

Customer training workshop: HAL_Watchdog_Timer for KIT_T2G-B-H_EVK

TRAVEO™ T2G CYT4BF series Microcontroller Training
V1.0.2 2023-02



Please read the [Important notice and warnings](#) at the end of this document

Scope of work

- › This example explains how to set up a watchdog timer (WDT). The WDT resets the device if it is not serviced or "kicked" within the configured timeout interval. This helps in recovering the program from an unintended lock-up.

By default, the WDT is reset at least once within each timeout interval to avoid a device reset. The user LED (P16.1) toggles every 1 second in the main loop to indicate that the CPU is in action. In addition, the user LED blinks once for power cycling or an external reset event.

Enable an infinite loop in the ***main()*** function to block the execution. The device resets in ~4 seconds. The user LED blinks twice after the device comes out of reset.

- › Device
 - The TRAVEO™ T2G CYT4BFBCH device is used in this code example.

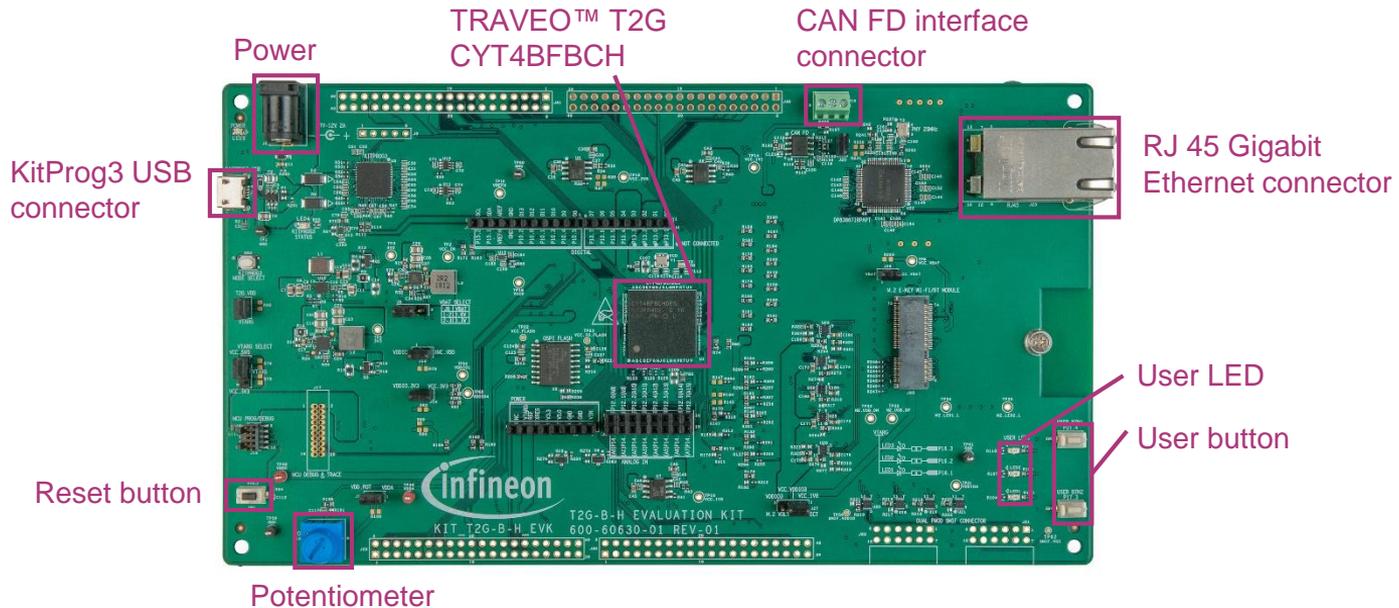
- › Board
 - The TRAVEO™ T2G KIT_T2G-B-H_EVK board is used for testing.

› **Watchdog timer has the following features:**

- One 32-bit free-running basic WDT with:
 - ILO0 as the input clock source
 - Programmable early threshold, warning threshold, and timeout threshold
 - Device reset generation if not serviced within a configurable interval
 - Warning threshold generates an interrupt to request servicing
 - Interrupt/wakeup generation in Active, Sleep, DeepSleep, and Hibernate power modes
 - Window mode
 - Running and freezing timers during DeepSleep mode
 - Debug

Hardware setup

- › This code example is 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

The WDT in the MCU is a 32-bit timer and uses the Internal Low-Speed Oscillator (ILO0) clock of 32 kHz. The WDT is configured using HAL APIs. These APIs configure the match count for the desired period.

