I2C_Master_EZI2C_Slave 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 with an EZI2C slave. The I2C master is configured to send command packets to control a user LED on the slave. Both the slave and the master can be configured on the same kit.

- This code example uses two LITE boards for master and slave mode.

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

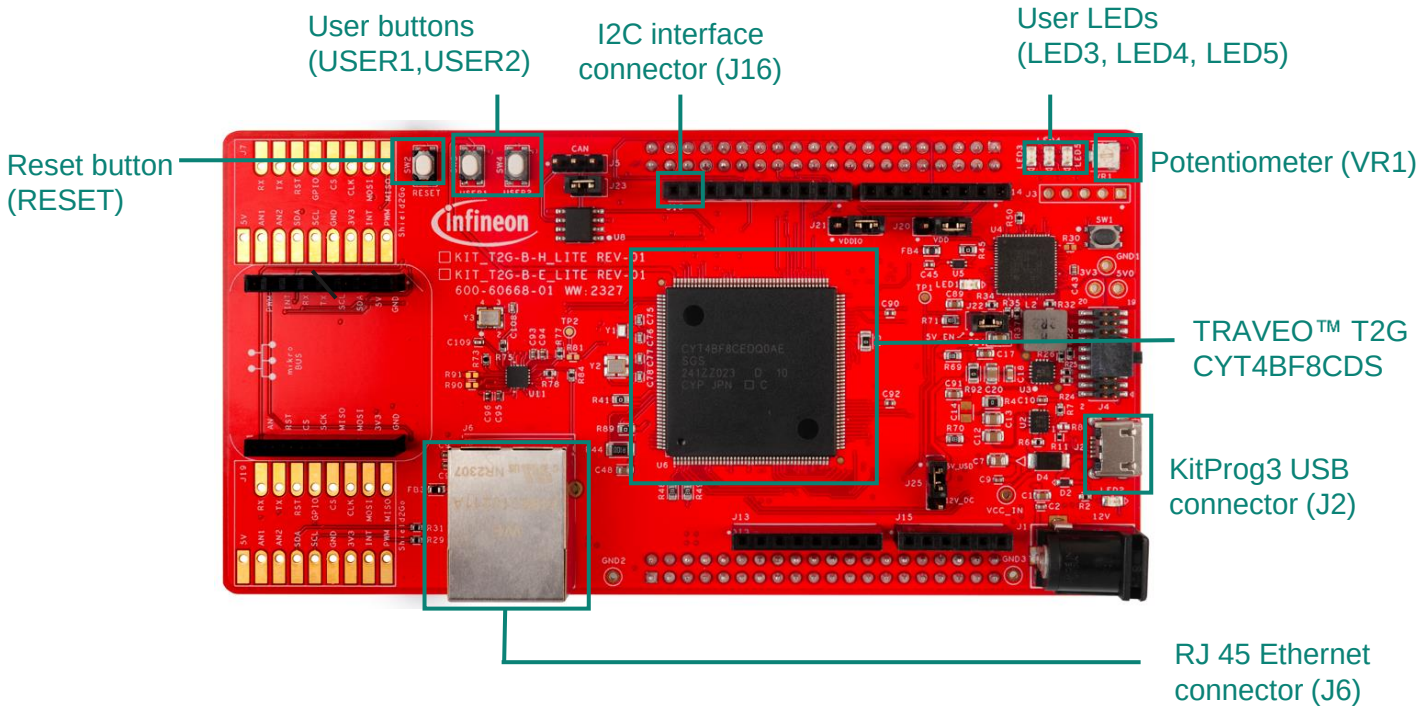
- **Board**
  - The TRAVEO™ T2G KIT_T2G-B-H_LITE board is used for testing

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 the I2C clock signal (SCL)
  - Auto ACK when RX FIFO is 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

