Customer training workshop: Code_Flash_Sector_Data_Updating for KIT_T2G-B-H_EVK

TRAVEO^{TM} T2G CYT4BF series Microcontroller Training
V1.0.0 2022-11

Please read the **Important notice and warnings** at the end of this document

Scope of work

- This code example demonstrates the use of the Flash HAL driver to perform code flash Read/Write/Erase basic operate and update sector data online. The running steps logs on a UART terminal emulator.

- Device
  - The TRAVEO^{TM} T2G CYT4BFBCH device is used in this code example.

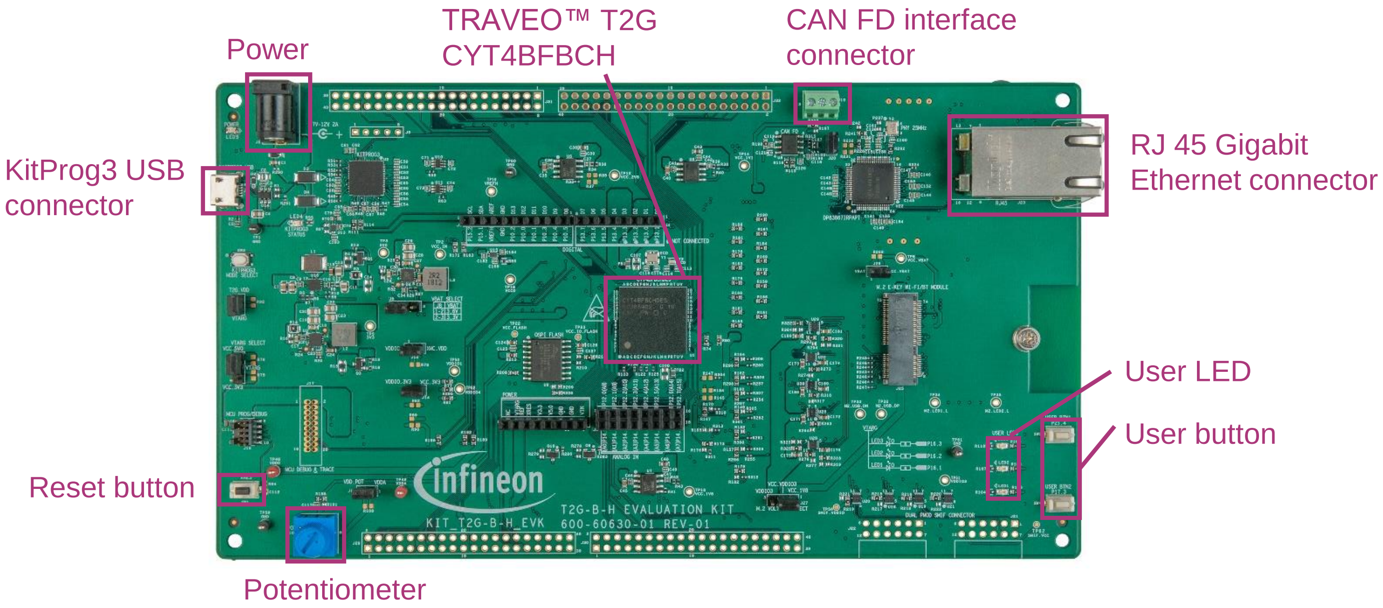
- Board
  - The TRAVEO^{TM} T2G KIT_T2G-B-H_EVK board is used for testing.

Introduction

- **Code Flash has the following features**
  - Supports optional memory size: 4 MB, 6 MB, and 8 MB
  - Programming and erasing functions
  - ECC function: 64b + 8b
  - Erase sector size of 32 KB for large sector and 8 KB for small sector
  - Program size: 64b, 256b, and 4096b
  - Supports Single Bank and Dual Bank modes
  - Supports reading while programming/erasing
  - Endurance of 1k
  - Retention of 20 years

Hardware setup

- This code example has been developed for the KIT-T2G-B-H-EVK board.
- Connect your PC to the board using the provided USB cable through the KitProg3 USB connector.



Implementation

- Flash memory provides nonvolatile storage for user firmware, user configuration data, and bulk data storage.
- This driver allows data to be read from and written to flash. It also provides the ability to obtain information about the address and characteristics of the flash block(s) contained on the device. During flash write time, the device should not be reset (including XRES pin, software reset, and watchdog), or unexpected changes may be made to portions of the flash. Also, the low-voltage detect circuits should be configured to generate an interrupt instead of a reset.
- In this example, the Flash HAL driver is configured to read sector data from the target sector, and programming these data to other empty sectors. Then, erase the target sector, readback source data from the empty sector, and process data (modify some data value), and program the processed data to the target sector.

Implementation (contd.)

Follow these steps to configure this code example:

- STDOUT setting
- Disabling data cache of the CPU
- Initialize code flash
- Access to the code flash

STDOUT setting

- Call the ***cy_retarget_io_init()*** function to use UART as STDOUT
  - Initialize P13.1 as UART TX, P13.0 as UART RX (these pins are connected to the KitProg3 COM port)
  - The serial port parameters change to 8N1 and 115200 baud

Disabling data cache of the CPU

- To ensure this code example's behavior is as intended, the data cache of the CPU is disabled by calling the ***SCB_DisableCache()*** function

Implementation (contd.)

