CAN_FD
for KIT_T2G-B-H_LITE

Customer training workshop

Q3 2024

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 (SW1) 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 (LED3) toggles.

- This code example uses two LITE boards for CAN NODE 1 and CAN NODE 2

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

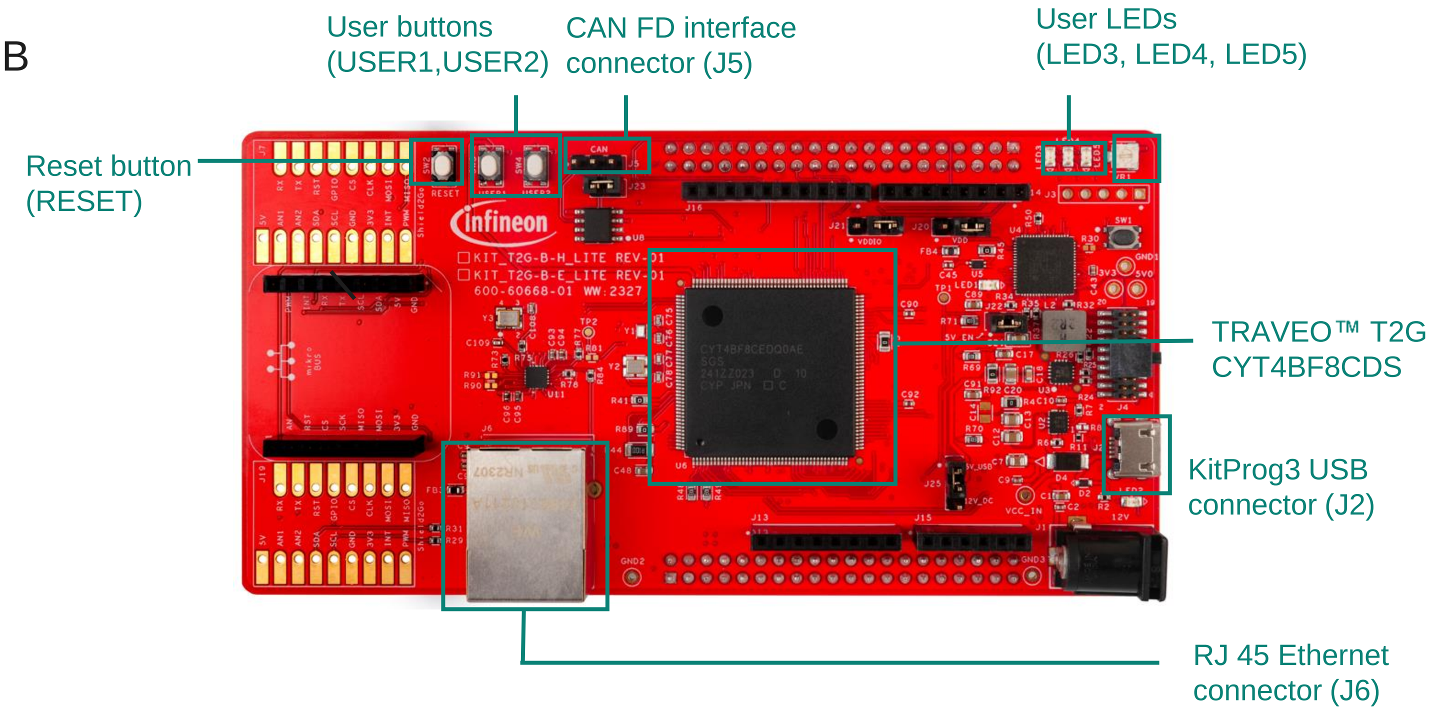
- **Board**
  - The TRAVEO™ T2G KIT_T2G-B-H_LITE board is used for testing

Introduction

- **CAN FD 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
  - 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_LITE board
- Connect the PC to the board using the provided USB cable through the KitProg3 USB connector (J2)



User buttons (USER1,USER2)

CAN FD interface connector (J5)

User LEDs (LED3, LED4, LED5)

Reset button (RESET)

TRAVEO™ T2G CYT4BF8CDS

KitProg3 USB connector (J2)

