



# HAL\_Watchdog\_Timer for KIT\_T2G-B-H\_LITE

Customer training workshop

Q3 2024



## Scope of work

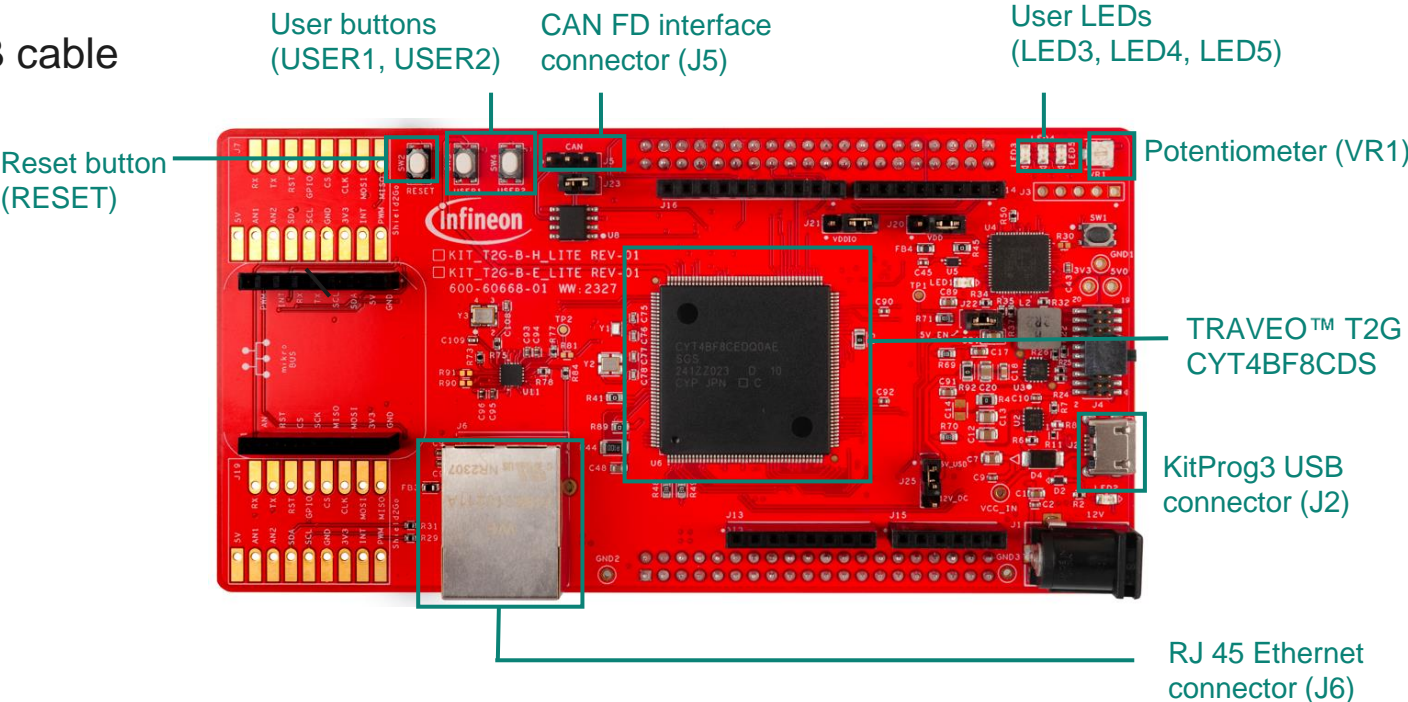
- This code example demonstrates how to set up a watchdog timer (WDT). The WDT resets the device if it is not serviced or "kicked" within the configured timeout interval. This helps in recovering the program from an unintended lock-up. By default, the WDT is reset at least once within each timeout interval to avoid a device reset. The user LED (LED3) 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 (LED3) blinks twice after the device comes out of reset.
- **Device**
  - The TRAVEO™ T2G CYT4BF8CDS device is used in this code example
- **Board**
  - The TRAVEO™ T2G KIT\_T2G-B-H\_LITE board is used for testing

# Introduction

- **Watchdog timer has the following features:**
  - One 32-bit free-running basic WDT:
    - 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, Deep Sleep, and Hibernate power modes
    - Window mode
    - Running and freezing timers during Deep Sleep mode
    - Debug

# Hardware setup

- This code example has been developed for the KIT\_T2G-B-H\_LITE board
- Connect the PC to the board using the provided USB cable through the KitProg3 USB connector (J2)



# 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
  - STDOUT Setting
  - Indicating cause of reset using the LED
  - Clear the reset cause registers
  - WDT initialization
  - User LED toggling
- **User LED initialization**
  - The [cyhal\\_gpio\\_init\(\)](#) function initializes the user LED (LED3) and the pin configuration once
    - If the device restarts by WDT reset, the LED blinks twice
    - If the device restarts by power-on reset, the LED blinks once
- **STDOUT setting**
  - The [cy\\_retarget\\_io\\_init\(\)](#) function initializes the GPIO for UART
    - Initializes P0.1 as UART TX, P0.0 as UART RX (these pins are connected to the KitProg3 COM port)
    - The serial port parameters change to 8N1 and 115200 baud

