GPIO_Pins for KIT_T2G-B-H_LITE

Customer training workshop

Q3 2024

Scope of work

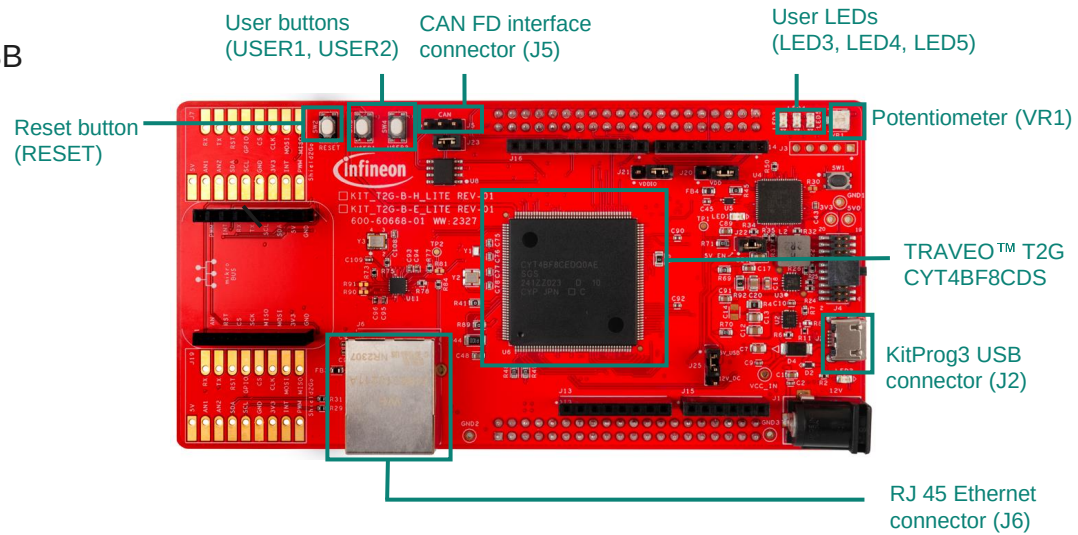
- This code example demonstrates the GPIO pin operation on the MCU, using Eclipse IDE for ModusToolbox™ software. This includes reading, writing, interrupts, and full configuration.
- **Device**
  - The TRAVEO™ T2G CYT4BF8CDS device is used in this code example
- **Board**
  - The TRAVEO™ T2G KIT_T2G-B-H_LITE board is used for testing

Introduction

- **GPIO features**
  - Analog and digital input and output capabilities
  - Eight drive strength modes
  - Separate port Read and Write registers
  - Edge-triggered interrupts on the rising edge, falling edge, or on both 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 the 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 is 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 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 requirements of the project.
- To demonstrate the individual GPIO pin access, this example reads the value from the reference pin- user button (USER1) and writes it to the user LED (LED3). The user LED (LED3) blinks twice to demonstrate various GPIO functions. The user button (USER1) 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.

- **The following steps are used to configure the 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 ***port13_Init***
    - This is the most code-efficient method to configure all attributes for a full port of pins
    - The type of ***port13_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 (LED3) is connected to P5.0 as output and the user button (USER1) is connected to P5.3 as an input
  - It supports only parameterized configuration of the Drive mode, output logic level, and high-speed input/output multiplexer (HSIOM) setting
  - This method configures all attributes of a single pin and uses the ***Cy_GPIO_Pin_Init()*** function and a pin configuration structure. It is easy to use and 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 (P5.3)
- Configure interrupt in the ***Cy_SysInt_Init()*** function
  - Set the interrupt source (Port5), interrupt priority (7), interrupt vector, and ISR (***gpio_interrupt_handler_PDL()***)
  - Then, clear the IRQ request of the configured interrupt by ***NVIC_ClearPendingIRQ()***, before enabling IRQ by ***NVIC_EnableIRQ()***
- The ***Cy_GPIO_ClearInterrupt()*** function clears the interrupt flag
  - Clear the GPIO port interrupt flag

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 (P5.3) condition

GPIO interrupt detection

- A push of the user button is detected as the ***gpio_interrupt_handler_PDL()*** interrupt is called as ISR (Interrupt Service Routine)
  - Sets ***gpio_intr_flag***
  - Clears the interrupt by ***Cy_GPIO_ClearInterrupt()***

Implementation (contd.)

