Customer training workshop: I2C_Slave_Using_Callbacks for KIT_T2G-B-H_EVK

TRAVEO™ T2G CYT4BF series Microcontroller Training
V1.0.0 2022-06

Please read the **Important notice and warnings** at the end of this document

Scope of work

- This code example demonstrates the operation of the I2C (HAL) resource in slave mode using callbacks.
- 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.

Copyright © Infineon Technologies AG 2022. All rights reserved.

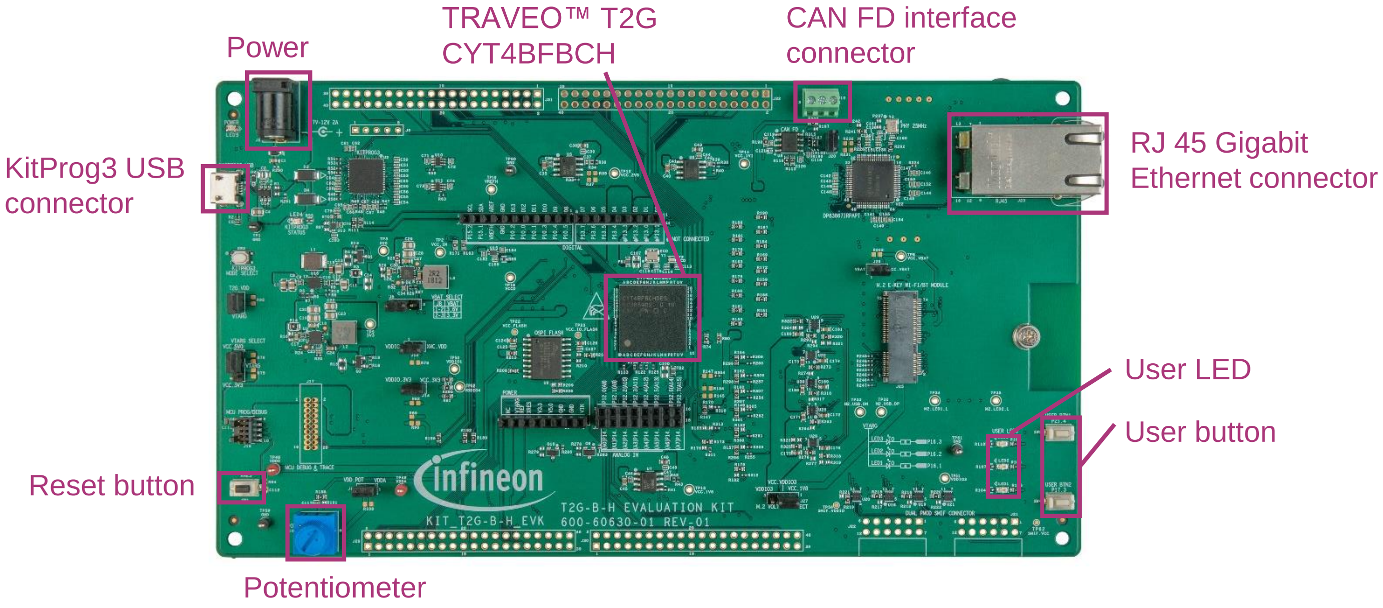
Introduction

› **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

Copyright © Infineon Technologies AG 2022. All rights reserved.

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.



Copyright © Infineon Technologies AG 2022. All rights reserved.

Implementation

This code example demonstrates the operation of the I2C (HAL) resource in slave mode using callbacks. It initializes PWM to control the LED and initializes I2C to slave mode. If the initialization of I2C/PWM fails, the system will be in an infinite loop. After initializing, the slave receives packets from the master and configures the PWM to drive the LED. The slave responds with the acknowledgment packet.

Follow these steps to configure this code example:

- PWM initialization with LED
- I2C slave initialization
- I2C slave receives packets from the master and control LED
- I2C slave configure read buffer for the next read

PWM initialization with LED:

- ***led_pwm_init()*** creates and configures a PWM object to drive PWM on LED.
  - The ***cyhal_pwm_init_adv()*** function initializes the PWM.
  - The ***cyhal_pwm_set_period()*** function sets the PWM period and initial pulse width**.**
  - The ***cyhal_pwm_start()*** function starts the PWM.

Copyright © Infineon Technologies AG 2022. All rights reserved.

Implementation (contd.)

I2C slave initialization:

- Creates and configures an I2C slave object to communicate with the I2C master by ***i2c_slave_init()***.
  - The ***cyhal_i2c_init()*** function initializes the I2C peripheral.
  - The ***cyhal_i2c_configure()*** function configures the I2C resource as a slave.
  - The ***cyhal_i2c_slave_config_write_buffer()*** function configures the I2C slave write buffer such that the I2C master can write into it**.**
  - The ***cyhal_i2c_slave_config_read_buffer()*** function configures the I2C slave read buffer such that the I2C master can read from it.
  - Register ***handle_i2c_slave_events()*** are call-backed when an I2C slave event occurs using the ***cyhal_i2c_register_callback()*** function.
  - The ***cyhal_i2c_enable_event()*** function configures and enables the I2C Interrupt event.

