Customer training workshop: HAL_I2C_Master for KIT_T2G-B-H_EVK

TRAVEO™ T2G CYT4BF series Microcontroller Training
V1.0.2 2023-02

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 CYT4BFBCH device used in this code example.

- Board
  - The TRAVEO™ T2G KIT_T2G-B-H_EVK board used for testing.

Introduction

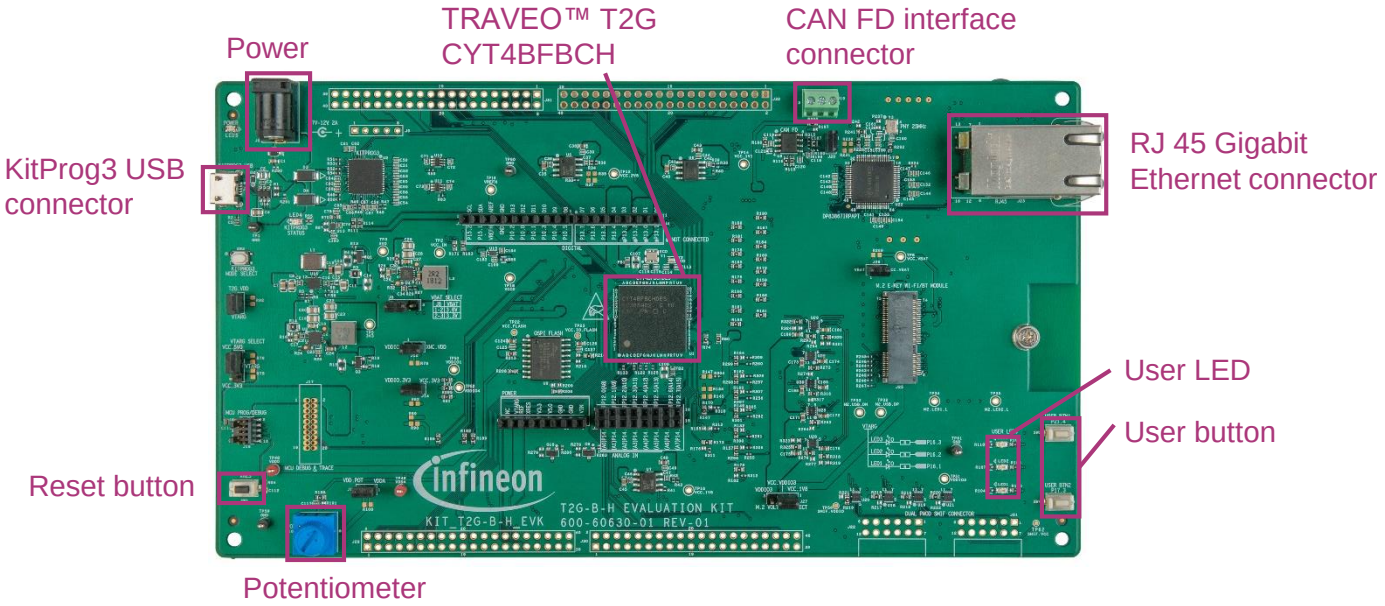
I2C has the following features

- 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-EVK board.

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

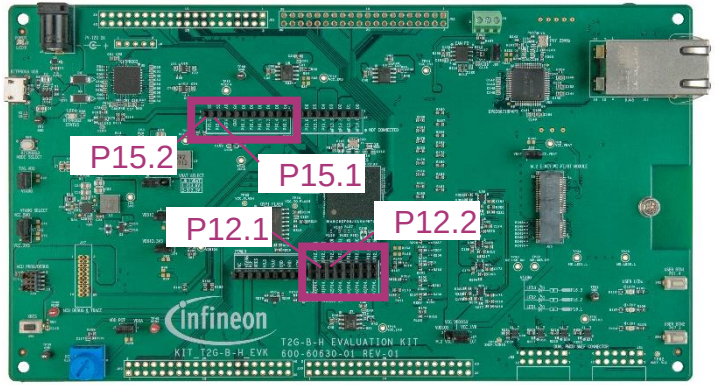


Hardware setup

› 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. Both 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 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

STDOUT setting

- Call the ***cy_retarget_io_init()*** function to use UART as STDOUT.
  - Initialize P13.1 as UART TX, P13.0 as UART RX (these pins are connected to KitProg3 COM port)
  - The serial port parameters change to 8N1 and 115200 baud

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

Implementation (contd.)