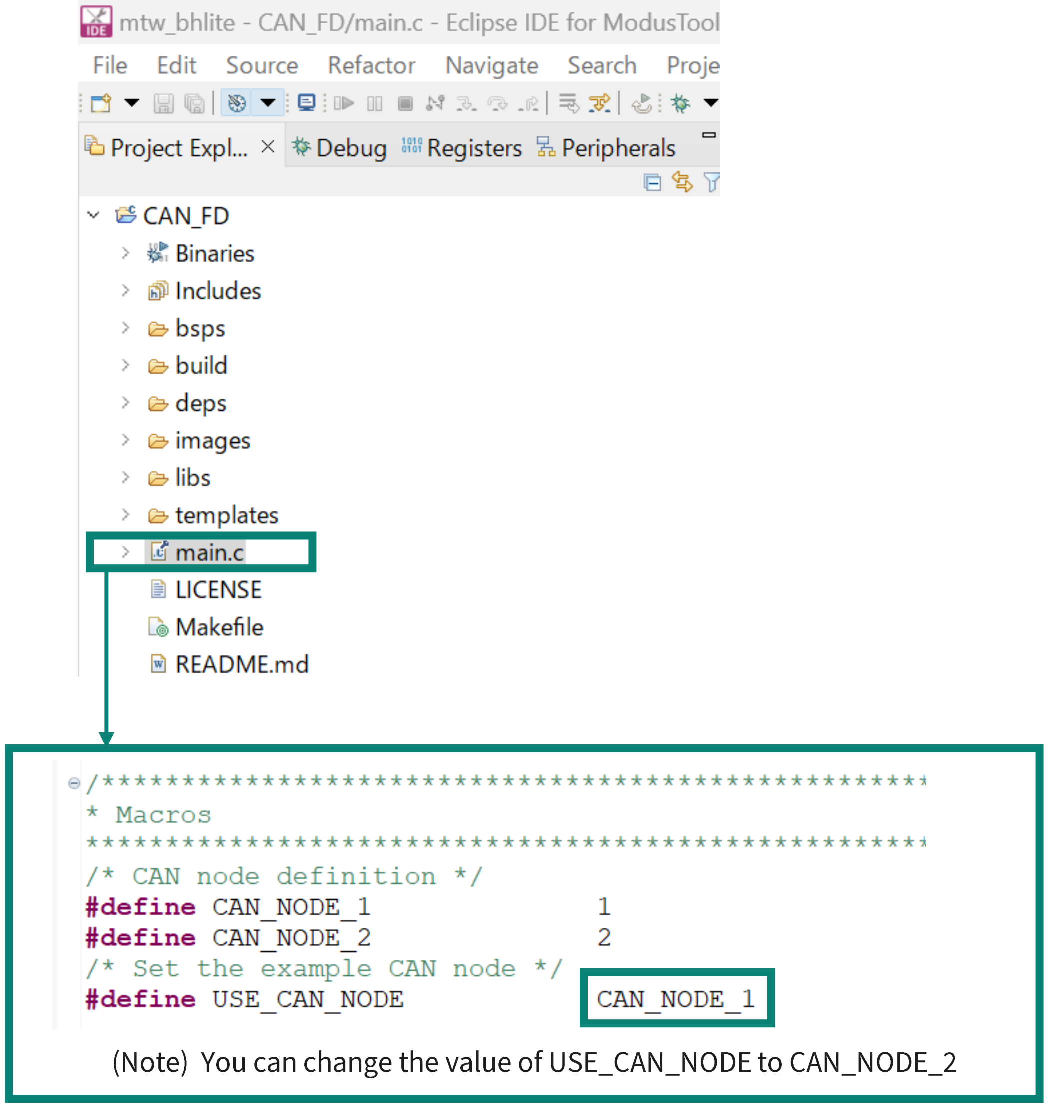
RJ 45 Ethernet connector (J6)

Implementation

- In this code example, the CAN FD block is configured. On pressing the user button (USER1) 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 (LED3) toggles at the receiver node on proper reception of the CAN frame.
- The default settings in this code example are configured to operate in “**Node-1**". To set it to “**Node-2**", you can change the value of USE_CAN_NODE to CAN_NODE_2 in the main.c file

- **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 P0.1 as UART TX, P0.0 as UART RX (these pins are connected to KitProg3 COM port)
    - The serial port parameters change to 8N1 and 115200 baud



```
/*******************************************************
 * Macros
 *******************************************************
/* CAN node definition */
#define CAN_NODE_1                  1
#define CAN_NODE_2                  2
/* Set the example CAN node */
#define USE_CAN_NODE                CAN_NODE_1
```

(Note) You can change the value of USE_CAN_NODE to CAN_NODE_2

Implementation (contd.)

