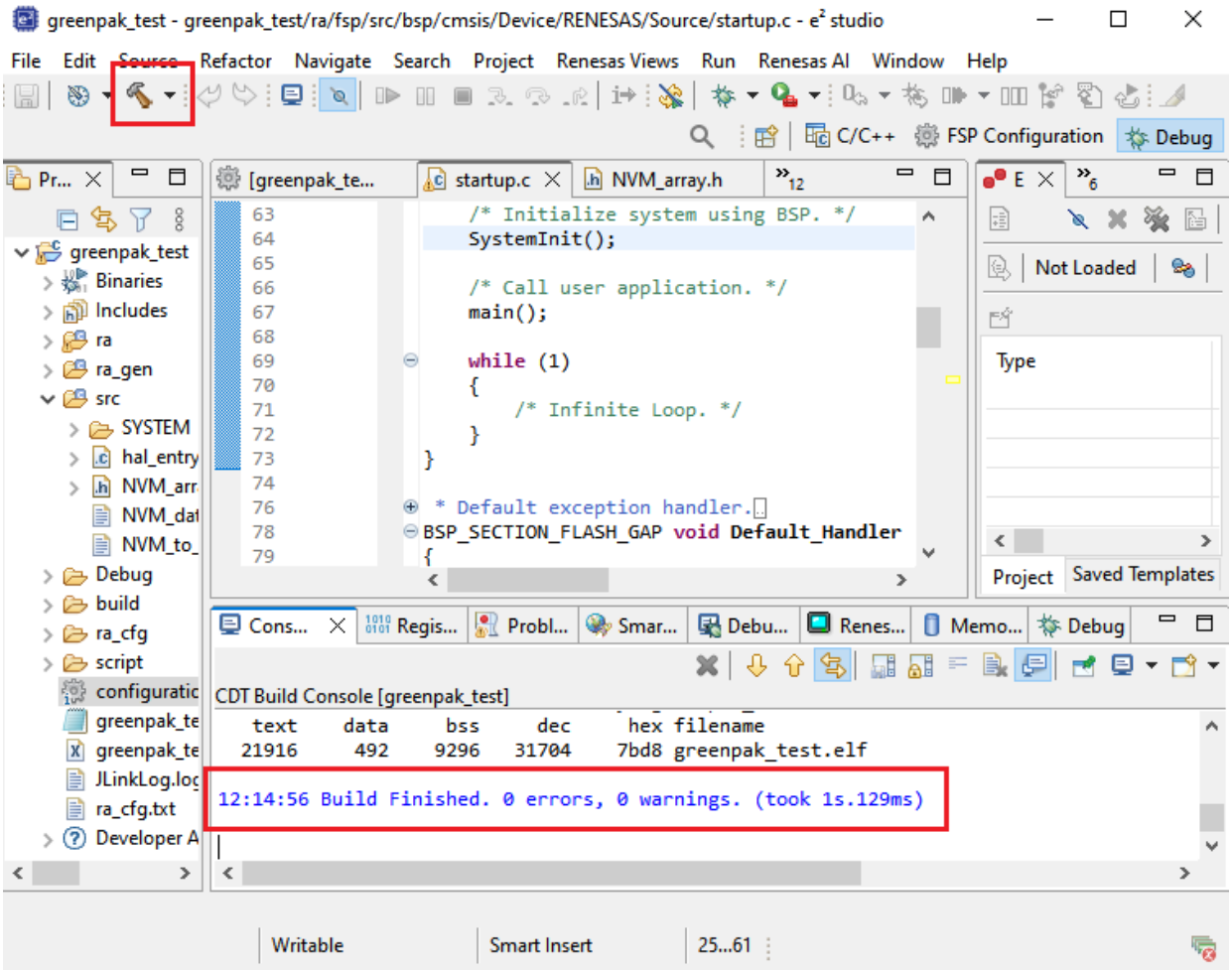

Figure 7. Building and Compiling with 0 Errors and 0 Warnings

6. Connect J1 of the QCIOT-SLG46826POCZ board to PMOD1 of the FPB-RA4E1 board as shown in Figure 8.

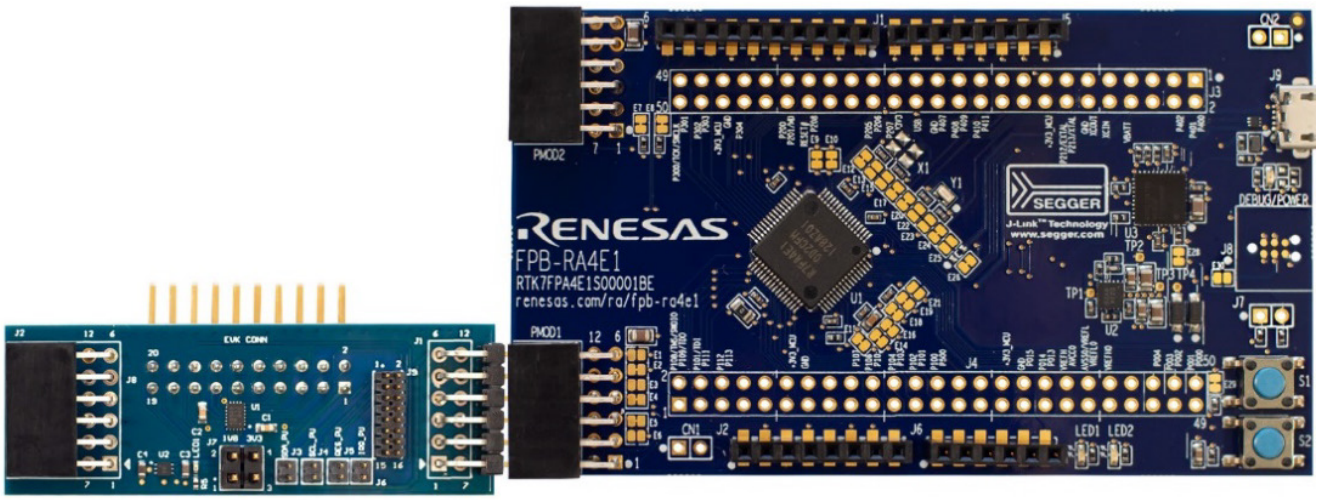

Figure 8. RA MCU and QCIOT Board Connection

Note: The FPB-RA4E1 and QCIOT-SLG46826POCZ boards both have 4.7kΩ internal pull-up resistors. For the QCIOT-SLG46826POCZ board, internal SDA and SCL pull-ups are on J3 and J4 respectively. The FPB-RA4E1 also has an internal pull-up, so there is no need to connect pull-ups on the QCIOT-SLG46826POCZ board. If the MCU board does not have internal pull-ups, then the QCIOT-SLG46826POCZ pull-ups can be connected.

