

Customer training workshop: GPIO_Pins for KIT_T2G-B-H_EVK

TRAVEO™ T2G CYT4BF series Microcontroller Training
V1.0.1 2023-03



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

Scope of work

- › This example demonstrates the GPIO pin operation on the MCU, using Eclipse IDE for ModusToolbox™ software. This includes reading, writing, interrupts, and full configuration.

- › 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.

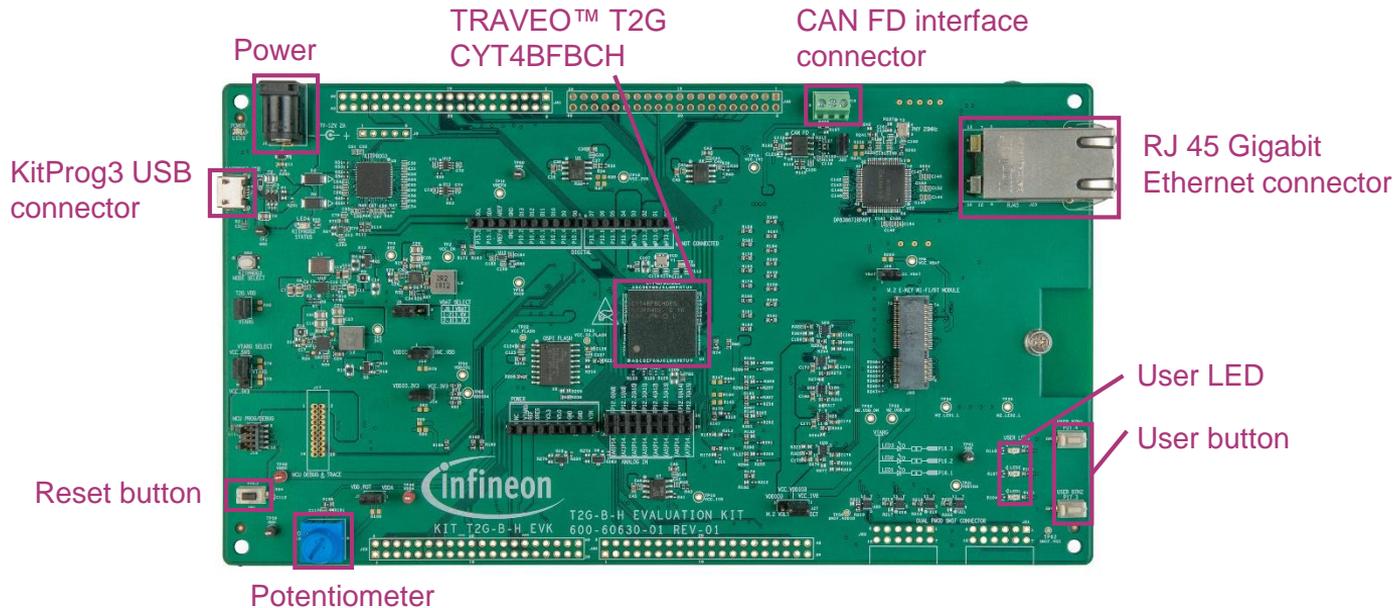
Introduction

› **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 previous state (used to retain the I/O state in DeepSleep mode)
- Selectable CMOS, TTL, and automotive input buffer modes
- 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-EVK board.
- › Connect your PC to the board using the provided USB cable through the KitProg3 USB connector.



Implementation

- › This example demonstrates the GPIO pin configuration, reading, writing, and interrupts using multiple GPIO PDL methods. This example shows various ways of using the GPIO pins to meet the needs of the project.
- › To demonstrate individual GPIO pin access, this example reads the value from the reference pin (user button) and writes it to the user LED. The user LED blinks twice to demonstrate various GPIO functions. The user button is configured to generate an interrupt on a falling edge, which occurs on a button release. The interrupt routine sets a flag to run the blinking sequence within the example loop.

Follow these steps to configure this code example:

- › GPIO port initialization
- › GPIO port pin initialization
- › GPIO port pin interrupt configuration
- › GPIO port pin read
- › GPIO interrupt detection
- › GPIO port pin write
- › GPIO port simultaneous access

Implementation (contd.)

GPIO port initialization

- › The [**Cy_GPIO_Port_Init\(\)**](#) function initializes the GPIO port.
 - Initialize all Port 5 pins and configure it with parameters in structure **port5_Init**
 - This is the most code-efficient method to configure all attributes for a full port of pins
 - The type of **port5_Init** is [**cy_stc_gpio_prt_config_t**](#), includes all configurations for each pins

GPIO port pin initialization

- › The [**Cy_GPIO_Pin_FastInit\(\)**](#) function initializes the GPIO port pin once.
 - The user LED is connected to P16.1 as output and the user button is connected to P21.4 as input.
 - It supports only parameterized configuration of the drive mode, output logic level, and high-speed input/output multiplexer (HSIOM) setting.
 - A method to configure all attributes of a single pin is to use the [**Cy_GPIO_Pin_Init\(\)**](#) function and a pin configuration structure. While easy to use, it generates a larger code than other configuration methods.