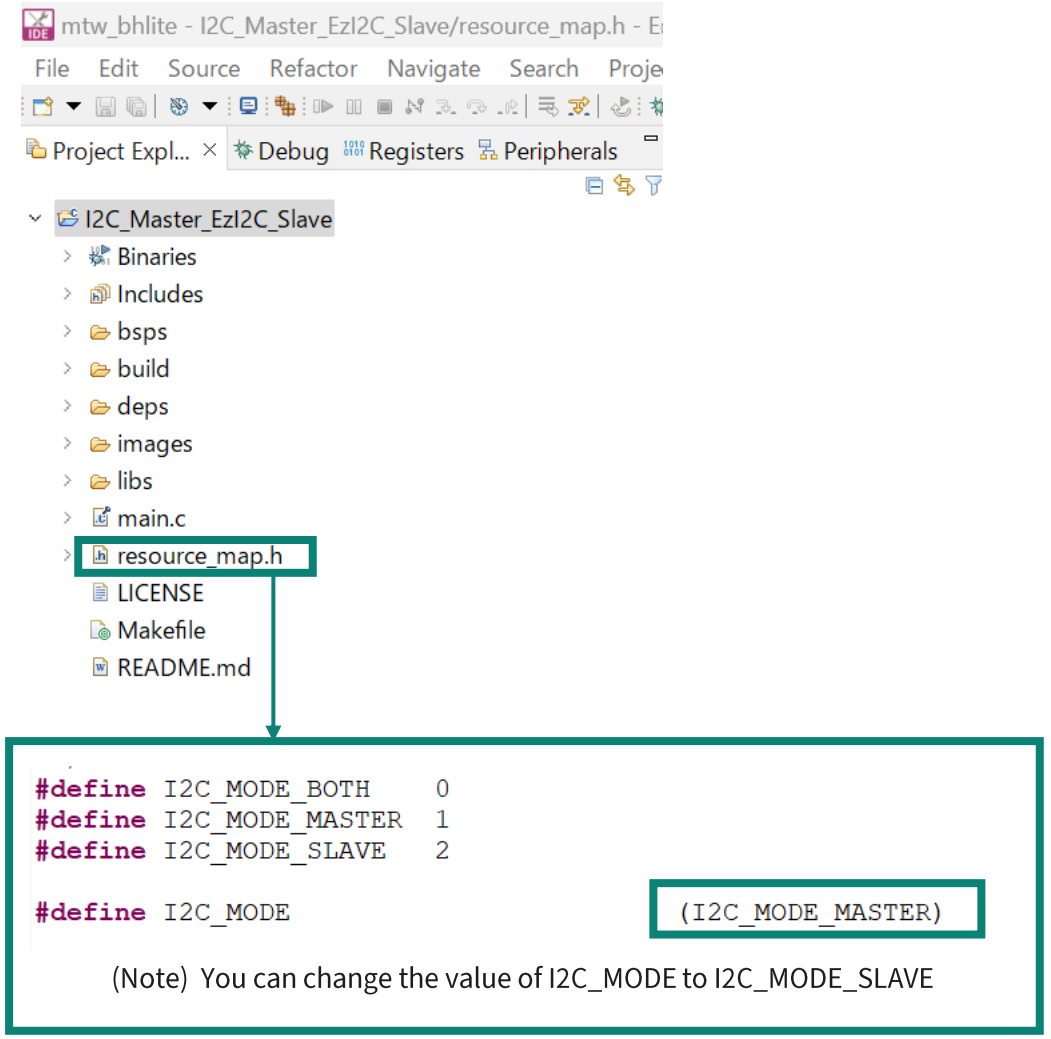
- This code example has been developed for the KIT_T2G-B-H_LITE board
- Connect the PC to the board using the provided USB cable through the KitProg3 USB connector (J2)



Implementation

- This code example demonstrates the use of the I2C (HAL) resource in master mode with an EZI2C slave. After I2C master initialization is complete, it will start to send the command packet to the slave.
- The command packets are used to control a user LED (LED3) on the slave.
- The I2C master reads the response from the slave and generates the next command.
- The default settings in this code example are configured to operate in master mode. To set it to slave mode, change the value of I2C_MODE to I2C_MODE_SLAVE in the resource_map.h file.

- **The following steps are used to configure the code example:**
  - STDOUT setting
  - GPIO port pin initialization
  - EZI2C slave initialization
  - I2C master initialization
  - Send command packet to the slave
  - Read the response from the slave



Implementation (contd.)

