TLE9893_2QKW62S_NVM_PROG_UCODE

About this document

Scope and purpose

The aim of this guide is to present the scope, the implementation, the algorithm and a demonstration of the **TLE9893_2QKW62D_NVM_PROG_UCODE** example code for the TLE989x Infineon Embedded Power ICs based on Arm® Cortex® M3. This example code can be found in the Keil µVision Pack Installer.

The full functionalities and characteristics of the embedded power devices are described in the datasheets and user's manual. Please refer to these documents for more detailed information. Furthermore, a low level (line-by-line) description of the code is not the aim of this document, although occasionally some code blocks might be reported if necessary to the comprehension.

Note: *The following information is given as a hint for the implementation of the system only and shall not be regarded as a description or warranty of a certain functionality, condition or quality of the referred devices or presented software example.*

Intended audience

Design engineers, system engineers, embedded power designers

Table of contents

1 Introduction

In the TLE9893_2QKW62S_NVM_PROG_UCODE example, data are programmed into the UCODE (user code) area of the FLASH1 bank of the device (see Figure 1).

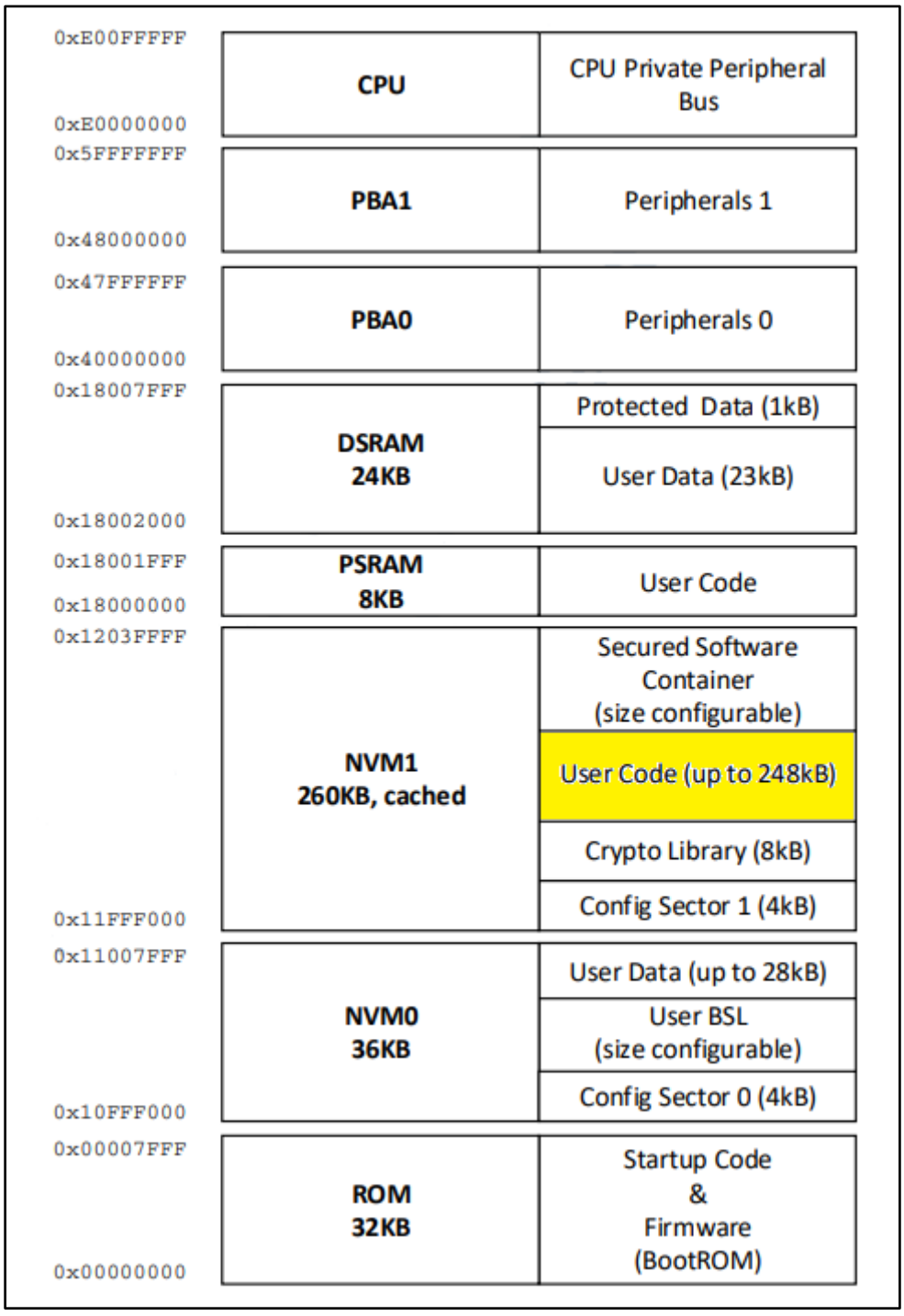

Figure 1 Memory Modules Overview

The success or fail of the write operation is indicated by LEDs via the GPIOs:

- A rising edge on P0.1 indicates that the data were successfully written into the user code area of the FLASH1 bank.
- A rising edge on P1.0 occurs if any error happened during the write operation.

In Figure 2, a successful writing operation is shown capturing the GPIO P0.1 (yellow) signal and the GPIO P1.0 (blue) signal.

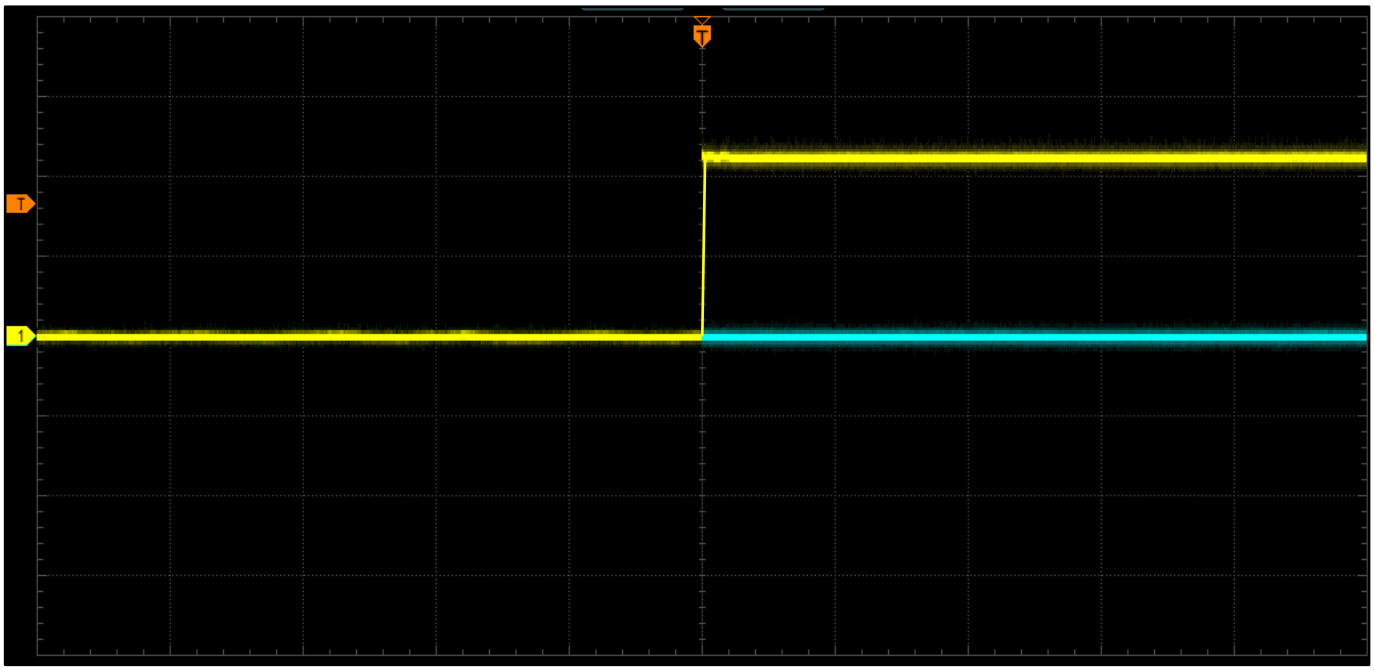

Figure 2 Capture of GPIO P0.1 (yellow) signal and GPIO P1.0 (blue) signal

The written data are then read and a checksum calculation is done over them:

