

Customer training workshop: CAN_FD 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 how to use controller area network flexible data-rate (CAN FD). In this example, the CAN Node-1 sends a CAN FD or standard frame to CAN Node-2 on pressing the user button (P21.4) and vice versa. Both the CAN nodes log the received data over the UART serial terminal. Each time a CAN frame is received, the user LED (P16.1) toggles.
- › This code example uses two EVK boards for CAN NODE 1 and CAN NODE 2.

- › 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.

› **The CAN FD controller has the following features:**

- Flexible data-rate (FD) (ISO 11898-1: 2015)
 - Up to 64 data bytes per message
 - Maximum 8 Mbps supported
- Time-Triggered (TT) communication on CAN (ISO 11898-4: 2004)
 - TTCAN protocol level 1 and level 2 completely in hardware
- Acceptance filtering
- Two configurable receive FIFOs
- Up to 64 dedicated receive buffers
- Up to 32 dedicated transmit buffers
- Configurable transmit FIFO
- Configurable transmit queue
- Configurable transmit event FIFO

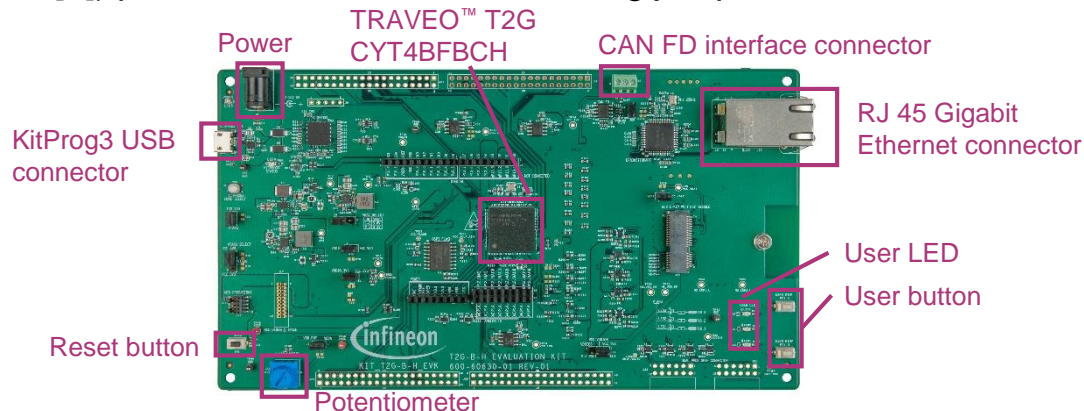
Introduction (contd.)

› **The CAN FD controller has the following features:**

- Programmable loop-back test mode
- Power-down support
- Shared message RAM
- ECC protection for message RAM
- Global fault structure to handle ECC errors
- Receive FIFO top pointer logic
 - Enables DMA access on the FIFO
- DMA for debug message and received FIFOs
- Shared time stamp counter

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
- › This code example requires two kits with the following settings:
 - One kit should be programmed with the **CAN_NODE_1** macro and another kit should be programmed with the **CAN_NODE_2** macro in the main.c file
 - Connect CANL (J19[1]) pin of Node-1 and Node-2 using jumper wires
 - Connect CANH (J19[2]) pin of Node-1 and Node-2 using jumper wires



Implementation

In this code example, the CAN FD block is configured. On pressing the user button on any node, the CAN frame is sent to the other node. The received frame can be viewed in the serial terminal. The USER_LED toggles at the receiver node on proper reception of the CAN frame.

Follow these steps to configure this code example:

- › STDOUT setting
- › Configure the button and CAN FD interrupt
- › Initialize the CAN FD channel
- › Send the CAN FD frame to the other node

STDOUT setting

- › The [cy_retarget_io_init\(\)](#) function initializes the GPIO for UART.
 - 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

Implementation (contd.)

Configure the button and CAN FD interrupt

- › The interrupt service routines (ISR) are registered by the *isr_button ()* and *isr_canfd ()* function
 - *isr_button ()*
 - When the user button (P21.4) is pressed, set the **ButtonIntrFlag** to send the CAN FD frame
 - Checks the interrupt status by [Cy_GPIO_GetInterruptStatusMasked \(\)](#)
 - Then clear the interrupt by [Cy_GPIO_ClearInterrupt \(\)](#)
 - *isr_canfd ()*
 - The ISR only needs to call [Cy_CANFD_IrqHandler \(\)](#) to make the configured CAN FD function work

Initialize the CAN FD channel

- › The [Cy_CANFD_Init \(\)](#) function initializes the CAN FD channel.
 - The function to be called back when received, **canfd_rx_callback ()**, is registered
 - Toggle the user LED (P16.1) by calling [Cy_GPIO_Inv \(\)](#)
 - Print the received CAN FD message
 - You can select the CAN frame type by setting **USE_CAN_MODE** to **CAN_CLASSIC_MODE** (CAN standard frame) or **CAN_FD_MODE** (CAN FD frame, default)

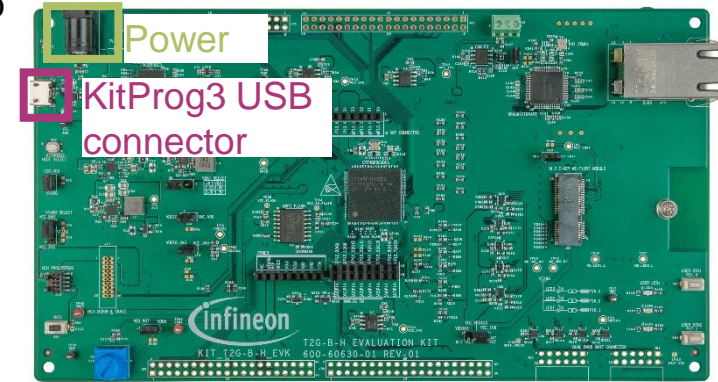
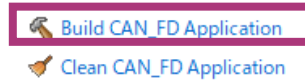
Implementation (contd.)