Copyright © Infineon Technologies AG 2022. All rights reserved.

Implementation (contd.)

I2C slave receives packets from the master and control LED:

- After the slave is received, ***CYHAL_I2C_SLAVE_WR_CMPLT_EVENT*** calls back ***handle_i2c_slave_events()*** by HAL and sets the pulse width to control the brightness of the LED using the ***cyhal_pwm_set_period()*** function.
- Configure write buffer for the next write using the ***cyhal_i2c_slave_config_write_buffer()*** function.

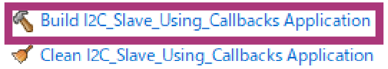
I2C slave configure read buffer for the next read:

- After the read event of the slave ***CYHAL_I2C_SLAVE_RD_CMPLT_EVENT*** is notified, configure the read buffer for the next read by ***cyhal_i2c_slave_config_read_buffer()*** function.

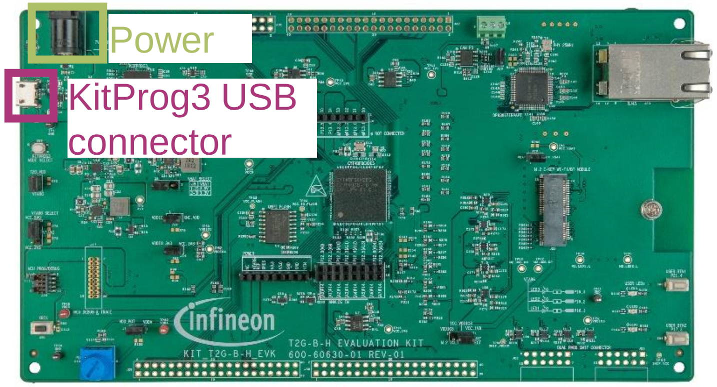
Copyright © Infineon Technologies AG 2022. All rights reserved.

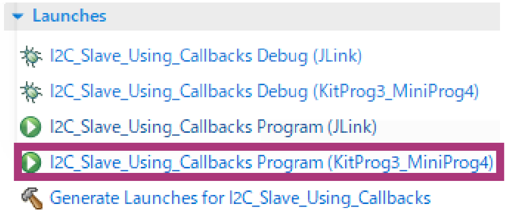
Compiling and programming

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_Slave_Using_Callbacks Application" in I2C Slave Using Callbacks(KIT-T2G-B-H-EVK)


Build I2C_Slave_Using_Callbacks Application
Clean I2C_Slave_Using_Callbacks Application

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_Slave_Using_Callbacks Program (KitProg3_MiniProg4)" in Launches

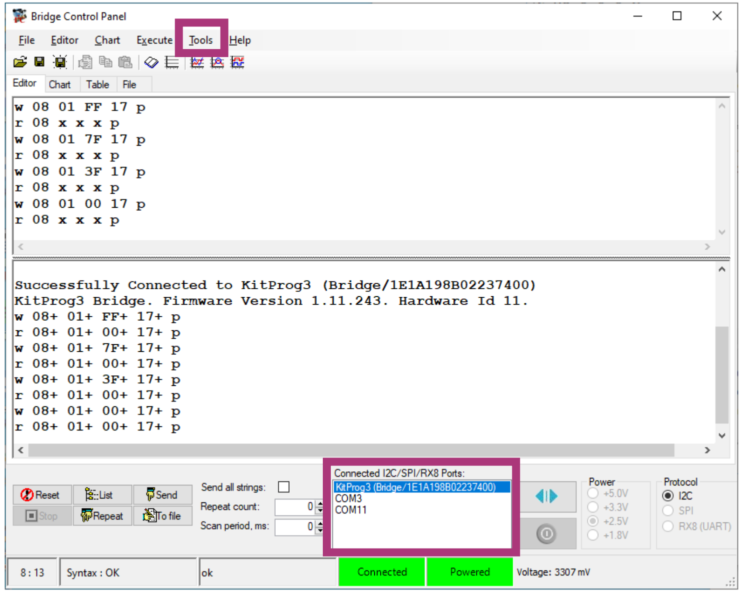



Launches
I2C_Slave_Using_Callbacks Debug (JLink)
I2C_Slave_Using_Callbacks Debug (KitProg3_MiniProg4)
I2C_Slave_Using_Callbacks Program (JLink)
I2C_Slave_Using_Callbacks Program (KitProg3_MiniProg4)
Generate Launches for I2C_Slave_Using_Callbacks

Copyright © Infineon Technologies AG 2022. All rights reserved.

