CAN FD Peripheral on the RutDevKit-PSoC62 Development Board

Gintaras Drukteinis,
RUTRONIK Electronics Worldwide, Kaunas, Lithuania

***Abstract* — This application note describes the CAN FD hardware and firmware example on the RutDevKit-PSoC62 development platform with CY8C6245AZI-S3D72 microcontroller and TLE9251VLE driver.**

Index Terms* — *Microcontroller (MCU), Medium Access Control (MAC), Electrostatic discharge (ESD).

I. Introduction

The RutDevKit-PSoC62 is a unique development platform for the PSoC6 microcontroller which has the CAN FD peripheral development-ready on board. A simple CAN FD "Loop-back Test" example is also prepared, so developers must not start to develop the custom application from scratch.

II. Hardware

The onboard CY8C6245AZI-S3S72 has an integrated CAN FD controller which complies with the ISO11898-1 and ISO11898-4 CAN FD protocol specifications. The external CAN FD driver TLE9251VLE was chosen. Though the PSoC6 CAN FD allows speeds up to the 8 Mbps data transfers, the driver is supposed to be used up to the 5 Mbps, hence this is the maximum data rate the RutDevKit-PSoC62 can provide.

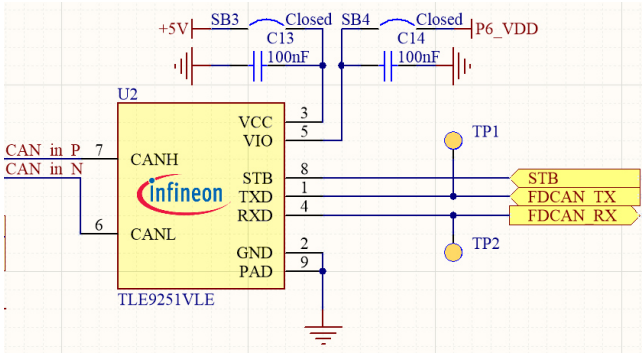

Fig. 1 The Infineon CAN FD driver on the RutDevKit-PSoC62.

The TLE9251VLE is not isolated from the microcontroller electrically, but there is additional ESD protection and resettable fuses on the external CAN+ and CAN- lines (Fig. 2). Additionally, the 120Ω line termination resistor can be enabled using solder bridge SB5 and it must be done if testing in a "Loop-back Mode".

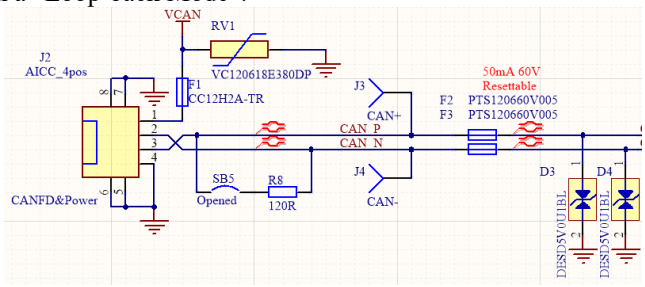

Fig. 2 The CAN FD terminal.

The TLE9251VLE driver is qualified for automotive applications. Therefore, the automotive 4 positions connector Minitek MicroSpace™ 10142344-104KLF is used for providing power for the board and connecting to the CAN network.

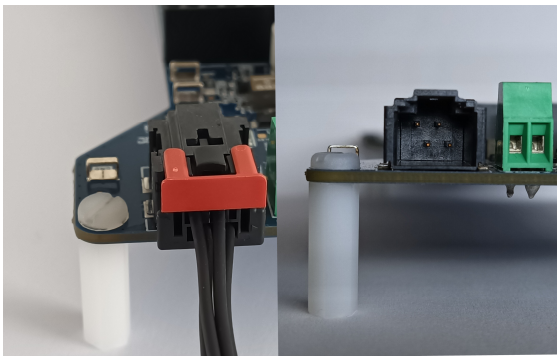

Fig. 3 The Amphenol ICC Minitek MicroSpace™ 10142344-104KLF connector and cable.

The TLE9251VLE driver is qualified for automotive applications. Therefore, the automotive 4 positions connector Minitek MicroSpace™ 10142344-104KLF is used for providing power for the board and connecting to the CAN network (Fig. 3).