- **STDOUT setting**
  - 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 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

- **EZI2C slave initialization:**
  - The ***cyhal_ezi2c_init()*** function initializes the I2C peripheral
    (This can only be done when ***I2C_MODE_SLAVE*** is defined)
    - Initializes an I2C resource as a slave and selects pins for SDA and SCL, clock = 400 KHz
    - Register callback function using ***cyhal_ezi2c_register_callback()***; the function will be called when one of the events that configured using the ***cyhal_ezi2c_enable_event()*** (***CYHAL_EZI2C_STATUS_ERR*** or ***CYHAL_EZI2C_STATUS_WRITE1*** or ***CYHAL_EZI2C_STATUS_READ1***) occurs

Implementation (contd.)

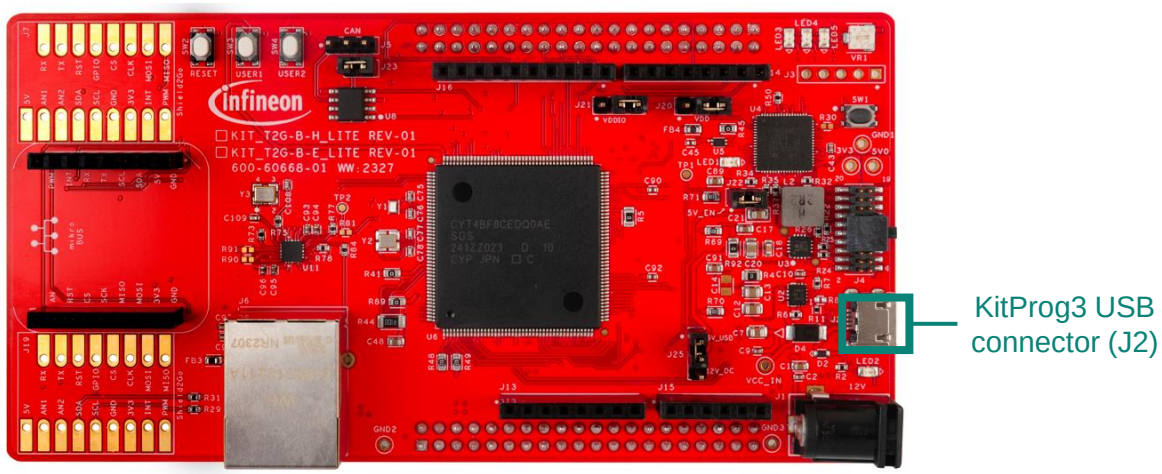
- **I2C master initialization:**
  - The ***cyhal_i2c_init()*** function initializes the I2C peripheral
    - Initializes an I2C resource as a master and assigns the SDA and SCL pins
    - The ***cyhal_i2c_configure()*** function configures the I2C block to set it as master

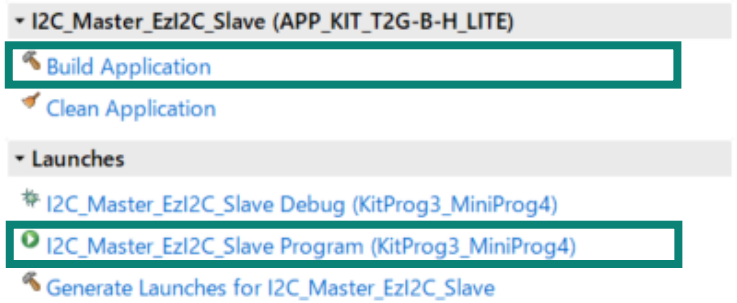
- **Send command packet to the slave:**
  - I2C master sends command packet to the slave using the ***cyhal_i2c_master_write()*** function

- **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 using the ***cyhal_i2c_master_read()*** function
    - After the I2C master reads the command packet from the slave successfully, it will generate the next command after 1 second delay using the ***cyhal_system_delay_ms()*** function

