# Customer training workshop: I2C_Master_EZI2C_Slave for KIT_T2G-B-H_EVK

TRAVEO™ T2G CYT4BF series Microcontroller Training
V1.0.1 2022-11

# 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.

- 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.

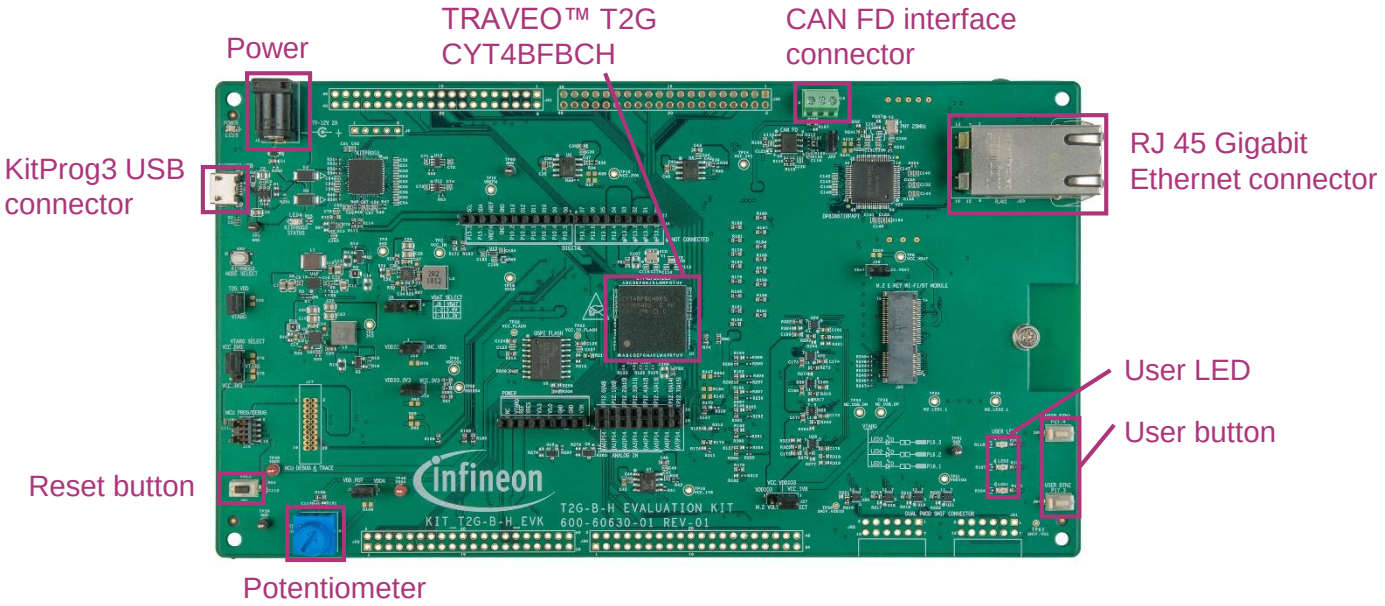
# Introduction

- **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-EVK board.

› Connect your PC to the board using the provided USB cable through the KitProg3 USB connector.

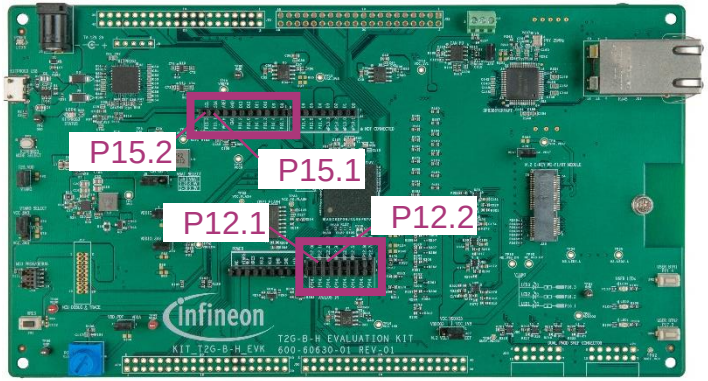


# Hardware setup (contd.)

› This code example is configured to work in the “Master only” mode.
Therefore, use jumper wires to establish a connection between the master and the slave on the kit.