7. Connect the FPB-RA4E1 board to the PC USB port and click on the debug icon. Click the play button icon twice.

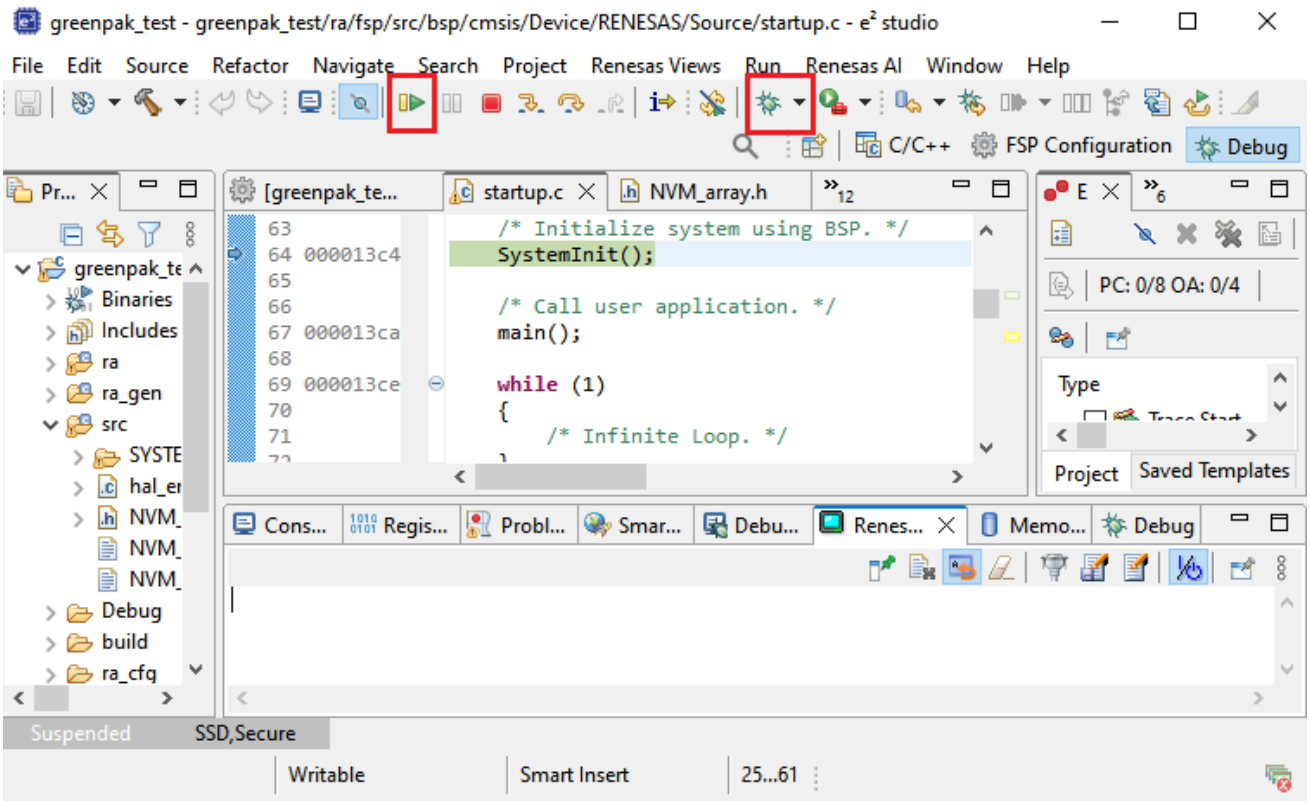

Figure 9. Flash and Run the Code

4. Renesas Debug Console Output

To view the Debug Console details, select *Renesas Views → Debug → Renesas Debug Virtual Console*.

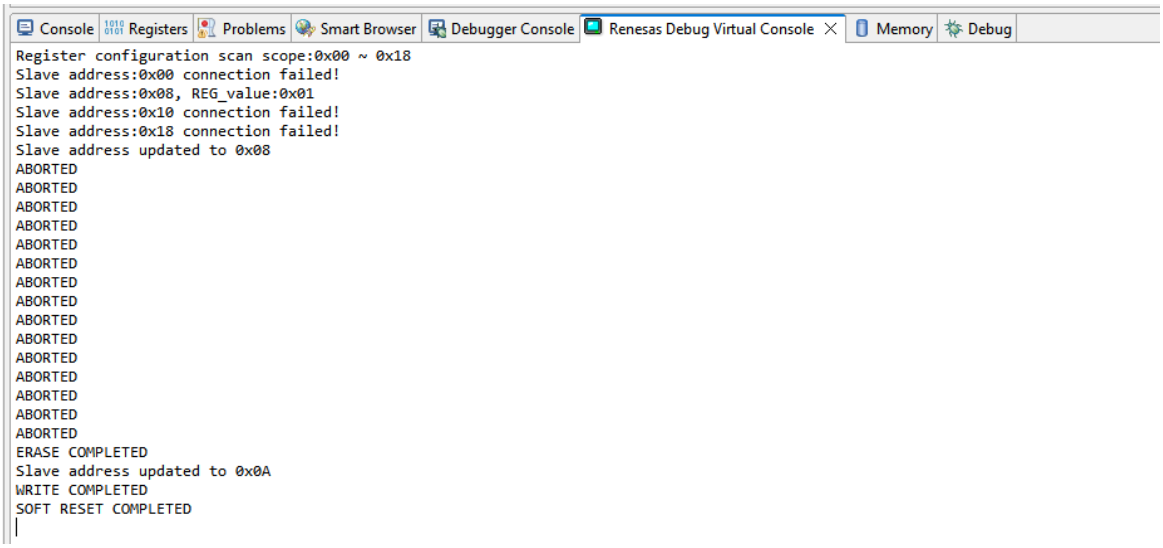

Figure 10. Debug Console Output

5. Software Flow

This section provides a brief overview of the software flow.

