HAL_I2C_Master for KIT_T2G-B-H_LITE

Customer training workshop

Q3 2024

Scope of work

- This code example demonstrates the use of the I2C (HAL) resource in master mode. The I2C master is configured to send command packets to control a user LED on the slave. Both the slave and master can be configured on a single kit.

- **Device**
  - The TRAVEO™ T2G CYT4BF8CDS device is used in this code example.

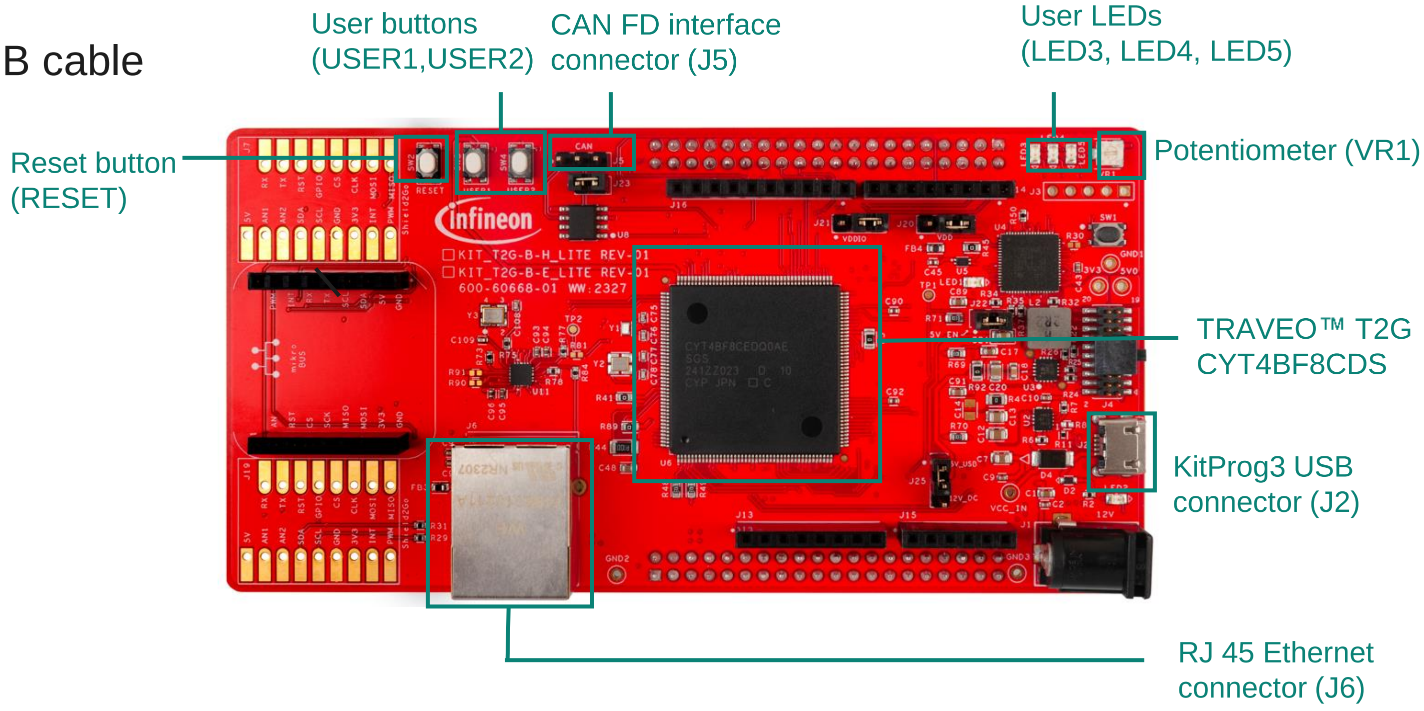
- **Board**
  - The TRAVEO™ T2G KIT_T2G-B-H_LITE board is used for testing.

Introduction

- **I2C has the following features:**
  - Master, slave, and master/slave mode
  - Standard-mode (100 kbps), fast-mode (400 kbps), and fast-mode plus (1000 kbps) data-rates
  - 7-bit slave addressing
  - Clock stretching
  - Collision detection
  - Programmable oversampling of I2C clock signal (SCL)
  - Auto ACK when RX FIFO not full, including address
  - General address detection
  - FIFO Mode
  - EZ and CMD_RESP modes
  - Interrupts or polling CPU interface
  - Analog glitch filter
  - Local loop-back control