Since the CAN FD physical layer has differential CAN+ and CAN- signals capable of handling data frames up to 5 Mbps, the PCB design should be provisioned accordingly. The best would be to keep CAN+ and CAN- traces under strict differential lines rules control and 120Ω impedance match. The very beginning of the traces at the CAN FD terminal has to be wide enough to hold excessive currents flowing at the ESD or any other fault event.

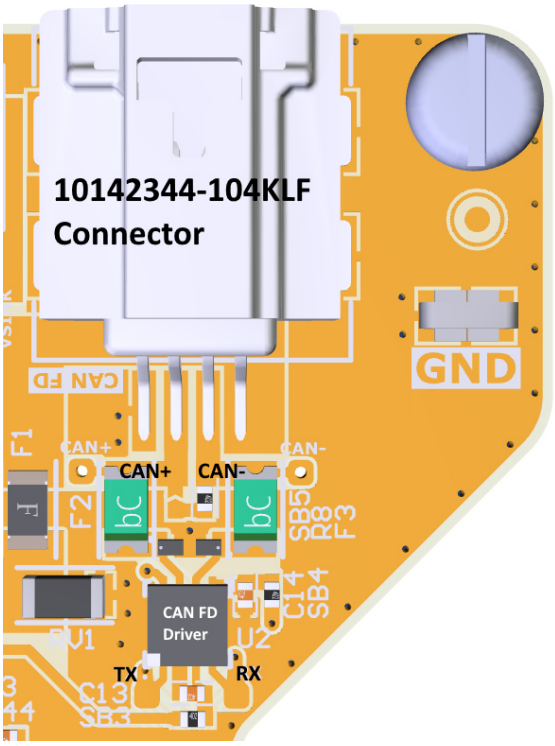

Fig. 4 The Amphenol ICC Minitek MicroSpace™ 10142344-104KLF connector and cable.

III. Firmware Example

The purpose of the firmware example "RutDevKit-PSoC62 CAN FD Test" is to get the user familiarized with the "PSoC6 Peripheral Driver Level - PDL" library functions for the CAN FD peripheral. The example does not require the CAN network to be attached since the system is initialized to work in external loop-back mode all the time. All the transmitted packets are received again by the controller.

The CAN FD configuration is done using the "Device Configurator" which is one of many tools provided by ModusToolbox™ IDE. The generated configuration structure *CAN_FD_config* in *cycfg_peripherals.c* can be used directly to initialize the CAN FD as it is shown below.

```
/* Configuration structure and CAN FD global variables */
const cy_stc_canfd_config_t CAN_FD_config = {...};
cy_stc_canfd_context_t canfd_context;

/* Initialization of the CAN FD Peripheral */
if(CY_CANFD_SUCCESS != Cy_CANFD_Init(CANFD0, 0,
&CAN_FD_config, &canfd_context))
{
  handle_error();
}
```

All the received data is printed out via the KitProg3 UART terminal (Fig. 5). The data is printed in the CAN FD reception callback function. This only works if the interrupt handler of the CAN FD peripheral is configured and enabled.

```
const cy_stc_sysint_t irq_cfg =
{
    canfd_0_interrupts0_0_IRQn,
    3UL
};
/* Hook the interrupt service routine and enable the
interrupt */
(void) Cy_SysInt_Init(&irq_cfg, &CanfdInterruptHandler);
NVIC_EnableIRQ(canfd_0_interrupts0_0_IRQn);
```

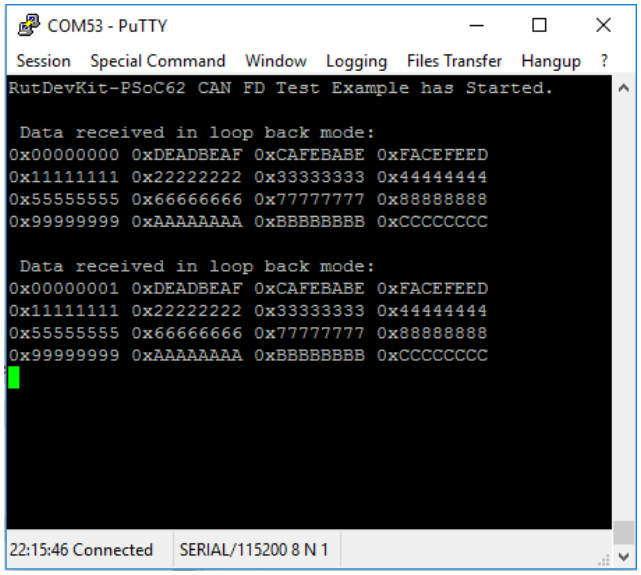

Fig. 5 The data received and printed from the reception callback function.