GPIO port pin initialization

- The ***cyhal_gpio_init()*** function initializes the GPIO port pin once.
  - Initialize P16.1 as output (initial level = H, LED turns off)
  - This can only be done when the code is configured as I2C slave (***I2C_MODE_BOTH*** or ***I2C_MODE_SLAVE*** is defined)

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***) occurs.

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 blinks according to the received packet by calling ***cyhal_gpio_write()***, and then updates the read buffer as ***STS_CMD_DONE.***
  - Configures write buffer for the next request by calling ***cyhal_i2c_slave_config_write_buffer()*.**

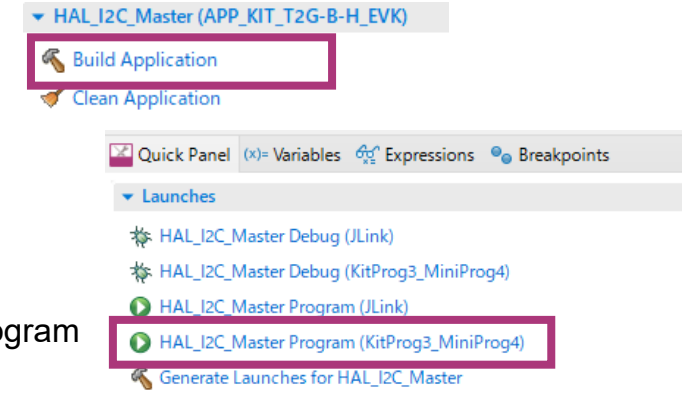
Implementation (contd.)

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.

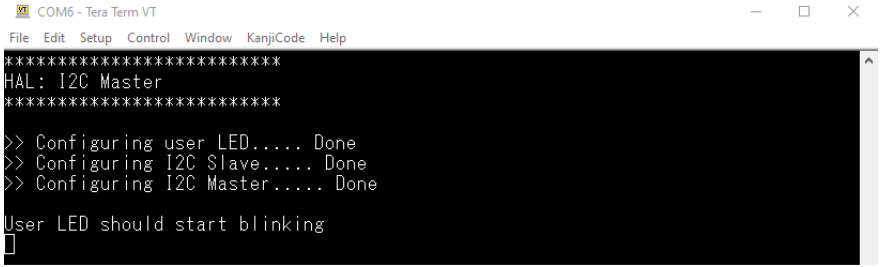
Compiling and programming

1. Connect to power and USB cable
2. Use Eclipse IDE for ModusToolbox™ software for compiling and programming
3. Compile
   a) Select the target application project in the Project Explorer.
   b) In the Quick Panel, scroll down, and click "Build HAL_I2C_Master Application" in HAL_I2C_Master(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 "HAL_I2C_Master Program (KitProg3_MiniProg4)" in Launches

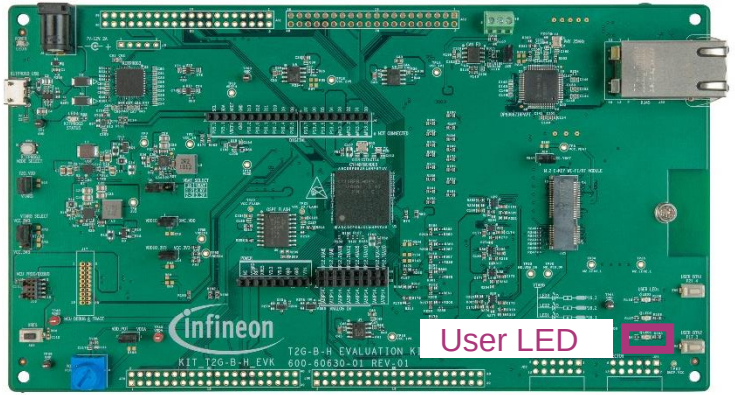


Run and test

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 when I2C communication starts.



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
**	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”

Important notice and warnings

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

Edition 2023-02
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Go to:
www.infineon.com/support

Document reference
002-35571 Rev. *B

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.