– Connect ***mI2C_SCL*** (master) to ***sI2C_SCL*** (slave)
– Connect ***mI2C_SDA*** (master) to ***sI2C_SDA*** (slave)

| Code setting (*I2C_MODE*) | Master side | | Slave side | |
|---|---|---|---|---|
| | *mI2C_SDA* | *mI2C_SCL* | *sI2C_SDA* | *sI2C_SCL* |
| ***I2C_MODE_BOTH*** | P12.1 | P12.2 | P15.1 | P15.2 |
| ***I2C_MODE_MASTER*** | P15.1 | P15.2 | N/A | N/A |
| ***I2C_MODE_SLAVE*** | N/A | N/A | P15.1 | P15.2 |




› 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 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 on the slave. The I2C master reads the response from the slave and generates the next command.

**Follow these steps to configure this code example:**

- STDOUT setting
- GPIO port pin initialization[1]
- EZI2C 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.

# Implementation (contd.)

**STDOUT setting:**

› Call the ***cy_retarget_io_init()*** function to use UART as STDOUT
  - Initialize P13.1 as UART TX and P13.0 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.
  - Initialize P16.1 as output (initial level = H, LED turns off)
  - This is done when the code is configured as I2C slave (***I2C_MODE_BOTH*** or ***I2C_MODE_SLAVE*** is defined)

**EZI2C slave initialization:**

› The ***cyhal_ezi2c_init()*** function initializes the I2C peripheral
(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, 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 mI2C_SDA and mI2C_SCL pins
  - The ***cyhal_i2c_configure()*** function configures the I2C block to set it as master

**Send command packet to the slave:**

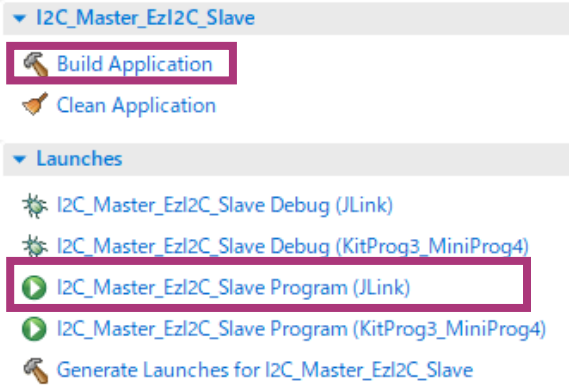
- I2C master sends command packet to the slave using the ***cyhal_i2c_master_write()*** function

**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[s] delayed using the ***cyhal_system_delay_ms()*** function

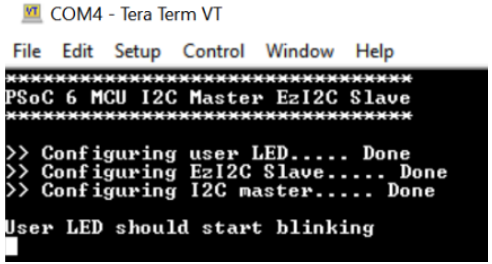
# Compiling and programming

1. Connect to power and USB cable
2. Use Eclipse IDE for ModusToolbox™ software for compiling and programming
3. Compile
   a) Select the target application project in the Project Explorer
   b) In the Quick Panel, scroll down and click "Build I2C_Master_EzI2C_Slave Application" in I2C Master EzI2C Slave (KIT-T2G-B-H-EVK)
4. Open a terminal program and select the KitProg3 COM port. Set the serial port parameters to 8N1 and 115200 baud.
5. 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 Launches

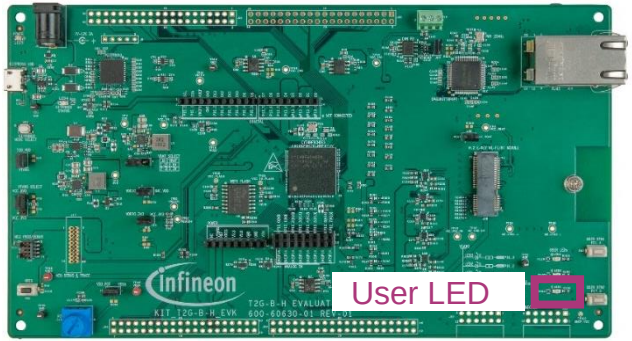



# Run and test

1. After successful programming, the application starts automatically. Confirm that the UART terminal displays the following:




2. Observe that the kit user LED blinks at 1 Hz.



# 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 |
|---|---|---|---|
| ** | 7782644 | 2022/07/06 | Initial release |
| *A | 7837066 | 2022/11/08 | Added comments on page 6 and page 7 |

# Important notice and warnings

All referenced product or service names and trademarks are the property of their respective owners.

**Edition 2022-11**
**Published by**
**Infineon Technologies AG**
**81726 Munich, Germany**



**Do you have a question about this document?**
**www.infineon.com/support**

**Document reference**
**002-35573 Rev. *A**

**IMPORTANT NOTICE**

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

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.