Compiling and programming (contd.)

6. Open the Bridge Control Panel (BCP) software for transmitting and receiving data over I2C. The BCP software is installed automatically as part of the PSoC™ Programmer installation.
   1) Open **Bridge Control Panel**.
   2) Select **KitProg3** under **Connected I2C/SPI/RX8 Ports**.
   3) Select **Tools** > **Protocol Configuration**, navigate to the **I2C** tab, and set the **I2C speed** to **100 kHz.** Click **OK**.



```
w 08 01 FF 17 p
r 08 x x x p
w 08 01 7F 17 p
r 08 x x x p
w 08 01 3F 17 p
r 08 x x x p
w 08 01 00 17 p
r 08 x x x p
```

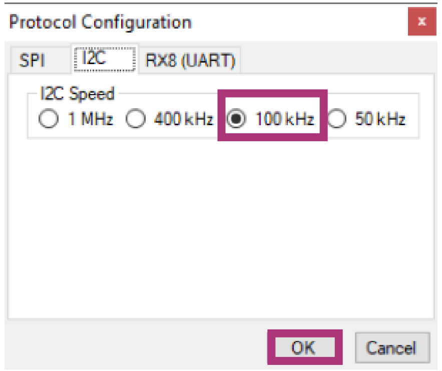
```
Successfully Connected to KitProg3 (Bridge/1E1A198B02237400)
KitProg3 Bridge. Firmware Version 1.11.243. Hardware Id 11.
w 08+ 01+ FF+ 17+ p
r 08+ 01+ 00+ 17+ p
w 08+ 01+ 7F+ 17+ p
r 08+ 01+ 00+ 17+ p
w 08+ 01+ 3F+ 17+ p
r 08+ 01+ 00+ 17+ p
w 08+ 01+ 00+ 17+ p
r 08+ 01+ 00+ 17+ p
```

Send all strings: ☐
Repeat count: 0
Scan period, ms: 0

Connected I2C/SPI/RX8 Ports:
[✓ KitProg3 (Bridge/1E1A198B02237400)]
COM3
COM11

Power: +5.0V, +3.3V, [✓ +2.5V], +1.8V
Protocol: [✓ I2C], SPI, RX8 (UART)

8 : 13 | Syntax : OK | ok | Connected | Powered | Voltage: 3307 mV



Protocol Configuration
SPI | [✓ I2C] | RX8 (UART)
I2C Speed: 1 MHz, 400 kHz, [✓ 100 kHz], 50 kHz
OK | Cancel

BCP is now ready for transmitting and receiving data.

Copyright © Infineon Technologies AG 2022. All rights reserved.

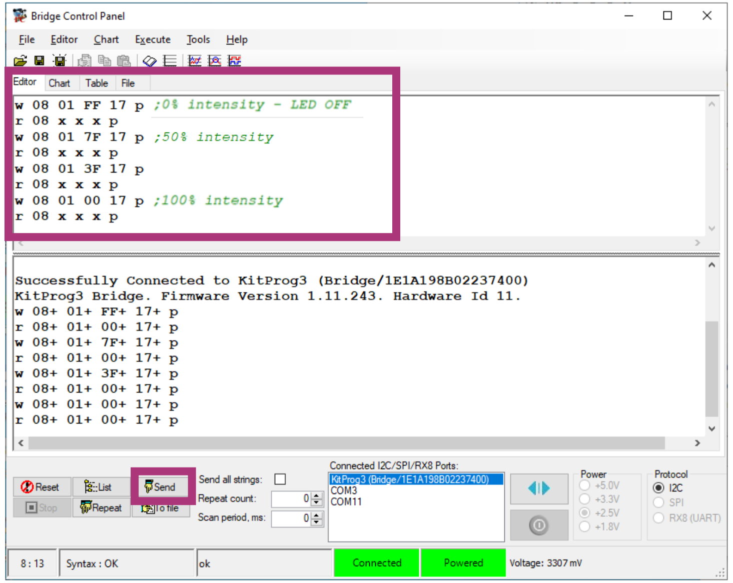
Run and test

1. After successful programming, the application starts automatically.
2. Open BCP[1] to send the command to control the LED.
3. In the **Editor** tab of BCP, type the command to send the LED intensity data and then click **Send**. Observe that the LED turns ON with the specified intensity.
4. The command format that is used to write the data to the slave.

   In this example, BCP is used as the I2C master.