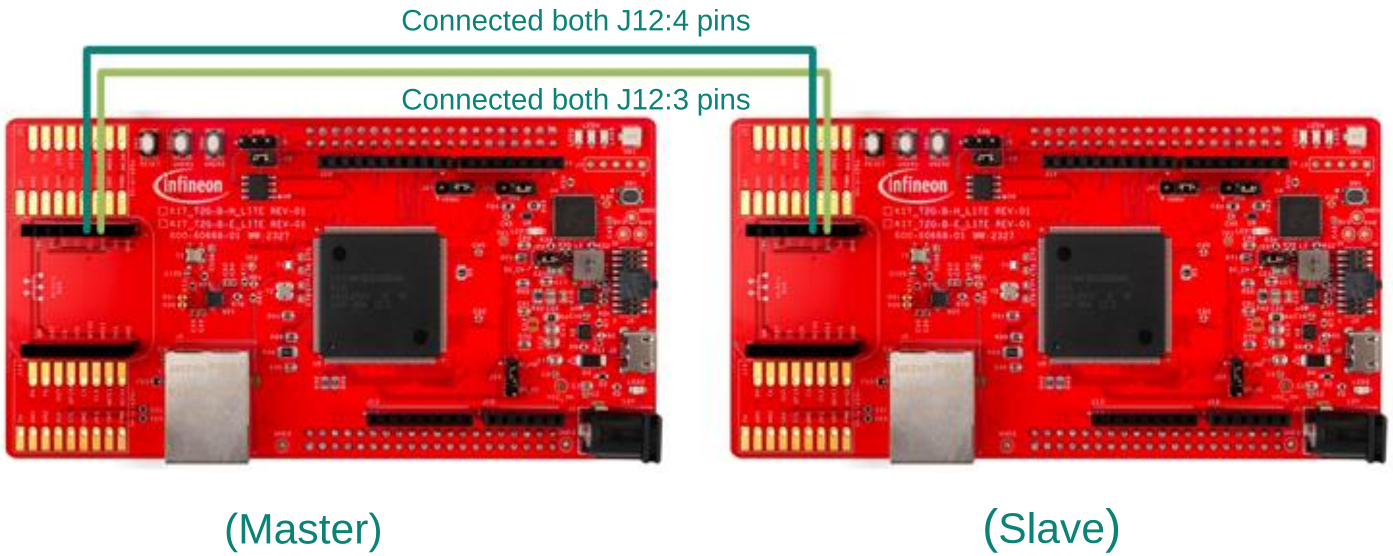
Hardware setup

- 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)


User buttons (USER1,USER2)
CAN FD interface connector (J5)
User LEDs (LED3, LED4, LED5)
Reset button (RESET)
Potentiometer (VR1)
TRAVEO™ T2G CYT4BF8CDS
KitProg3 USB connector (J2)
RJ 45 Ethernet connector (J6)

Hardware setup (contd.)

- Use jumper wires to establish a connection between this kit (as the master) and other the kit (as the slave). In this case, load the program for I2C-Slave in flash memory of the slave device. (See the Customer training workshop: **HAL_I2C_Slave for KIT_T2G-B-H_LITE**)
  - I2C_SDA (P6.1): J12:3 pin
  - I2C_SCL (P6.2): J12:4 pin



- By default, the code example is configured to work in the ‘master only' mode. In the *resource_map.h* file, it can change the value of the ***I2C_MODE*** macro to ***I2C_MODE_BOTH***.

Implementation

- This code example demonstrates the use of the I2C (HAL) resource in the master mode. Both the slave and master can be configured on a kit. After the slave and master are initialized, the send/read starts. The master is configured to send command packets to control a user LED (LED3) on the slave.

- **Follow these steps to configure this code example:**
  - STDOUT setting
  - GPIO port pin initialization[1]
  - I2C slave initialization[1]
  - I2C master initialization
  - Send command packet to the slave
  - Read the response from the slave

[1] Only execute when ***I2C_MODE_BOTH*** or ***I2C_MODE_SLAVE*** is defined.

- **STDOUT setting**
  - Call the ***cy_retarget_io_init()*** function to use UART as STDOUT.
    - Initialize P0.1 as UART TX, P0.0 as UART RX (these pins are connected to KitProg3 COM port)
    - The serial port parameters change to 8N1 and 115200 baud

Implementation (contd.)