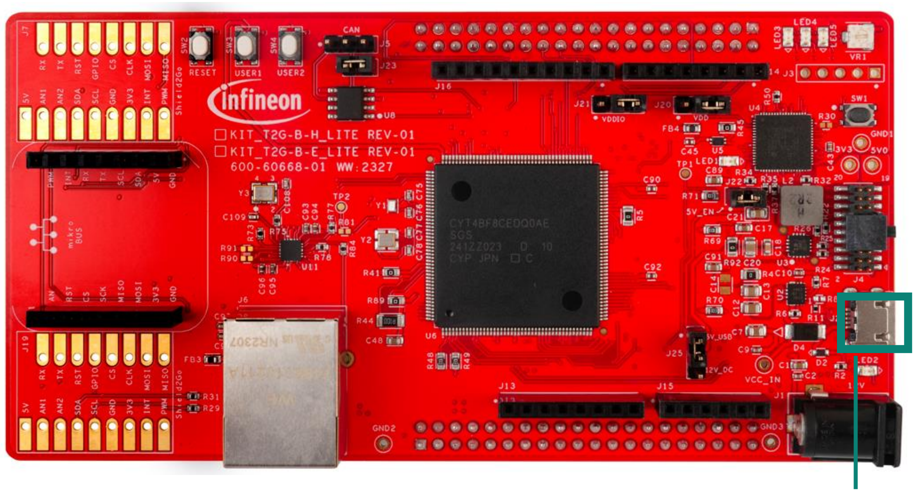
- **Configure the button and CAN FD interrupt**
  - The interrupt service routines (ISR) are registered by the ***isr_button ()*** and ***isr_canfd ()*** function
    - ***Isr_button ()***
      - When the user button (P5.3/USER1) is pressed, set the ***ButtonIntrFlag*** to send the CAN FD frame
      - 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 (P5.0/LED3) 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)

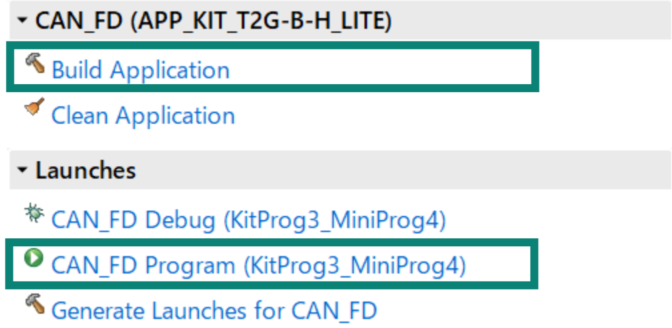
- **Sending the CAN FD frame to the other node**
  - In the main loop, send the CAN FD frame to the other node by calling the ***Cy_CANFD_UpdateAndTransmitMsgBuffer ()*** function if the ***ButtonIntrFlag*** is set

Compiling and programming

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 CAN_FD (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 **CAN_FD Program (KitProg3_MiniProg4)** in the Launches