# Implementation

## – 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 (LED3)

## – 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

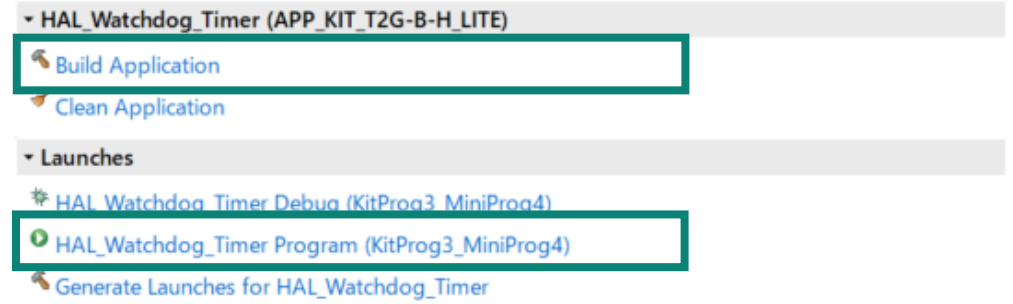
- User LED (LED3) toggles in the [cyhal\\_gpio\\_toggle\(\)](#) function per 1 second (if **ENABLE\_BLOCK\_FUNCTION** is '1')

# Compiling and programming

1. Connect to power and USB cable
2. Use Eclipse IDE for ModusToolbox™ software for compiling and programming
3. For compilation
  - a. Select the target application project in the Project Explorer
  - b. In the Quick Panel, scroll down, and click **Build Application** in HAL\_Watchdog\_Timer (KIT\_T2G-B-H\_LITE)
4. Open a terminal program (such as Tera Term) and select the KitProg3 COM port. Set the serial port parameters to **8N1** and **115200 baud**.
5. For 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)** in the Launches



KitProg3 USB connector (J2)



# Run and test

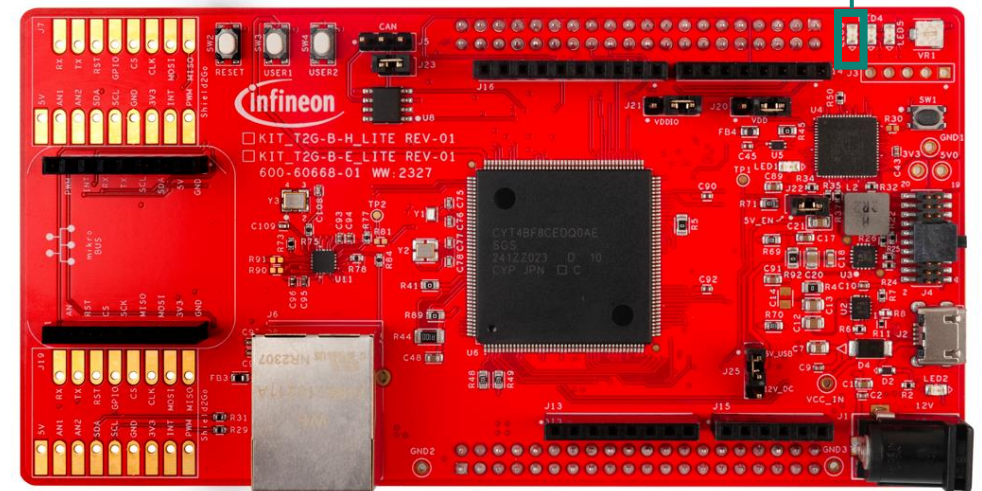
1. After programming, the application starts automatically.
2. Observe that the fault handling example title is displayed on the UART terminal as shown in the figure.
3. Configure the blocking function with the macro `ENABLE_BLOCKING_FUNCTION` in the `main.c` as follows:
  - `#define ENABLE_BLOCKING_FUNCTION 0` or `1`
4. After compiling and programming, the status of the user LED (LED3) is based on different events summarized in the table:

```
*****HAL: Watchdog Timer*****
Reset event from Power-On or XRES
█
```

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 (LED3) 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 (LED3) blinks once on a power cycle or an external reset event.

User LED (LED3)





# References

- **Datasheet**
  - [CYT4BF TRAVEO™ T2G 32-bit Automotive MCU based on Arm® Cortex®- M7 dual](#)
  
- **Architecture reference manual**
  - [TRAVEO™ T2G Automotive MCU body controller high architecture reference manual](#)
  
- **Registers reference manual**
  - [TRAVEO™ T2G Automotive MCU: TVII-B-H-8M body controller high registers reference manual](#)
  
- **PDL/HAL**
  - [Peripheral driver library \(PDL\)](#)
  - [Hardware abstraction layer \(HAL\)](#)
  
- **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 cyhal_system_clear_reset_reason () function
*B	7876273	2023/02/20	Updated the title Updated figures in “Compiling and programming”
*C	8085973	2024/11/20	Replaced development board from KIT_T2G-B-H_EVK to KIT_T2G-B-H_LITE