- **GPIO port pin initialization**
  - The ***cyhal_gpio_init()*** function initializes the GPIO port pin once
    - Initialize P5.0 as output (initial level = H, LED turns off)
    - The serial port parameters change to 8N1 and 115200 baud

- **I2C slave initialization**
  - The ***cyhal_i2c_init()*** function initializes the I2C peripheral once (This can only be done when ***I2C_MODE_BOTH*** or ***I2C_MODE_SLAVE*** is defined)
    - Initializes an I2C resource as a slave and selects pins for SDA and SCL
    - Configures the I2C block by ***cyhal_i2c_configure()*** and sets it as slave
    - Configures the read buffer by the ***cyhal_i2c_slave_config_read_buffer().*** Next, configure the write buffer by ***cyhal_i2c_slave_config_write_buffer()***
    - Registers the callback function by ***cyhal_i2c_register_callback().*** The function is called when one of the events that is configured by ***cyhal_i2c_enable_event()*** (*CYHAL_I2C_SLAVE_WR_CMPLT_EVENT* or *CYHAL_I2C_SLAVE_RD_CMPLT_EVENT* or *CYHAL_I2C_SLAVE_ERR_EVENT*) occur

Implementation (contd.)

- **I2C master initialization**
  - The ***cyhal_i2c_init()*** function initializes the I2C peripheral once
    - Initializes an I2C resource as a master and selects pins for SDA and SCL
    - Configures the I2C block by ***cyhal_i2c_configure()*** and sets it as master

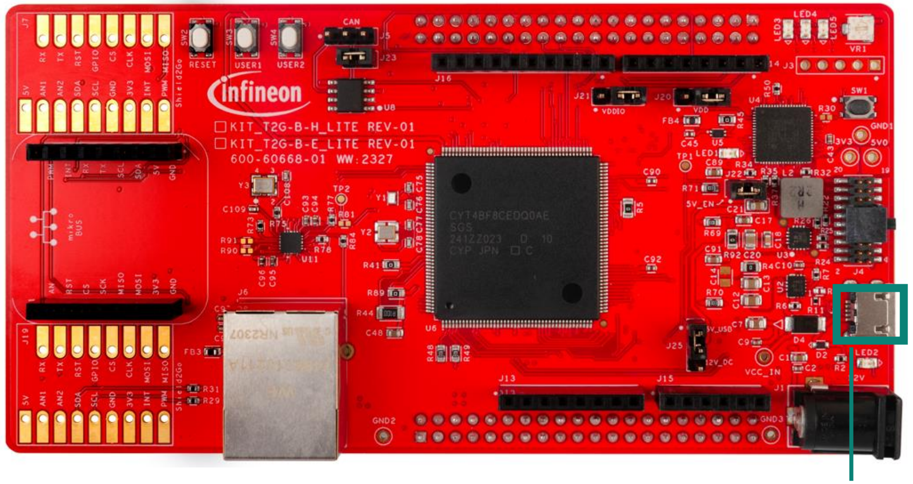
- **Send the command packet to the slave**
  - I2C master sends the command packet to the slave by ***cyhal_i2c_master_write()***
  - In the I2C slave, the ***handle_slave_event*** callback function is called once the packet is received
    - LED (LED3) blinks according to the received packet by calling ***cyhal_gpio_write()***, and then updates the read buffer as ***STS_CMD_DONE.*** Configures the write buffer for the next request by calling ***cyhal_i2c_slave_config_write_buffer().***

Implementation (contd.)

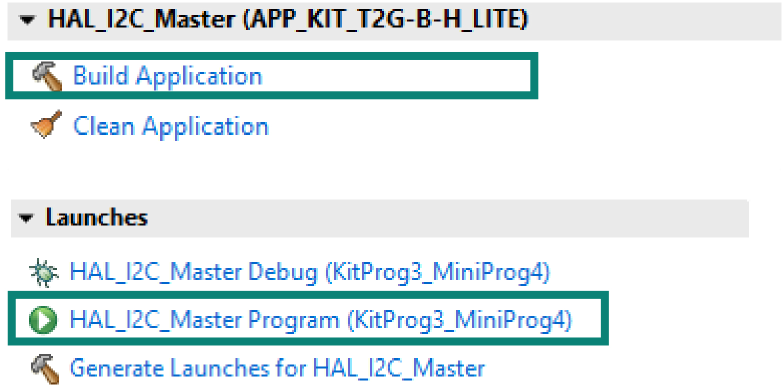
- **Read the response from the slave**
  - I2C master reads the response packet to generate the next command
    - After the I2C master sends the command packet to the slave successfully, it will read the response from the slave by ***cyhal_i2c_master_read()***
    - After the I2C master reads the command packet from the slave successfully, it will generate the next command

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_I2C_Master (KIT_T2G-B-H_LITE)
4. Open a terminal program 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_I2C_Master Program (KitProg3_MiniProg4)” in Launches



KitProg3 USB connector



▾ HAL_I2C_Master (APP_KIT_T2G-B-H_LITE)