GPIO port pin write

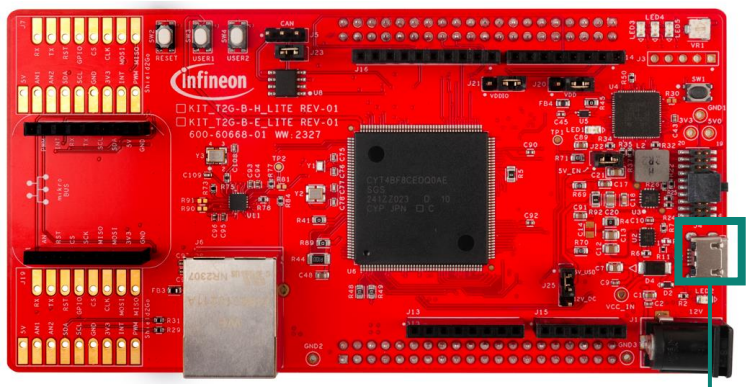
- In the main loop, various methods are used for output to user LED (LED3) when the ***gpio_intr_flag*** is set
  - ***Cy_GPIO_Write()*** can specify output directly
  - ***Cy_GPIO_Inv()*** can invert the current pin state
  - ***Cy_GPIO_Clr()*** can set the pin state to LOW; conversely, ***Cy_GPIO_Set()*** is set the pin state to HIGH
- In this example, these functions are called to blink the user LED (LED3), the blinking intervals are generated by ***Cy_SysLib_Delay()***
  - Value is specified in milliseconds (ms)
  - The period of the user LED (LED3) is changed by modifying these parameters (default= 500: 1 Hz)

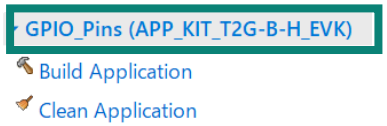
GPIO port simultaneous access

- 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

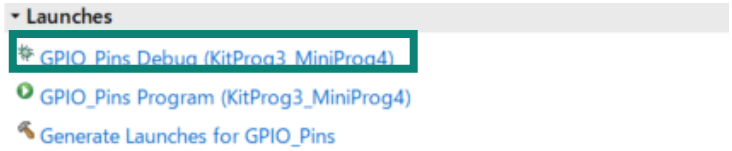
Compiling and programming

1. Connect to power and USB cable.
2. Use Eclipse IDE for ModusToolbox™ software for compiling and programming.
3. For compilation:
   a. Select the target application project in the Project Explorer.
   b. In the Quick Panel, scroll down, and click **Build Application** in GPIO pins (KIT_T2G-B-H_LITE).
4. For 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 the Launches.





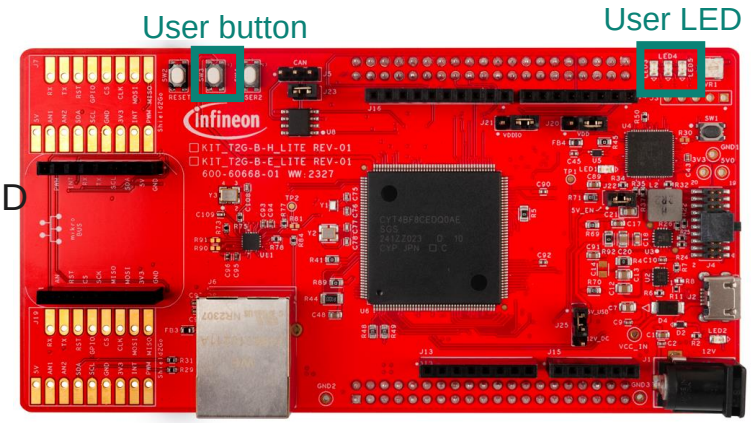
KitProg3 USB connector



Run and test

1. After successful programming, press the user button1 (P5.3) and observe that the user LED3 (P5.0) turns ON demonstrating the GPIO read and write function.

2. Release the user button(USER1), observe that the user LED (LED3) turns OFF and then the user LED blinks twice demonstrating the pin interrupt functionality.



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
**	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”
*B	8088172	2024/11/13	Replaced development board from KIT_T2G-B-H_EVK to KIT_T2G-B-H_LITE