



HAL_GPIO_Interrupt for KIT_T2G-B-H_LITE

Customer training workshop

Q3 2024



Scope of work

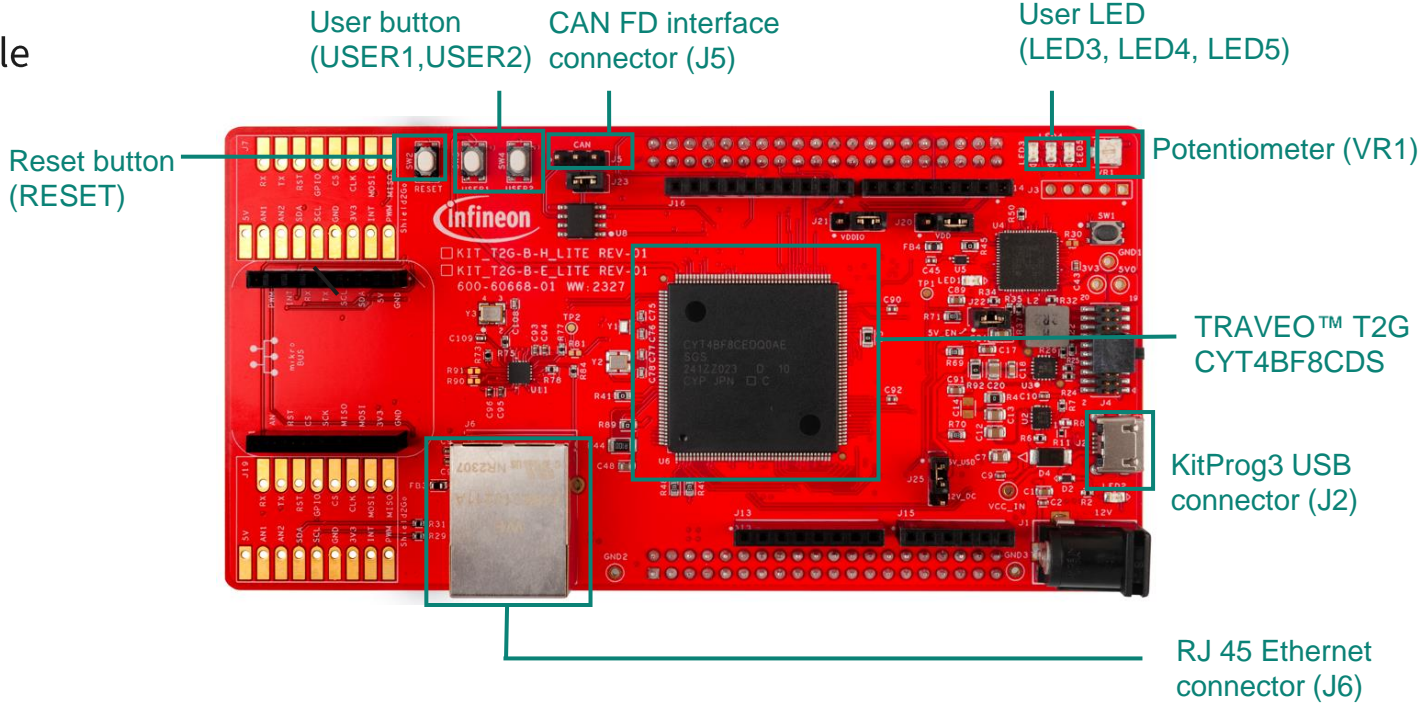
- This code example demonstrates the use of a GPIO configured as an input pin to generate interrupts on an Infineon MCU. The GPIO signal interrupts the CPU and executes a user-defined interrupt service routine (ISR). The GPIO interrupt acts as a wakeup source to wake the CPU from Deep Sleep.
- **Device**
 - 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

- **GPIO has the following features:**
 - Analog and digital input and output capabilities
 - Eight drive strength modes
 - Separate port Read and Write registers
 - Edge-triggered interrupts on rising edge, falling edge, or on both the edges, on all GPIOs
 - Slew rate control
 - Hold mode for latching the previous state (used to retain the I/O state in Deep Sleep mode)
 - Selectable CMOS, TTL, and Automotive input buffer mode
 - Smart I/O provides the ability to perform Boolean functions in the I/O signal path

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

- This code example uses GPIO interrupt to wake the Arm® Cortex®-M4 or Cortex®-M7 CPU from Deep Sleep. A LED is connected to an output pin; it is used for indicating the current state of the CPU. If the LED blinks, the CPU is in the Active state. After four successive blinks, the CPU enters the Deep Sleep. Since the GPIO state is retained during Deep Sleep, the LED stops blinking and turns OFF, to indicate that the CPU is in Deep Sleep.
- An input pin, externally connected to a switch, is configured to generate an interrupt when the switch turns ON. The interrupt triggers the following actions:
 - Generates a signal to wake the CPU from Deep Sleep
 - Executes an ISR: Once ISR is executed, a flag is updated, which is used to change the rate at which the LED blinks. With every press of the switch, the LED alternates the blinking interval.
- **The following steps are used to configure the code example:**
 - STDOUT setting
 - GPIO port pin initialization
 - Configure the interrupt
 - Configure the LED
 - Enters Deep Sleep

Implementation (contd.)

– 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.

– GPIO port pin initialization

- The [cyhal_gpio_init\(\)](#) function initializes the GPIO port pin once.
 - User LED (LED3) is connected to P5.0 as output and the user button (USER1) is connected to P5.3 as input.

– Configure the interrupt

- The [cyhal_gpio_register_callback\(\)](#) function registers the interrupt service routine (ISR).
 - The ISR name in this sample is *gpio_btn_callback_data*.
- The [cyhal_gpio_enable_event\(\)](#) function configures the interrupt.
 - Press the user button (USER1), an interrupt occurs, and the LED (LED3) starts to blink.