- A rising edge on the GPIO P0.2 occurs if the calculation is validated correctly.
- A rising edge on the GPIO P1.1 is set in case of an error in reading or validating the data.

2 Hardware

This chapter shows how to run the TLE9893_2QKW62S_NVM_PROG_UCODE example with the TLE988X/TLE989X evaluation board. For this the project must be opened and compiled.

Figure 3 shows the TLE988X/TLE989X evaluation board. The application code must be loaded via a debugger (e.g. ULINK or J-Link) to the board. The board must be powered with 12V (red and black banana connectors).

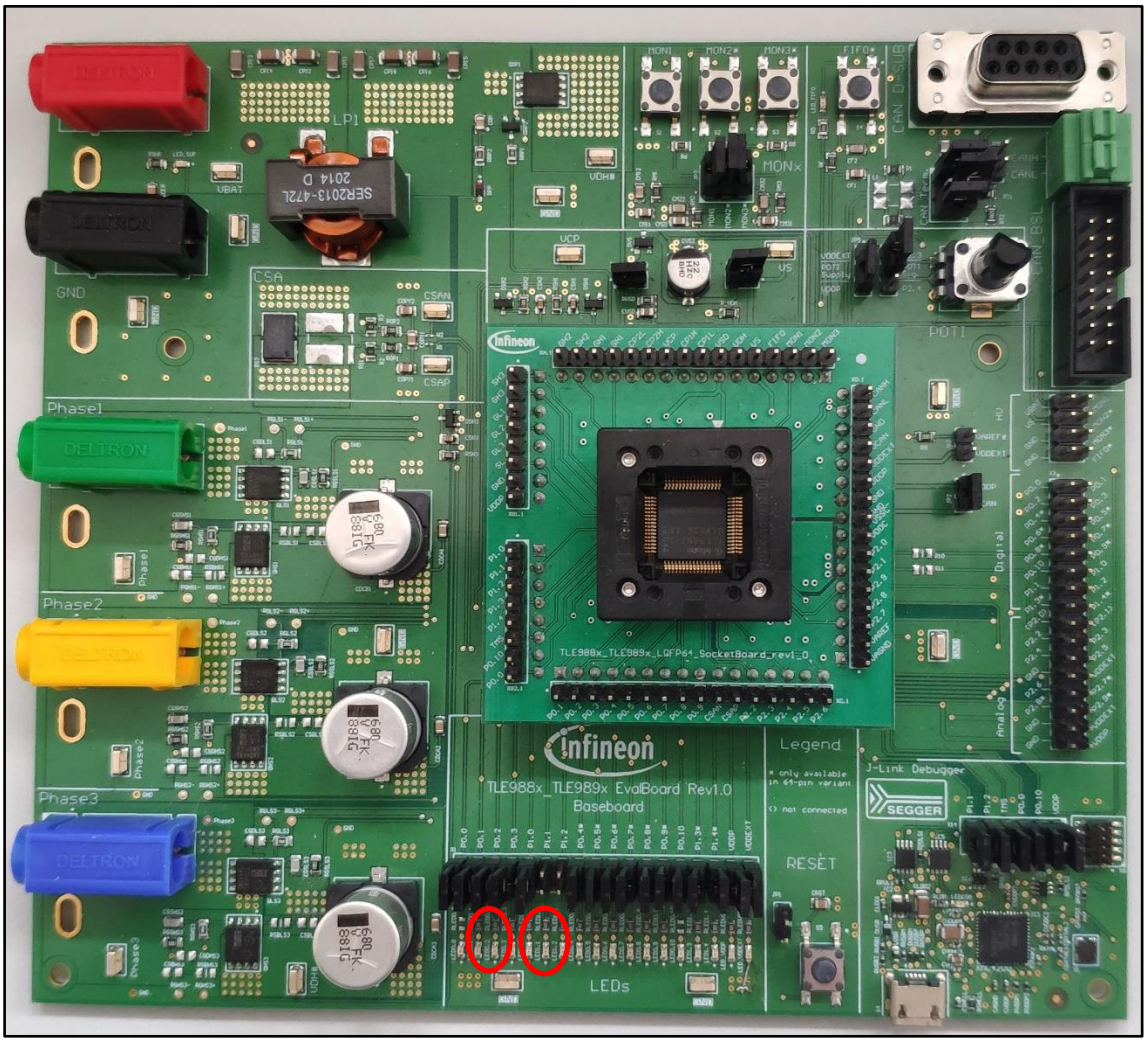

Figure 3 TLE988X/TLE989X evaluation board

3 Implementation