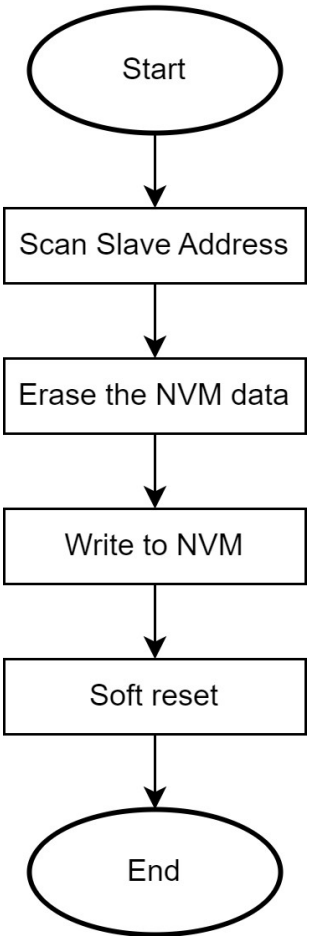

Figure 11. Software Flow Diagram

- **Scan Slave Address** – The sample code scans for the slave (target) address and displays the selected slave address. The scan scope can be 0x00, 0x08, 0x10 or 0x18.
- **Erase the NVM data** – Clears the NVM data prior to writing the data to the NVM memory.
- **Write to NVM** – Writes 16 bytes to the NVM memory using the page write.
- **Soft reset** – To reload the register data from the NVM into the registers, a soft reset is required.

6. Programming Tips and Best Practices

This section provides various programming tips and practices for preventing common pitfalls associated with erasing and writing to the NVM address space of the SLG46824/6 device.

6.1 Executing Precise 16-Byte NVM Page Writes

Avoid these three techniques when writing data to the NVM:

1. Page writes with less than 16 bytes.
2. Page writes with more than 16 bytes.
3. Page writes that don't begin at the first register within a page (in other words, 0x10, 0x20, etc.).

If any of the above techniques are used, the MTP interface will disregard the I²C write to avoid loading the NVM with incorrect information.

Recommendation: Perform an I²C read of the NVM address space after writing to verify the correct data transfer.

6.2 Transferring NVM Data into the Matrix Configuration Registers

When the NVM is written, the matrix configuration registers are not automatically reloaded with the newly written NVM data. The transfer must be initiated manually by cycling the PAK VDD or by generating a soft reset using I²C. By setting register <1601> in address 0xC8, the device re-enables the Power-On Reset (POR) sequence and reloads the register data from the NVM into the registers.

6.3 Resetting the I²C Address after an NVM Erase

When the NVM is erased, the NVM address containing the I²C slave address is set to 0000. After the erase, the chip will maintain its current slave address within the configuration registers until the device is reset as described above. Once the chip has been reset, the I²C slave address must be set in address 0xCA within the configuration registers each time the GreenPAK is power-cycled or reset. Continue this until the new I²C slave address page has been written in the NVM.

7. Errata Discussion

When writing to the "Page Erase Byte" (Address: 0xE3), the SLG46824/6 produces a non-I²C compliant ACK after the "Data" portion of the I²C command. This behavior might be interpreted as a NACK depending on the implementation of the I²C master.

To accommodate for this behavior, abort the I²C transaction after waiting for 40ms as shown in Figure 12. This section of code provides a delay of 40ms at the end of every I²C command in the **eraseslg46826()** function. This function is used to erase the NVM and EEPROM pages.

```
p_ctrl->communication_finished = false;
err            =
    p_ctrl->p_comms_i2c_instance->p_api->write(p_ctrl->p_comms_i2c_instance->p_ctrl,

R_BSP_SoftwareDelay (40, BSP_DELAY_UNITS_MILLISECONDS);
R_IIC_MASTER_Abort (&g_i2c_master0_ctrl);
```

Figure 12. ACK Behavior Modification

Despite the presence of a NACK, the NVM and EEPROM erase functions will execute properly. For a detailed explanation of this behavior, refer to "Issue 2: Non-I2C Compliant ACK Behavior for the NVM and EEPROM Page Erase Byte" in the SLG46824/6 Errata document.

8. Conclusion

This application note provided a description for the process of programming the MTP SLG46826 GreenPAK using the Renesas RA MCU. For any questions regarding the sketch, contact one of Renesas' Field Application Engineers or post your question on our forum. For more in-depth information regarding MTP programming registers and procedures, refer to the In-System Programming Guide.

9. References

GreenPAK Designer Software, Software Download and User Guide

- Download for opening **.gp6** files and viewing the proposed circuit design

The GreenPAK Cookbook

- Download for opening and viewing **.gp6** files for the proposed circuit design

GreenPAK Development Tools

- Use the GreenPAK development tools to freeze the design into your own customized IC in minutes

GreenPAK Application Notes

- A complete library of application notes featuring design examples as well as explanations of features and blocks within the IC

In-System Programming Guide

- In-system programming procedures for SLG46824 and SLG46826

For related documents and software not listed above, visit GreenPAK™ Programmable Mixed-Signal Products.

10. Revision History

Revision	Date	Description
1.00	Sep 12, 2024	Initial release.

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.