

PIC32 CAN RTR Code Example using Extended ID CAN Message Communication

Richard W. Wall

October 28, 2014

* Note: the path for this project requires that "../h" must be added to the GCC compiler search path

* Jumpers JP1 and JP2 on the chipKIT Pro MX7 board must be in the "CAN" position.

What you need to run this code example:

1. chipKIT Pro MX7
2. CAN node connection cable.
3. MPLAB X version 2.10 +
4. XC32 1.30

About the Code Example:

This code example demonstrates the use of the PIC32MX CAN Peripheral Library and PIC32MX CAN Module to send and receive Extended ID Remote Transmit Request (RTR) messages. The RTR mechanism allows a node on a CAN bus to request a transmission from another node on the bus.

This reference design models a two-node CAN network using a single chipKIT Pro MX7 processor board. The concept diagram shown in Figure 1 divides the operation as if it were implemented on two separate chipKIT Pro MX7 processor boards. In concept, Processor A functions as a remote indication of LED1 on Processor B. This remote indication is provided by LED4.

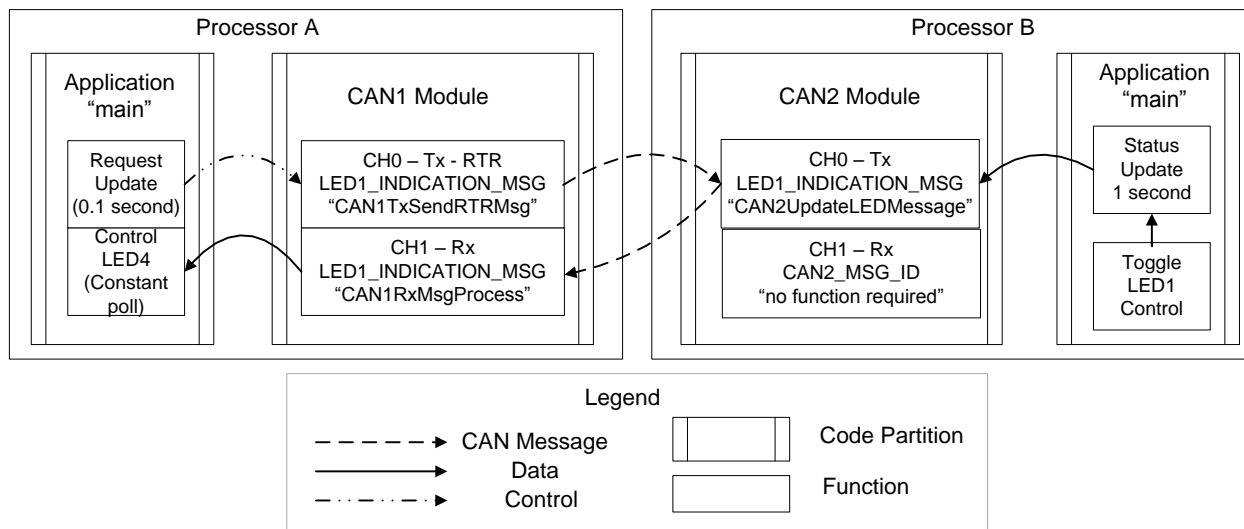


Figure 1. Concept diagram for the PIC32 CAN EID RTR Code Example.

Figure 2 shows the control flow diagram for the operations of the control, indication and instrumentation LEDs.