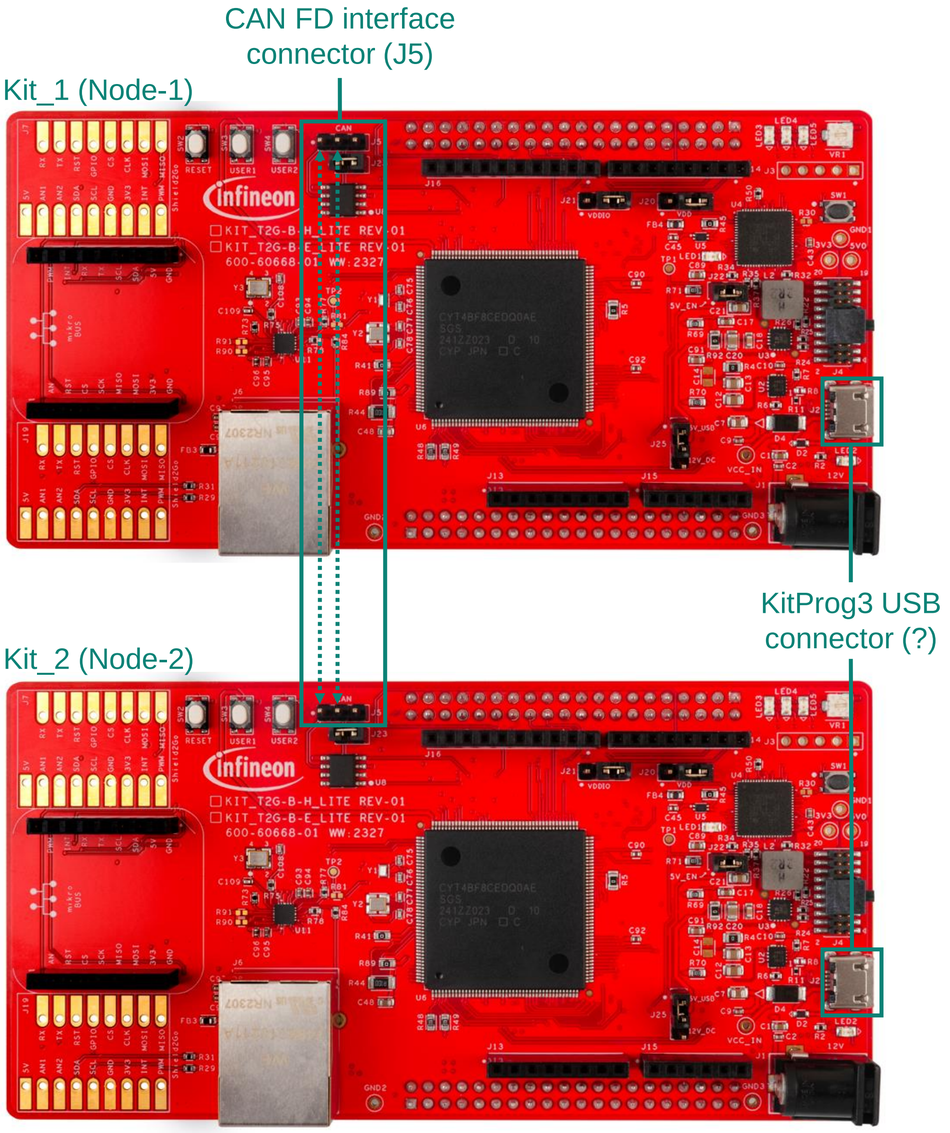
KitProg3 USB connector (J2)



▾ CAN_FD (APP_KIT_T2G-B-H_LITE)
Build Application
Clean Application
▾ Launches
CAN_FD Debug (KitProg3_MiniProg4)
CAN_FD Program (KitProg3_MiniProg4)
Generate Launches for CAN_FD