Compiling and programming

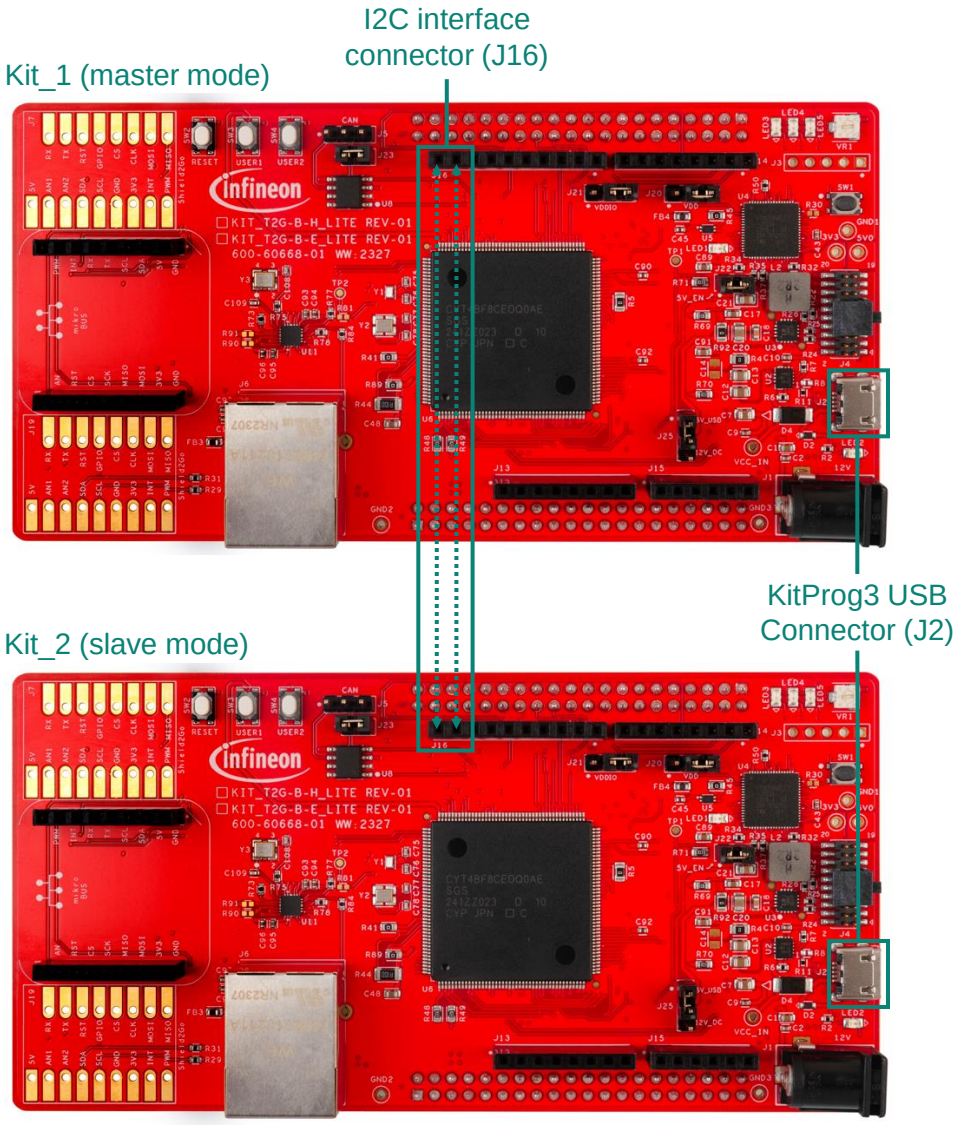
1. Connect to power and USB cable
2. Use Eclipse IDE for ModusToolbox™ software for compiling and programming
3. For compilation:
   a. Select the target application project in the Project Explorer
   b. In the Quick Panel, scroll down, and click **Build Application** in I2C Master EzI2C Slave (KIT_T2G-B-H_LITE)
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 **I2C_Master_EzI2C_Slave Program (KitProg3_MiniProg4)** in the Launches