   – Sending the “**w 08 01 00 17 p**” command turns ON the LED with full intensity; sending the “**w 08 01 FF 17 p**” command turns OFF the LED.

w/r	Slave address	SoP (Start of packet)	LED TCPWM compare value	EoP (End of packet)	Stop
w	0x08	0x1	0xFF	0x17	P
w	0x08	0x1	0x00	0x17	P


Bridge Control Panel
File Editor Chart Execute Tools Help
Editor Chart Table File
Reset List Send Stop Repeat To file
Power +5.0V +3.3V +2.5V +1.8V
Protocol I2C SPI RX8 (UART)

```
w 08 01 FF 17 p ;0% intensity - LED OFF
r 08 x x x p
w 08 01 7F 17 p ;50% intensity
r 08 x x x p
w 08 01 3F 17 p
r 08 x x x p
w 08 01 00 17 p ;100% intensity
r 08 x x x p
```

```
Successfully Connected to KitProg3 (Bridge/1E1A198B02237400)
KitProg3 Bridge. Firmware Version 1.11.243. Hardware Id 11.
w 08+ 01+ FF+ 17+ p
r 08+ 01+ 00+ 17+ p
w 08+ 01+ 7F+ 17+ p
r 08+ 01+ 00+ 17+ p
w 08+ 01+ 3F+ 17+ p
r 08+ 01+ 00+ 17+ p
w 08+ 01+ 00+ 17+ p
r 08+ 01+ 00+ 17+ p
```

Send all strings: ☐
Repeat count: 0
Scan period, ms: 0
Connected I2C/SPI/RX8 Ports: KitProg3 (Bridge/1E1A198B02237400), COM3, COM11
Power: +2.5V
Protocol: [✓ I2C]
8 : 13 | Syntax : OK | ok | Connected | Powered | Voltage: 3307 mV

[1]: This code example cannot be debug via “KitProg3_MiniProg4” during using BCP, because both of them use the same KitProg3 port.

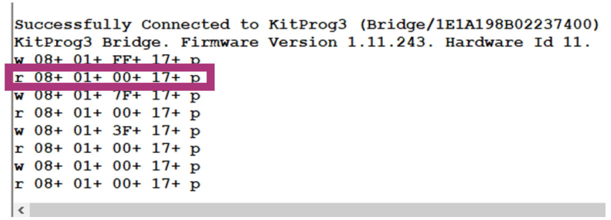
Copyright © Infineon Technologies AG 2022. All rights reserved.

Run and test (contd.)

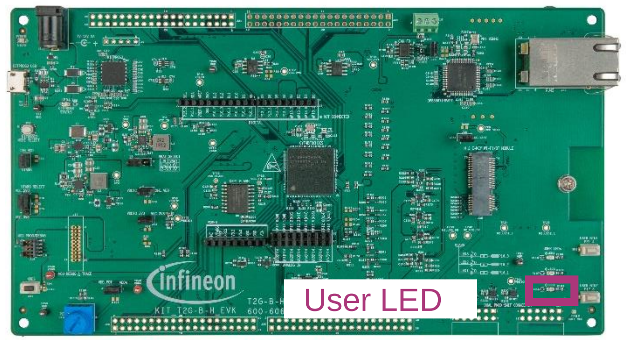
5. Type the “**r 08 x x x p**” command to read the status of the write performed.
   - The following is the command format to read the status from the slave’s read buffer. The symbol ‘**x**’ denotes one byte to read from the slave’s read buffer. In this example, three bytes are read from the slave.
   - After each command is sent, the status packet must be read from the read buffer of the slave by sending the “**r 08 x x x p**” command.

w/r	**Slave address**	**SoP (Start of packet)**	**LED TCPWM compare value**	**EoP (End of packet)**	**Stop**
r	0x08	x	x	x	P

   - If the packet read is in the format “**r 08 01 00 17 p**”, the status is set as '**success**'; if the packet read is “**r 08 01 FF 17 p**”, the status is set as '**fail**' for the command sent by the master.



```
Successfully Connected to KitProg3 (Bridge/1E1A198B02237400)
KitProg3 Bridge. Firmware Version 1.11.243. Hardware Id 11.
w 08+ 01+ FF+ 17+ p
r 08+ 01+ 00+ 17+ p
w 08+ 01+ 7F+ 17+ p
r 08+ 01+ 00+ 17+ p
w 08+ 01+ 3F+ 17+ p
r 08+ 01+ 00+ 17+ p
w 08+ 01+ 00+ 17+ p
r 08+ 01+ 00+ 17+ p
```



6. Observe that the LED turns ON/OFF with the specified intensity.

Copyright © Infineon Technologies AG 2022. All rights reserved.

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

Copyright © Infineon Technologies AG 2022. All rights reserved.

Revision history

Revision	ECN	Submission date	Description of change
**	7782536	2022/07/06	Initial release

Copyright © Infineon Technologies AG 2022. All rights reserved.

Important notice and warnings

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

Edition 2022-05
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2022 Infineon Technologies AG.
All Rights Reserved.

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

Document reference
002-35572 Rev. **

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.