Sending the CAN FD frame to the other node

- › In the main loop, send the CAN FD frame to the other node by calling the [Cy_CANFD_UpdateAndTransmitMsgBuffer \(\)](#) function if the **ButtonIntrFlag** is set

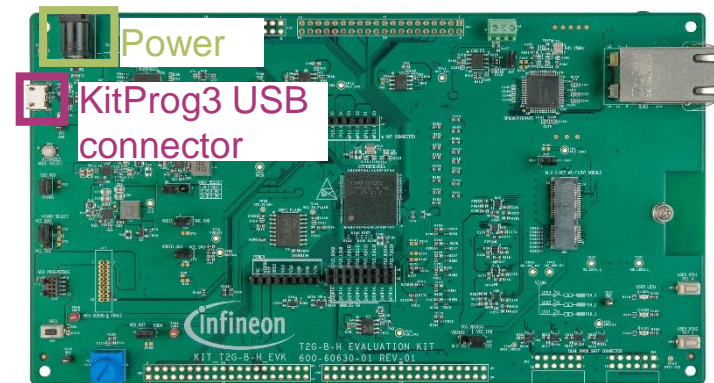
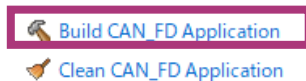
Compiling and programming (CAN Node-1)

1. Connect the CAN Node-1 kit to power and USB cable
2. Use Eclipse IDE for ModusToolbox™ software for compiling and programming with the CAN_NODE_1 macro
3. Compile
 - a) Select the target application project in the Project Explorer
 - b) In the Quick Panel, scroll down, and click “Build CAN_FD Application” in CAN_FD (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 “CAN_FD Program (KitProg3_MiniProg4)” under Launches



Compiling and programming (CAN Node-2)

1. Connect the CAN Node-2 kit to power and USB cable
2. Use Eclipse IDE for ModusToolbox™ software for compiling and programming with the CAN_NODE_2 macro
3. Compile
 - a) Select the target application project in the Project Explorer
 - b) In the Quick Panel, scroll down, and click “Build CAN_FD Application” in CAN_FD (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 “CAN_FD Program (KitProg3_MiniProg4)” under Launches

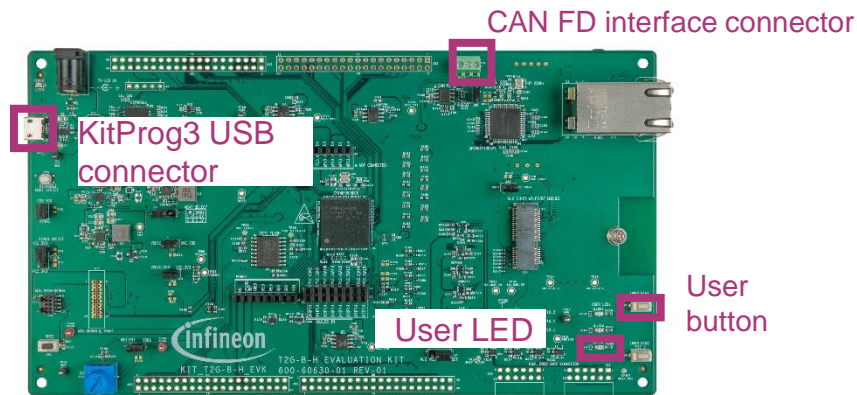


Run and test

1. Connect two kits as shown in the following screenshots and connect Node-2 to your PC using a USB cable:
 - Connect CANL (J19[1]) pin of Node-1 and Node-2 using jumper wires
 - Connect CANH (J19[2]) pin of Node-1 and Node-2 using jumper wires
2. On pressing the user button (P21.4) on Node-1, the CAN frame is sent to Node-2. The receiver frame can be viewed on the terminal as shown in the following screenshot. The user LED toggles (P16.1) at the receiver node on proper reception of the CAN frame.

```

COM14 - Tera Term VT
File Edit Setup Control Window Help
=====
XMC7000 MCU: CANFD example
=====
CAN FD Node-1
=====
CANFD frame sent from Node-1
8 bytes received from Node-2 with identifier 2
Rx Data : 1 2 3 4 5 6 7 8
  
```



3. You can also send the CAN frame from Node-2 by pressing the user button on Node-2, and confirm the receiver frame and LED toggling on Node-1.

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
**	7781933	2022/06/29	Initial release

Important notice and warnings

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

Edition 2022-06

**Published by
Infineon Technologies AG
81726 Munich, Germany**

**© 2022 Infineon Technologies
AG.**

All Rights Reserved.

**Do you have a question about
this document?**

Go to:
www.infineon.com/support

**Document reference
002-35578 Rev. ****

IMPORTANT NOTICE

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

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.