This chapter shows the process to follow to get a working TLE9893_2QKW62_NVM_PROG_UCODE example.

3.1 Get the example via the Pack Installer for Keil

Open the Pack Installer within the Keil IDE. See Figure 4 below.

Choose the appropriate device (here TLE9893_2QKW62S) on the left-hand side. On the right-hand side, select the tab Examples, where you can access the TLE9893_2QKW62S_NVM_PROG_UCODE example.

Clicking on “Copy” will copy the example on your computer and open it.

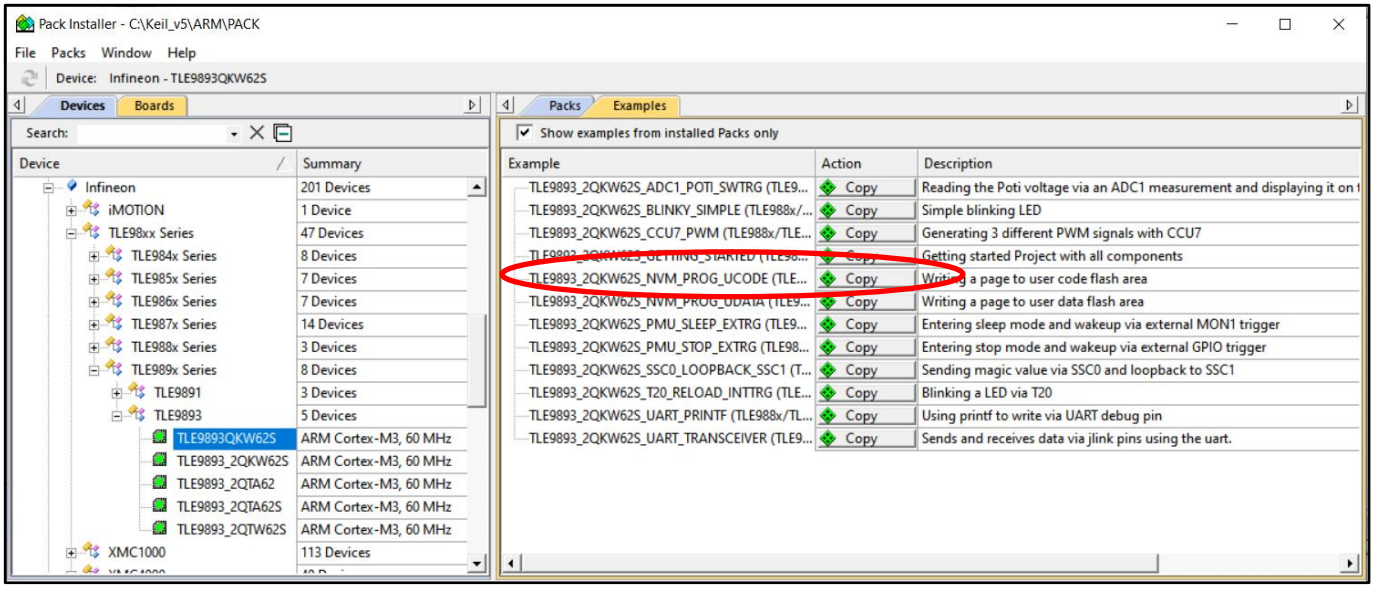

Figure 4 Keil Pack Installer

3.2 Configuration

In order to see the configured pins, start the tool Config Wizard. It is available within the Keil IDE through a shortcut in the Tools menu.