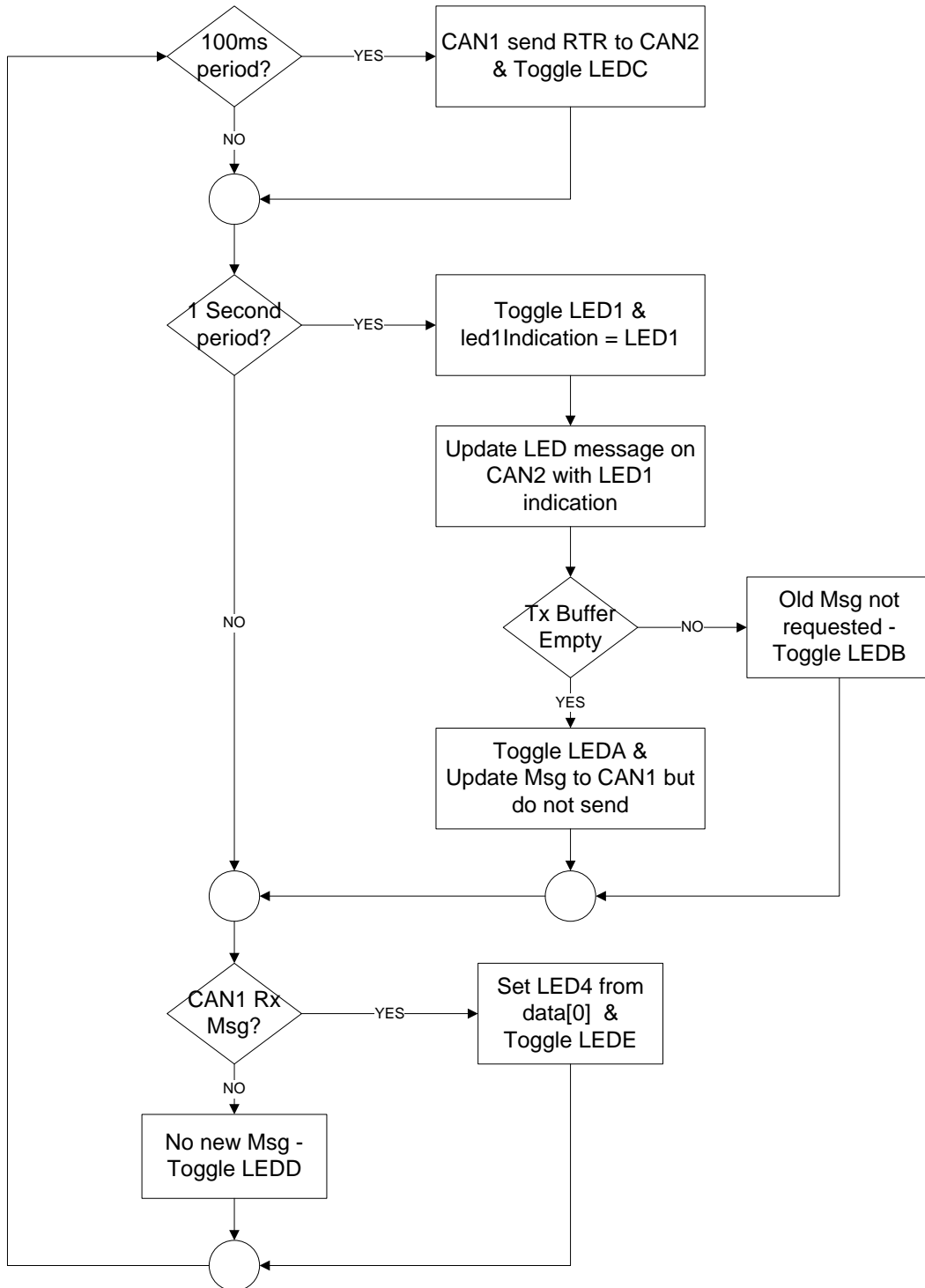


Figure 2. Control flow diagram of the LED controls.

All LED operations are defined in Table I. LED1 represents a control action and is changed on to off and off to on once each second. The state of this action is provided in the LED1_INDICATION_MSG and posted on the CAN2 Tx message channel once each tenth of a second. LED4 is the remote indication of the state of LED1. LED4 is updated when a CAN message is received in response to a RTR message by Processor A. Processor A continuously checks for a received LED1_INDICATION_MSG even though Processor A only implements an RTR message requests once each tenth of a second. The continuous checking for a CAN_INDICATION_MSG allow for Processor B to initiate an unsolicited CAN_INDICATION_MSG message.

LED	Function	Description
LED1	Function control	Process control local status
LED4	Control indication	Process control remote indication
LEDA	Instrumentation	CoS indicates new LED1 indication message posted
LEDB	Instrumentation	CoS indicates last posting was not requested
LEDC	Instrumentation	CoS indicates a RTR message was sent
LEDD	Instrumentation	CoS indicates RxMsgProcess had nothing to do
LEDE	Instrumentation	CoS indicates RxMsgProcess received message

LEDA through LEDE are used for timing instrumentation. Figure 3 shows the timing of the events following a new posting of a CAN_INDICATION_MSG.