To have the "External Loop-Back Mode" enabled the following procedure must be done before transmitting any data:

```
/* Enables the configuration changes to set Test mode */
Cy_CANFD_ConfigChangesEnable(CANFD0, 0);
/* Sets the Test mode configuration */
Cy_CANFD_TestModeConfig(CANFD0, 0,
CY_CANFD_TEST_MODE_EXTERNAL_LOOP_BACK);
/* Disables the configuration changes */
Cy_CANFD_ConfigChangesDisable(CANFD0, 0);
```

Finally, the test data packed of 16 bytes can be sent and received in a callback as shown below:

```
uint32_t data[16] =
{0x00000000,0xDEADBEAF,0xCAFEBABE,0xFACEFEED,0x11111111,0
x22222222,0x33333333,0x44444444,0x55555555,0x66666666,0x7
7777777,0x88888888,0x99999999,0xAAAAAAAA,0xBBBBBBBB,0xCCC
CCCCC,};

Cy_CANFD_UpdateAndTransmitMsgBuffer(CANFD0, 0u,
&CAN_FD_txBuffer_0, 0u, &canfd_context);

/* CANFD reception callback */
void CANFD_RxMsgCallback(bool bRxFifoMsg, uint8_t
u8MsgBufOrRxFifoNum, cy_stc_canfd_rx_buffer_t*
pstcCanFDmsg)
{/* Data reception and printing is done here*/}
```

IV. MONITORING THE CAN FD

The "External Loop-Back Mode" allows users to monitor a differential signal using the oscilloscope. In (Fig. 6) it is seen how two-channel oscilloscope measures both CAN+/CAN- signals in parallel. Each CAN signal is measured with a reference to the GND. The CAN signals can be measured with a probe from test points placed on a board marked as "CAN+" and "CAN-" or it can be measured using the CAN cable. Note: The onboard termination resistor must be enabled while doing the measurements.

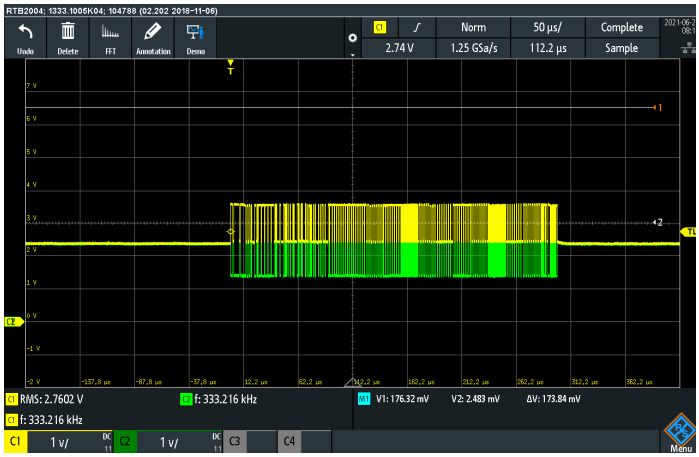

Fig. 6 The differential signal monitoring on CAN+ (yellow) and CAN- (green) lines.

V. SUMMARY

Although the CAN FD driver TLE9251VLE and connector 10142344-104KLF qualify for automotive applications, they can be used for industrial and consumer applications as well. The testing was done with 11-bit addressing and CAN FD frame format at 2 Mbps, but the driver could take the speeds up to 5 Mbps.

The RutDevKit-PSoC62 is tested and prepared for applications development that might have complex CAN FD protocols.

REFERENCES

[1] "PSoC 6 MCU: CY8C6xx5 Architecture Technical Reference Manual (TRM)" User Manual, by Infineon (July 2020).
[2] "PSoC 6 MCU: CY8C62X5 Datasheet" Datasheet, by Infineon (November 2020).
[3] "TLE9251V High Speed CAN FD Transceiver" Datasheet, by Infineon (April 2018).
[4] "PSoC 6 Peripheral Driver Library" Website.

Contact:
Gintaras Drukteinis
Technical Support Engineer
RUTRONIK Elektronische Bauelemente GmbH
Jonavos g. 30
44262 Kaunas
Lithuania
gdr@rutronik.com
www.rutronik.com