Implementation (contd.)

GPIO port pin interrupt configuration

- › The [**Cy_GPIO_SetInterruptEdge\(\)**](#) function configures the interrupt condition of the GPIO pin.
 - When a rising edge is detected, an interrupt occurs.
- › The [**Cy_GPIO_SetInterruptMask\(\)**](#) function configures the interrupt enable of the GPIO pin.
 - Enable interrupt from the user button (P21.4)
- › Configure interrupt in the [**Cy_SysInt_Init\(\)**](#) function
 - Set the interrupt source (Port21), interrupt priority (7), interrupt vector, and ISR (***gpio_interrupt_handler_PDL()***)
 - Then, clear the IRQ request of the configured interrupt by ***NVIC_ClearPendingIRQ()***¹, before enabling IRQ by ***NVIC_EnableIRQ()***¹
- › The [**Cy_GPIO_ClearInterrupt\(\)**](#) function clears the interrupt flag.
 - Clear the GPIO port interrupt flag

¹: The CPU interrupt enable and NVIC operation instructions are provided by Cortex microcontroller software interface standard (CMSIS) with intrinsic functions.

Implementation (contd.)

GPIO port pin read

- › In the main loop, value of the port pin is read by the [Cy_GPIO_Read\(\)](#) function
 - Read the user button (P21.4) condition

GPIO interrupt detection

- › A push of the user button is detected as the ***gpio_interrupt_handler_PDL()*** interrupt is called as ISR.
 - Sets ***gpio_intr_flag***
 - Clears the interrupt by [Cy_GPIO_ClearInterrupt\(\)](#)

Implementation (contd.)

GPIO port pin write

- › In the main loop, various methods are used for output to user LED (P16.1) when the ***gpio_intr_flag*** is set
 - [Cy_GPIO_Write\(\)](#) can specify output directly
 - [Cy_GPIO_Inv\(\)](#) can invert current pin state
 - [Cy_GPIO_Clr\(\)](#) can set the pin state to low; conversely, [Cy_GPIO_Set\(\)](#) can set the pin state to high.
- › In this example, these functions are called to blink the user LED, the blinking intervals are generated by [Cy_SysLib_Delay\(\)](#).
 - Value is specified in milliseconds
 - You can change the period of the user LED state by modifying these parameters (default="500": 1Hz).

GPIO port simultaneous access

- › All the pins included in a port can be accessed simultaneously by direct register access.
 - This feature enables efficient access if multiple pins need to be accessed simultaneously.
 - May not be thread or multi-core safe due to possible read-modify-write operations.
 - A single CPU core is needed to use this feature for a particular port.

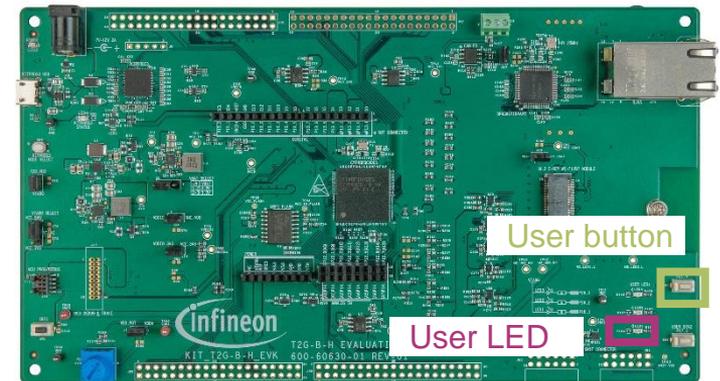
Compiling and programming

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 GPIO Pins (APP_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 “GPIO_Pins Program (KitProg3_MiniProg4)” under Launches



Run and test

1. After successful programming, press the user button (P21.4), and observe that the user LED (P16.1) turns ON demonstrating the GPIO read and write function.
2. Release the user button, observe that the user LED turns OFF and then the user LED blinks twice demonstrating the pin interrupt functionality



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
**	7782141	2022/07/05	Initial release
*A	7876713	2023/03/01	Changed cy_st_gpio_prt_config_t to cy_stc_gpio_prt_config_t in "Implementation" Changed 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 2023-03

**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
002-35579 Rev. *A**

IMPORTANT NOTICE

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

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.