CAN Full Example Project
1.0

Features

- Configures Transmit and Receive mailboxes in Full CAN Mode

General Description

This example project demonstrates how to configure the CAN component to transmit and receive messages over the CAN bus in the Full CAN mode.

This is only one part of the CAN example project. Use this example along with CAN_Basic_Example for complete demonstration.

Development kit configuration

This example project is designed to be executed on CY8CKIT-001 from Cypress Semiconductor. A full description of the kit, along with more example programs and ordering information, can be found at http://www.cypress.com/?rID=37464.

Also, the example project requires a CY8CKIT-017 CAN/LIN Expansion Board kit. A full description of the kit, along with more example programs and ordering information, can be found at http://www.cypress.com/?rID=40215.

CY8CKIT-017 CAN/LIN Expansion Board kit should be connected to DVK1 PORT A 2x20 connector with installed jumpers: JP2 (CAN Termination Resistor), CANEXTPWR and JP6 set to Vdd-V5_0 (default setting for 5V operation). Any jumper on the board not mentioned above should have no jumper installed.

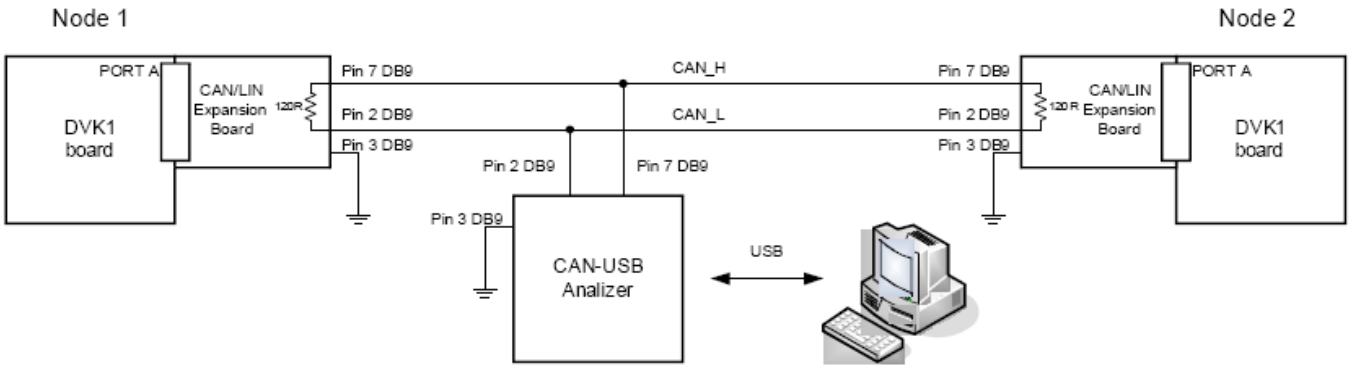
1. Build the project and program the hex file into the target device.
2. Power cycle the device and observe the results on the LCD.
3. A CAN – USB analyzer can be used to analyze the data traffic.

Project Configuration

The example project consists of the CAN and LCD components.

CAN received and transmitted data are displayed on the LCD. To ensure proper functioning of the examples projects, you should create a mini network from at least two CAN nodes as shown in Figure 1. The network as shown in Figure 1 also includes a CAN-USB analyzer to analyze the data traffic.

Figure 1. Test CAN Network Topology



Project Description

This example illustrates how to transmit and receive messages using the CAN component.

The CAN component is configured to receive the following messages from the remote node:

Message 1 - Status of Switch1.

Message 2 - ADC data.

The component is also configured to transmit data to control the pulse width of the PWM in the remote node. The transmitted data (pulse width value) increments at a switch press.

Both transmitted and received data are displayed on a 2x16 LCD.

Every 100ms ADC data measures by remote node receives over the CAN bus and displays on the LCD.

When the node receives Message 1 (with status of Switch1) it displays this status on the LCD and increments value of PWM pulse width by ten and sends back to the remote node updated PWM pulse width value along with displaying it on the LCD. Please note that PWM pulse width isn't being sent and displayed before first button connected to Switch1 press.

All transmitted and received data are doubled on LCDs for both nodes.

Expected Results

Program the device with the project and observe that the LCD in the first row displays the ADC data and PWM pulse width; in the second row – state of Switch1 (pressed or released).

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges. PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.