Follow these steps to configure this code example:

- › User LED initialization
- › Indicating cause of reset using LED
- › Clear the reset cause registers
- › WDT initialization
- › User LED toggling

User LED initialization

- › The [*cyhal_gpio_init\(\)*](#) function initializes the user LED and the pin configuration once.
 - User LED is connected to P16.1 as output
 - If the device restarts by WDT reset, blink the LED twice
 - If the device restarts by power-on reset, blink the LED once

Implementation (contd.)

Indicating cause of reset using LED

- › The cause of latest reset, which we get by calling the [cyhal_system_get_reset_reason\(\)](#) function, is notified by the user LED.

Return value	Meaning	LED pattern
CYHAL_SYSTEM_RESET_WDT	Reset by WDT	Blinks twice
Other value	NOT reset by WDT	Blinks once

Clear the reset cause registers

- › The [cyhal_system_clear_reset_reason\(\)](#) function clears the reset cause registers.

WDT initialization

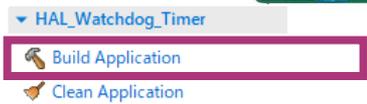
- › The [cyhal_wdt_init\(\)](#) function (called in **initialize_wdt()**) initializes the WDT once.
 - You can change the WDT timeout and WDT service by modifying the following parameter
 - **WDT_TIME_OUT_MS** (default="4000": 4000 ms)
 - **ENABLE_BLOCK_FUNCTION**
 - "0": WDT is serviced in the main loop (default)
 - "1": WDT is not serviced in the main loop, this will cause a WDT reset

User LED toggling

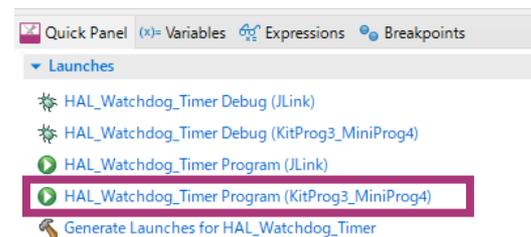
- › User LED toggles in the [cyhal_gpio_toggle\(\)](#) function per 1 second (if **ENABLE_BLOCK_FUNCTION** is "1")

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 HAL_Watchdog_Timer Application” in HAL_Watchdog_Timer (KIT-T2G-B-H-EVK)



4. Programming
 - a) Select the target application project in the Project Explorer
 - b) In the Quick Panel, scroll down, and click “HAL_Watchdog_Timer Program (KitProg3_MiniProg4)” under Launches

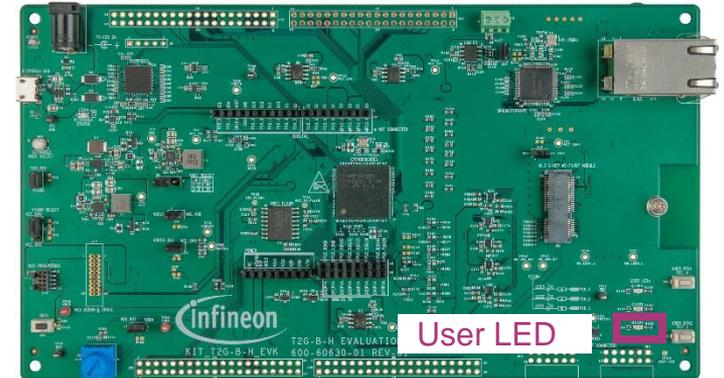


Run and test

1. Configure the blocking function with the macro `ENABLE_BLOCKING_FUNCTION` in the `main.c` as follows:
 - `#define ENABLE_BLOCKING_FUNCTION 0` or `1`
2. After compiling and programming, observe the status of the user LED (P16.1) based on different events summarized as follows:

Project setting	LED status
With the blocking function (<code>ENABLE_BLOCKING_FUNCTION = 1</code>)	After approximately 4 s, the device resets and the user LED blinks twice within a second to indicate a WDT reset
Without the blocking function (<code>ENABLE_BLOCKING_FUNCTION = 0</code>)	User LED toggles every 1 s to indicate that the CPU is in action

- **Note:** User LED blinks once on a power cycle or an external reset event



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
**	7781930	2022/06/27	Initial release
*A	7836293	2022/11/14	Added the <i>cyhal_system_clear_reset_reason ()</i> function.
*B	7876273	2023/02/20	Updated the title Updated figures in “Compiling and programming”

Important notice and warnings

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

Edition 2023-02

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2023 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Go to:

www.infineon.com/support

Document reference

002-35577 Rev. *B

IMPORTANT NOTICE

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

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.