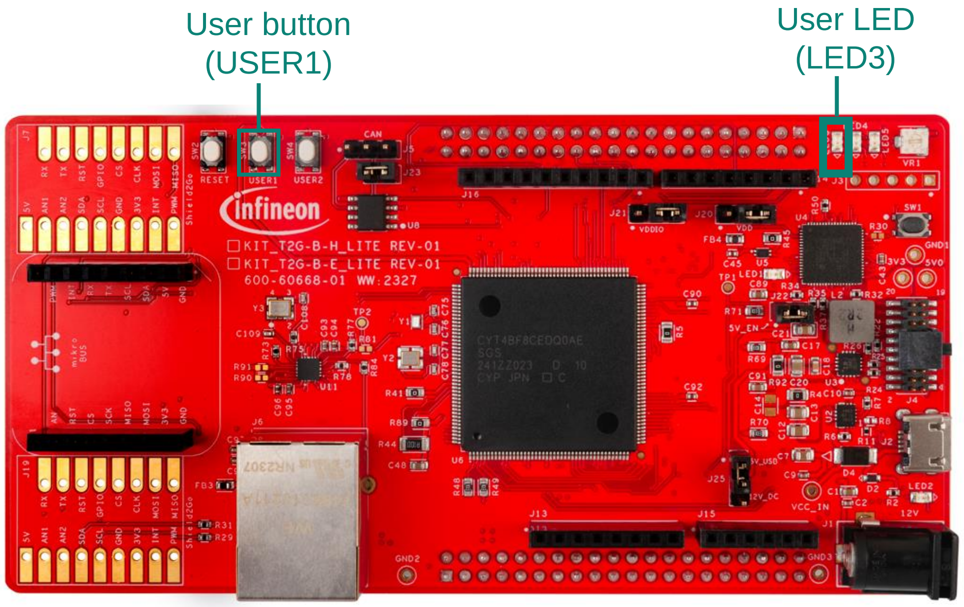
Run and test

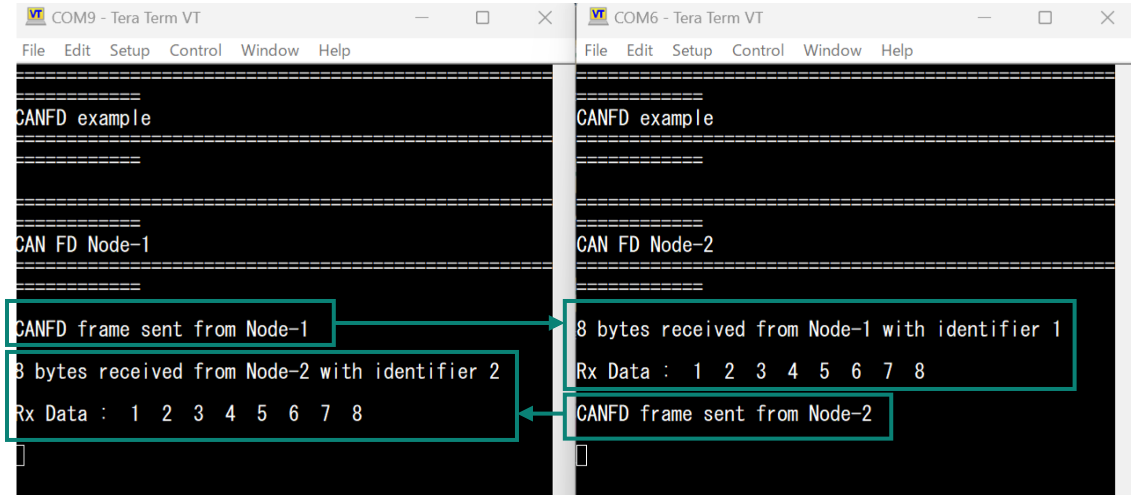
1. Connect two kits as shown in the following screenshots and connect two kits to your PC using an USB cable (see the dotted line in the figures).
   a. Connect CANL (J5[1]) pin of Node-1 and Node-2 using jumper wires
   b. Connect CANH (J5[2]) pin of Node-1 and Node-2 using jumper wires



Run and test (contd.)

2. On pressing the user button (USER1) 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 (LED3) toggles at the receiver node on proper reception of the CAN frame.
3. The CAN frame from Node-2 can be sent by pressing the user button on Node-2 and confirm the receiver frame and user LED (LED3) toggling on Node-1.




COM9 - Tera Term VT
File Edit Setup Control Window Help
COM6 - Tera Term VT
File Edit Setup Control Window Help

```
===============================================================
===========
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
```

Node-1

```
===============================================================
===========
CANFD example
===============================================================
===========

===============================================================
===========
CAN FD Node-2
===============================================================
===========
8 bytes received from Node-1 with identifier 1
Rx Data :  1  2  3  4  5  6  7  8
CANFD frame sent from Node-2
```

Node-2

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
**	7781933	2022/06/29	Initial release
*A	8085970	2024/10/31	Replaced development board from KIT_T2G-B-H_EVK to KIT_T2G-B-H_LITE