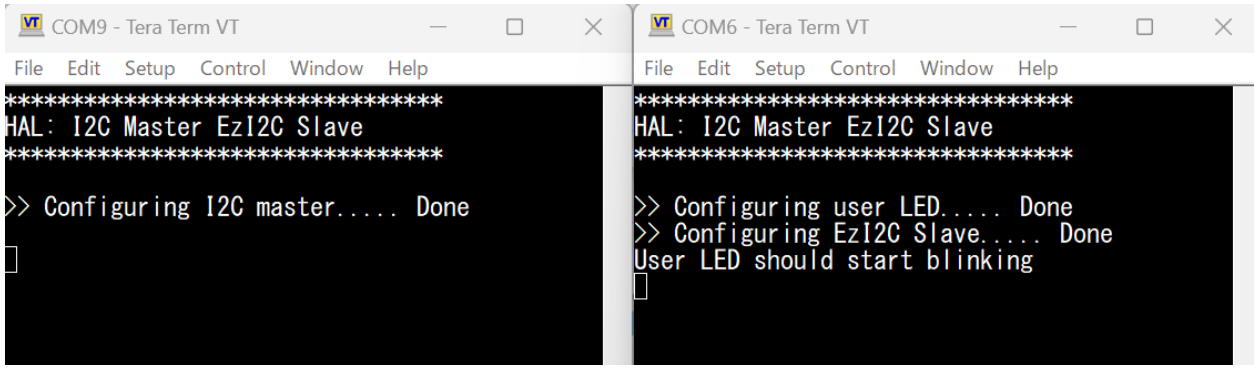
Run and test

1. Connect two kits as shown in the following screenshots and connect two kits to your PC using an USB cable.
   Note: See the dotted line in the figures.
   a. Connect I2C_SDA (J16-9) pin of kit_1 and kit_2 using jumper wires
   b. Connect I2C_SCL (J16-10) pin of kit_1 and kit_2 using jumper wires



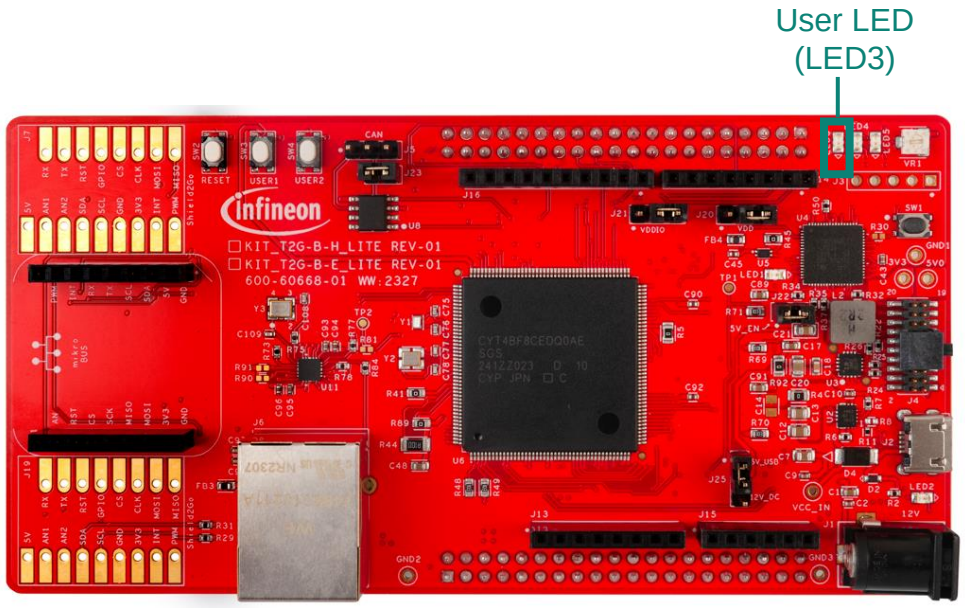
Run and test

2. After successful programming, the application starts automatically.
3. A message is displayed in the UART terminal window as shown in the figure.
4. Observe that the user LED (LED3) in the Kit_2 (Slave mode) blinks at 1 Hz.



Kit_1 (master mode)

Kit_2 (slave mode)



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
**	7782644	2022/07/06	Initial release
*A	7837066	2022/11/08	Added comments on page 6 and page 7
*B	8086122	2024/11/01	Replaced development board from KIT_T2G-B-H_EVK to KIT_T2G-B-H_LITE