Implementation (contd.)

– Configure the LED

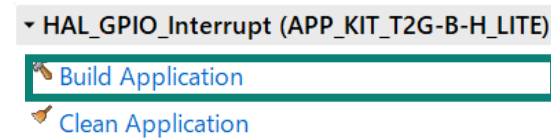
- If the GPIO interrupt is input by pressing the user button (USER1), the ISR is executed and the *gpio_intr_flag* is set
- When the flag is set, the main loop of the sample configures the user LED (LED3) by controlling the port output value to which it is connected
 - The [cyhal_gpio_write \(\)](#) function sets the output value of the GPIO pin
 - The [cyhal_system_delay_ms \(\)](#) function creates the blinking period of the user LED (LED3)
 - The blinking period of the user LED (LED3) changes by modifying these parameters:
 - *DELAY_SHORT_MS* (default = “250”: 2 Hz)
 - *DELAY_LONG_MS* (default = “500”: 1 Hz)

– Enter the Deep Sleep

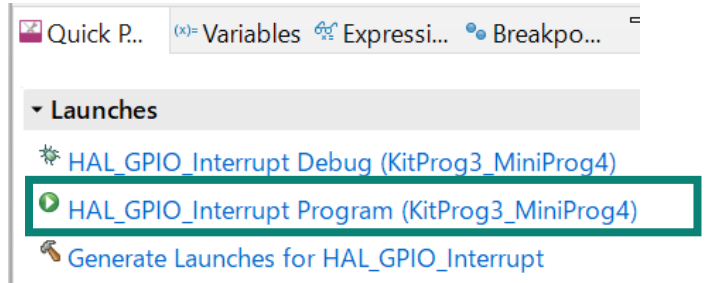
- After controlling the user LED (LED3), the main loop of the sample calls the [cyhal_syspm_deepsleep \(\)](#) function to set the CPU to Deep Sleep mode
- Once the user button (USER1) is pressed after the CPU enters Deep Sleep mode, the CPU wakes up from Deep Sleep and control the user LED (LED3) again

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_GPIO_Interrupt (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_GPIO_Interrupt Program (KitProg3_MiniProg4)”** in Launches

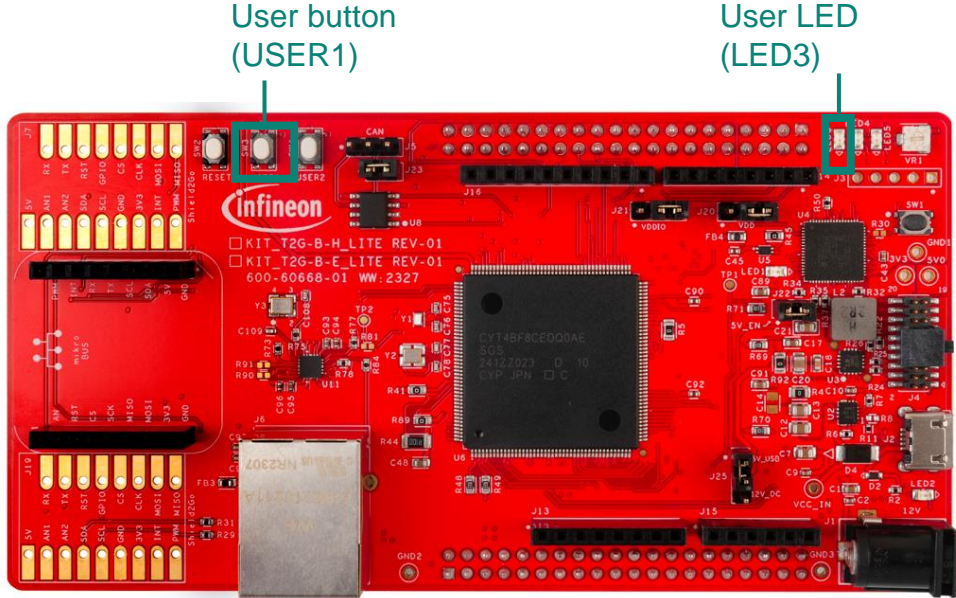
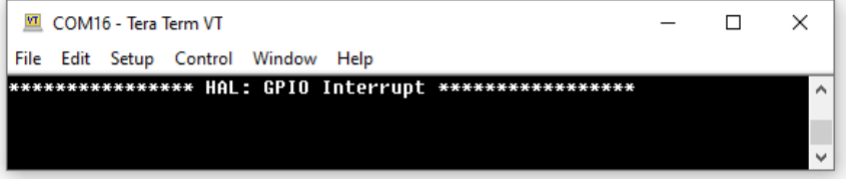


KitProg3 USB connector



Run and test

1. Once the programming is successfully complete, a message is displayed in the terminal window as shown in the figure.
2. Observe that the user LED (LED3) (P5.0) blinks four times and then turns OFF, indicating that the CPU has entered Deep Sleep.
3. Press the user button (USER1) (P5.3) to trigger an interrupt. This should wake up the device, causing the LED to resume blinking at a faster rate (default = 2 Hz). The LED blinks four times and the device enters Deep Sleep again.
4. Press the button again to repeat the wakeup cycle, and the LED resumes blinking at a slower rate. (default = 1 Hz)
5. With every interrupt and execution of ISR, the interval of blinking alternates between the slower and faster rates.



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
**	7782113	2022/07/05	Initial release
*A	7874757	2023/02/20	Updated the title Updated figures in “Compiling and programming”
*B	8065752	2024/08/21	Replaced development board from KIT_T2G-B-H_EVK to KIT_T2G-B-H_LITE