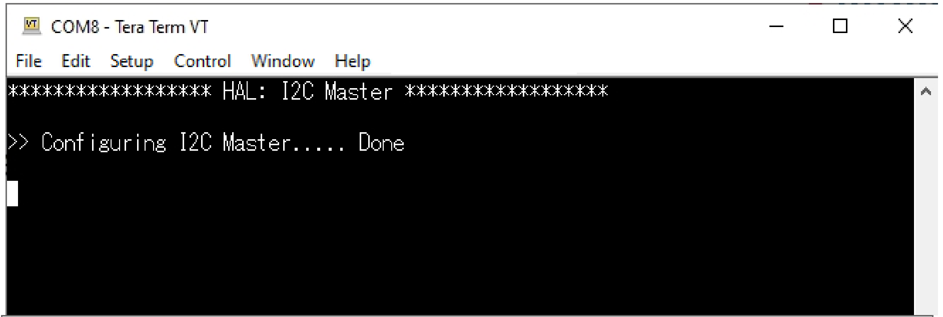
- Build Application
- Clean Application

▾ Launches

- HAL_I2C_Master Debug (KitProg3_MiniProg4)
- HAL_I2C_Master Program (KitProg3_MiniProg4)
- Generate Launches for HAL_I2C_Master

Run and test

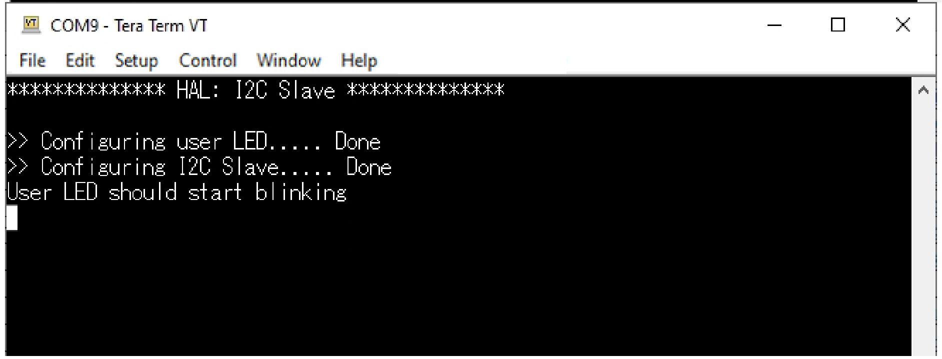
1. After successful programming, the application starts automatically. Observe that the UART terminal displays a message as shown in the figure.
2. Observe that the kit user LED1 (Slave kit) blinks at 1 Hz when I2C communication starts.



```
***************** HAL: I2C Master *****************

>> Configuring I2C Master..... Done
```

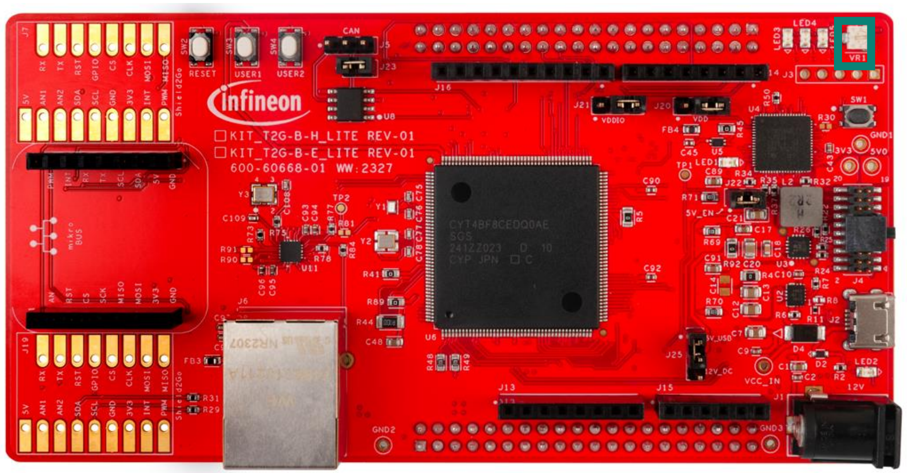
(Master)



```
************* HAL: I2C Slave *************

>> Configuring user LED..... Done
>> Configuring I2C Slave..... Done
User LED should start blinking
```

(Slave)



User LED1
(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
**	7782502	2022/07/06	Initial release
*A	7836538	2022/11/15	Added comments on page 6 and page 7
*B	7876266	2023/02/20	Updated the title Updated figures in “Compiling and programming”
*C	8080810	2024/10/03	Replaced development board from KIT_T2G-B-H_EVK to KIT_T2G-B-H_LITE