The Config Wizard opens and shows an overall status of the current pin configuration. In the example configuration, all the GPIOs P0.1, P0.2, P1.0 and P1.1 are used as outputs.

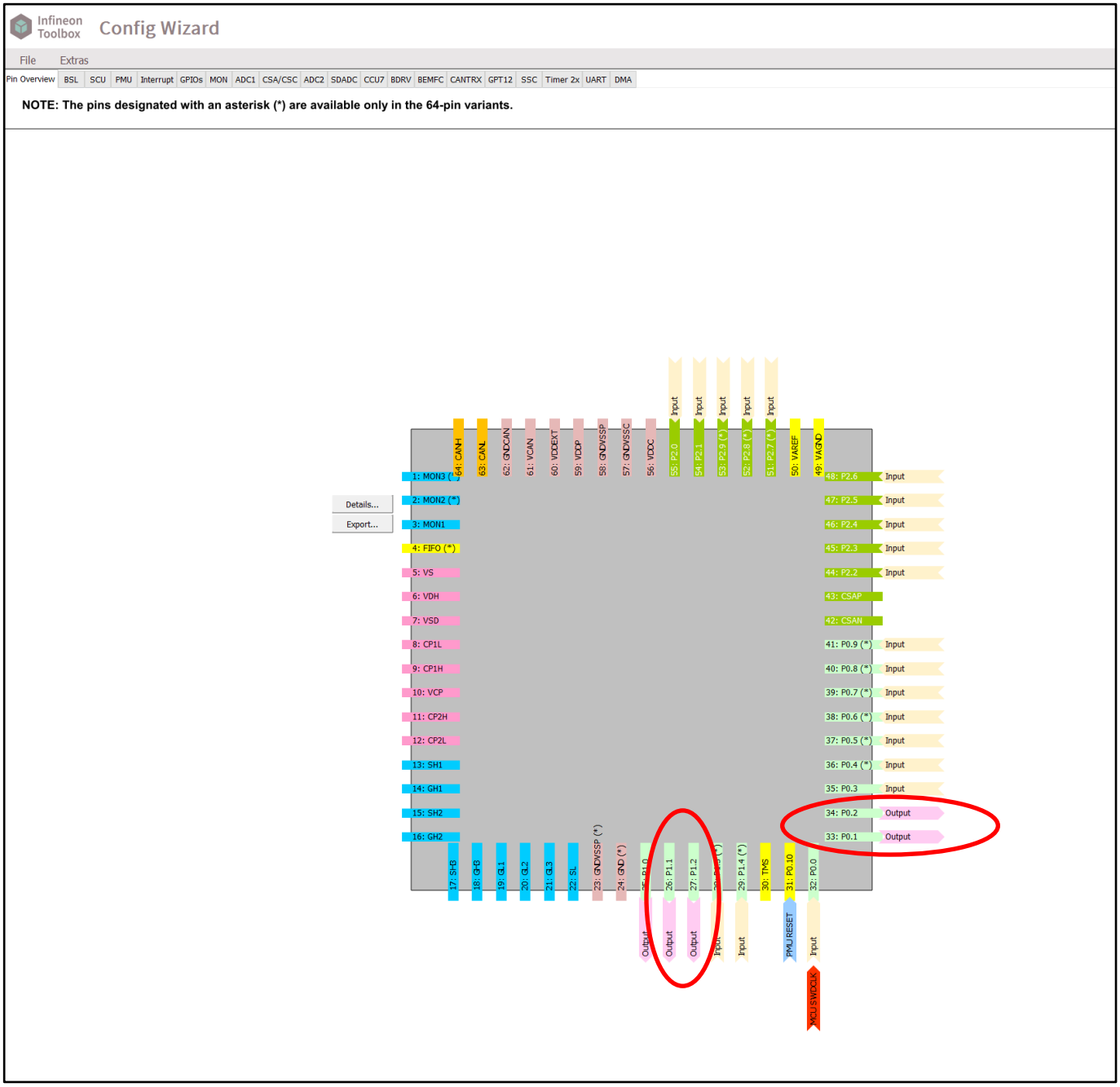

Figure 5 Config Wizard pin overview

Finally, save your configuration to take these changes into account (File -> Save).

3.3 Sample code

Figure 6 shows the application code of the TLE9893_2QKW62S_NVM_PROG_UCODE example.

In a first step an initial checksum over the data is calculated (see line 106-110).

The data are written to the NVM in line 119 with the API method `user_nvm_page_write(UC_FLASH1_START, &s_pageWrite)`. If the write operation is successful, the GPIO P0.1 is switched to high (line 130). Otherwise if any problems occur during the page programming, the GPIO P1.0 is set to high (line 135).

Before the endless loop, the data from the first page are read and verified. The read pointer is set to the beginning of the first page in the user data area (line 139). Then all the data from the page are read and a checksum is calculated (line 142-146). This checksum and the initial calculated checksum are compared. If the two checksums match, the GPIO P0.2 is switched to high (line 151). In case the checksums are not identical, the GPIO P1.1 is switched to high (line 156).

```
105   /* Calculate a simple xor checksum for the data quote */
106   u8_xorChecksum = 0;
107   for (u32_idx = 0; u32_idx < UC_FLASH_PAGE_SIZE; u32_idx++)
108   {
109     u8_xorChecksum ^= u8p_dataQuoteWrite[u32_idx];
110   }
111
112   /* Suspend and remember all enabled interrupts */
113   (void)CMSIS_Irq_Dis();
114
115   /* Open SOW */
116   PMU_serviceFailSafeWatchdogSOW();
117