Initialize code flash

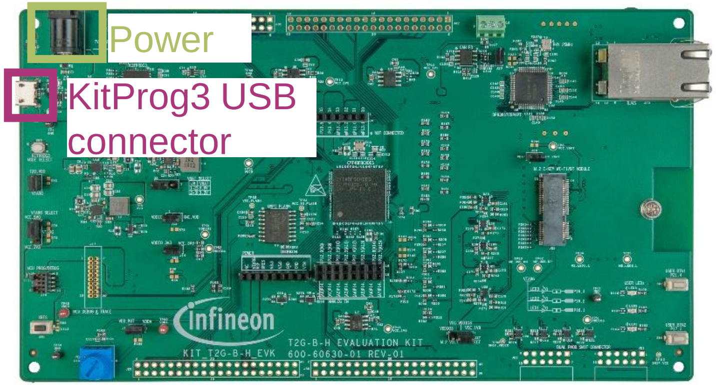
- First, the ***cyhal_flash_init()*** function initializes the code flash.
- Then, the ***Cy_Flashc_MainWriteEnable()*** function enables write operation of code flash.

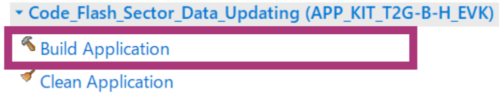
Access to the code flash

- Before writing, the code flash sector to be written should be erased by calling the ***cyhal_flash_erase()*** function
- Then, use the ***cyhal_flash_program()*** function for the write operation, giving the target address and the data to be written as argument.
- This code example uses ***cyhal_flash_read()*** to read out the data from code flash, but you can also use ***memcpy()*** to do so because the code flash is mapped in the address which CPU can access. The difference is only that the HAL function checks whether the specified address is included in code flash.

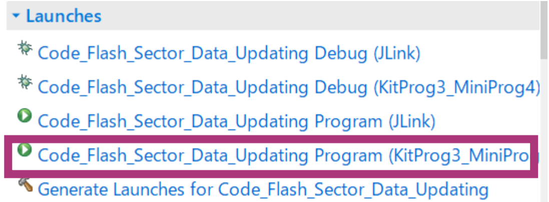
Compiling and programming

1. Connect to power and USB cable
2. Use Eclipse IDE for ModusToolbox™ software for compiling and programming
3. Compile
   a) Select the target application project in the Project Explorer.
   b) In the Quick Panel, scroll down, and click “Build Application” in Code_Flash_Sector_Data_Updating (KIT-T2G-B-H-EVK)



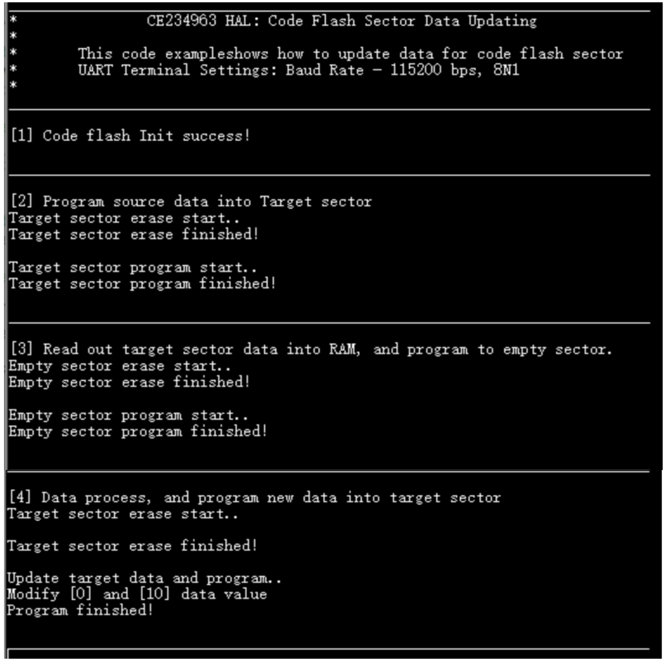


4. Open a terminal program and select the KitProg3 COM port. Set the serial port parameters to 8N1 and 115200 baud.
5. Programming
   a) Select the target application project in the Project Explorer
   b) In the Quick Panel, scroll down, and click
   “Code_Flash_Sector_Data_Updating Program (KitProg3_MiniProg4)” under Launches.


Launches
Code_Flash_Sector_Data_Updating Debug (JLink)
Code_Flash_Sector_Data_Updating Debug (KitProg3_MiniProg4)
Code_Flash_Sector_Data_Updating Program (JLink)
Code_Flash_Sector_Data_Updating Program (KitProg3_MiniPro
Generate Launches for Code_Flash_Sector_Data_Updating

Run and test

1. After successful programming, the application starts automatically. Confirm that "CE234963 HAL: Code Flash Sector Data Updating" is displayed on the UART terminal.
2. Observe the results in the terminal window as follows.



```
*               CE234963 HAL: Code Flash Sector Data Updating
*
*       This code exampleshows how to update data for code flash sector
*       UART Terminal Settings: Baud Rate - 115200 bps, 8N1
*

[1] Code flash Init success!


[2] Program source data into Target sector
Target sector erase start..
Target sector erase finished!

Target sector program start..
Target sector program finished!


[3] Read out target sector data into RAM, and program to empty sector.
Empty sector erase start..
Empty sector erase finished!

Empty sector program start..
Empty sector program finished!


[4] Data process, and program new data into target sector
Target sector erase start..

Target sector erase finished!

Update target data and program..
Modify [0] and [10] data value
Program finished!
```

References

Datasheet

- CYT4BF datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family

Architecture Technical reference manual

- TRAVEO™ T2G automotive body controller high family architecture technical reference manual

Registers Technical reference manual

- TRAVEO™ T2G Automotive body controller high registers technical reference manual

PDL/HAL

- PDL
- HAL

Training

- TRAVEO™ T2G Training

Revision History

Revision	ECN	Submission Date	Description of Change
**	7841075	2022/11/28	Initial release

Important notice and warnings

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

Edition 2022-11
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Go to:
www.infineon.com/support

Document reference
002-36104 **

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie") .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, 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.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

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.