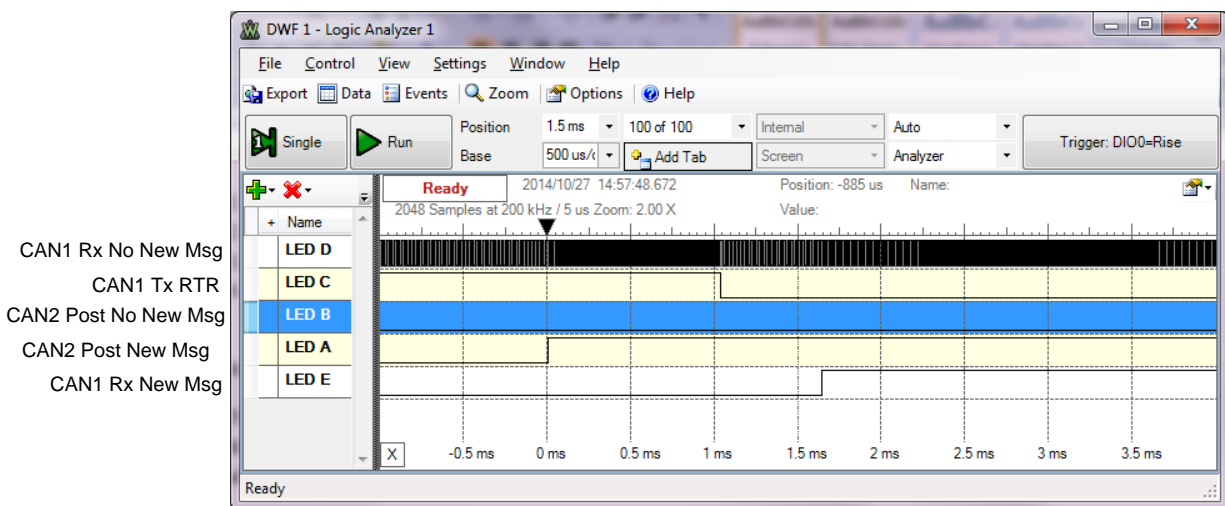


Figure 3. Timing diagram showing the sequence of updated message posted, RTR message requested, and updated message received.

An indication of a monitored event is indicated by a logic transition on the assigned trace. The top trace (LED D) shown in Figure 3 demonstrates that the code is constantly polling “CAN1RxMsgProcess” to check for a new CAN_INDICATION_MSG. Each time the top trace toggles, there was no CAN_INDICATION_MSG received. The LED A trace shows when a new LED1 indication message was posted in the CAN2 Tx buffer. The LED C trace shows that an RTR message was sent from CAAN1 to request the CAN_INDICATION_MSG. The positive transition of the LED E trace shows when CAN1

receives the CAN_INDICATION_MSG. There are no transitions on the LED B trace. This indicated that all posted messages had been requested. This is reasonable since the CAN_INDICATION_MSG is requested at a 100ms interval while updates are posted once each second. Figure 4 shows the events for the case when a message has been requested and the message has not been posted by the message generator. In this case there is no response message generated.

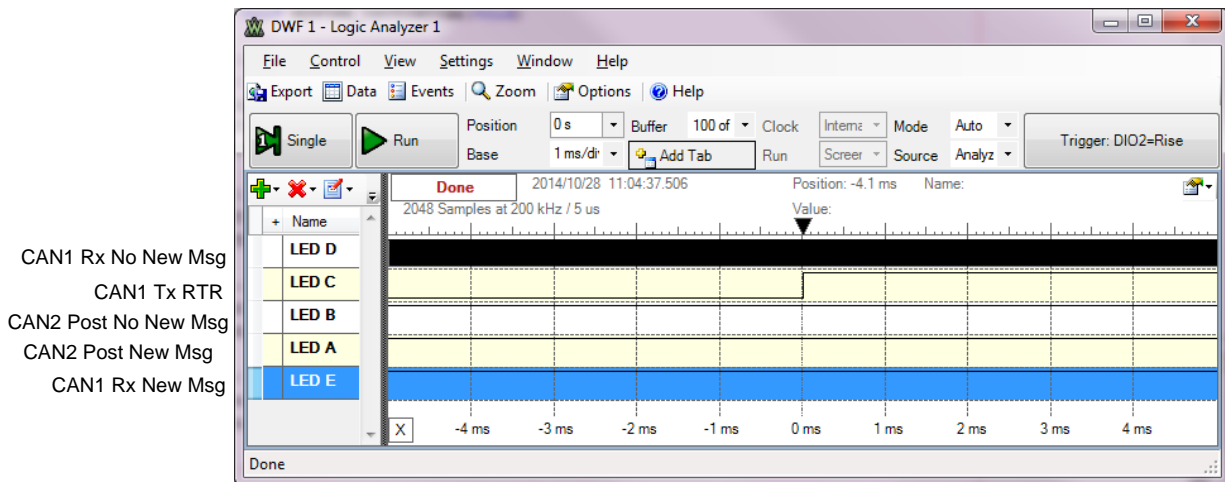


Figure 4. Event timing for a RTR message request with no update.

A possible alternative is set up the CAN2 TX channel to generate an interrupt on the CAN_TX_CHANNEL_EMPTY event. However, there is no guarantee that the data in the posted message is the most current. Further investigation is required. (RWW)