118   /* Write to the first page into the user code area of FLASH1 */
119   s32_returnValue = user_nvm_page_write(UC_FLASH1_START, &s_pageWrite);
120
121   /* Close SOW by regular WDT trigger */
122   PMU_serviceFailSafeWatchdog();
123
124   /* Reenable suspended interrupts */
125   CMSIS_Irq_En();
126
127   if (s32_returnValue == ERR_LOG_SUCCESS)
128   {
129     /* Indicate a successful page write by LED P0.1 */
130     GPIO_setP01State(GPIO_STATE_HIGH);
131   }
132   else
133   {
134     /* Indicate a page write error by LED P1.0 */
135     GPIO_setP10State(GPIO_STATE_HIGH);
136   }
137
138   /* Set read pointer to first page of the user code area */
139   u8p_userData = (uint8 *) UC_FLASH1_START;
140
141   /* Calculate a simple xor checksum for the data quote */
142   u8_verifyChecksum = 0;
143   for (u32_idx = 0; u32_idx < UC_FLASH_PAGE_SIZE; u32_idx++)
144   {
145     u8_verifyChecksum ^= u8p_userData[u32_idx];
146   }
147
148   if (u8_verifyChecksum == u8_xorChecksum)
149   {
150     /* Indicate a successful checksum verification by LED P0.2 */
151     GPIO_setP02State(GPIO_STATE_HIGH);
152   }
153   else
154   {
155     /* Indicate a wrong checksum verification by LED P1.1 */
156     GPIO_setP11State(GPIO_STATE_HIGH);
157   }
158
159   for (;;)
160   {
161     /* Main watchdog service */
162     (void) PMU_serviceFailSafeWatchdog();
163   }
164 }
```

Figure 6 TLE9893_2QKW62S_NVM_PROG_UCODE application code

References

See the code examples at **www.infineon.com**

Revision history

Document version	Date of release	Description of changes
1.0	2021-10-22	Initial version
1.1	2022-10-13	Editorial changes

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-10-13

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.