Using the Total Phase CAN/I2C Activity Board Pro as an I2C-to-CAN Translator

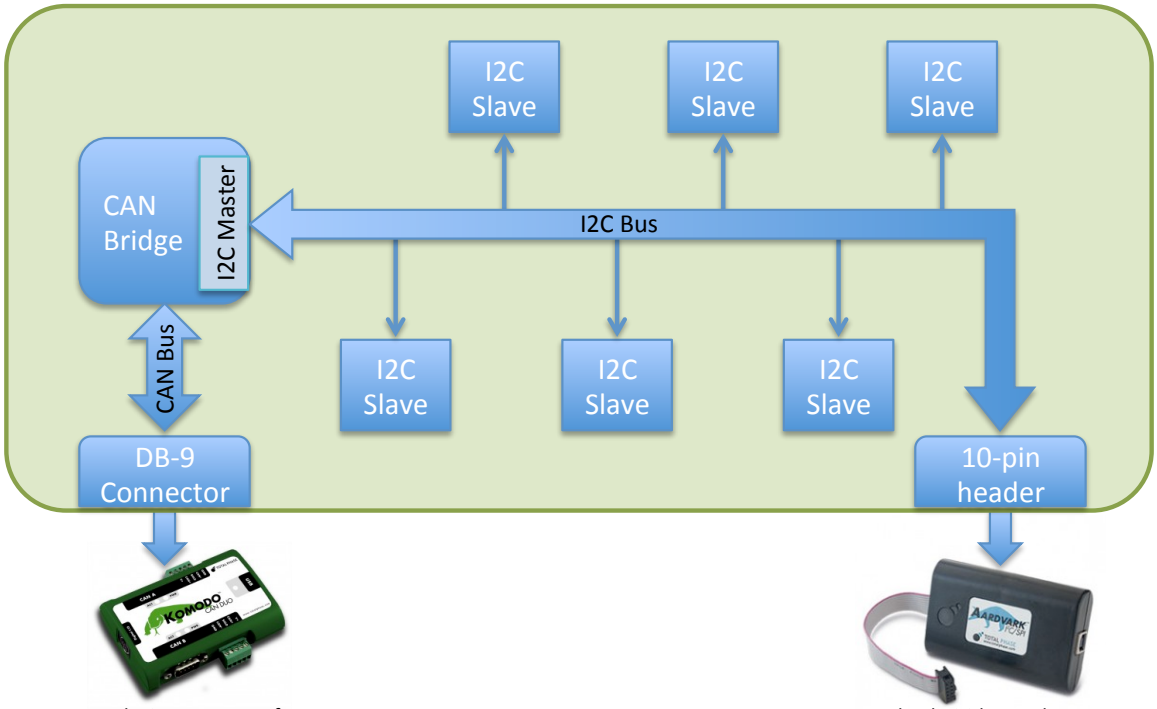
Application Brief by Rick Bogart

Introduction

A customer request came in asking if we had a CAN to I2C translator. My first thought was that we could solve that task with an Aardvark I2C Host Adapter and a Komodo CAN Duo Interface. Each could receive and send their respective bus commands and the translation could be done in the computer connecting the to adapters (via USB). But then after a little more thought I realized our CAN/I2C Activity Board Pro can support this function by itself, as it has a CAN bridge as part of its design.

The CAN/I2C Activity Board User Manual describes how the CAN bridge chip acts as an I2C Master to the I2C Slave devices on the board. (See section 1.4 Communication Protocol).

CAN/I2C Activity Board Pro



Komodo CAN Duo Interface

Aardvark I2C host Adapter

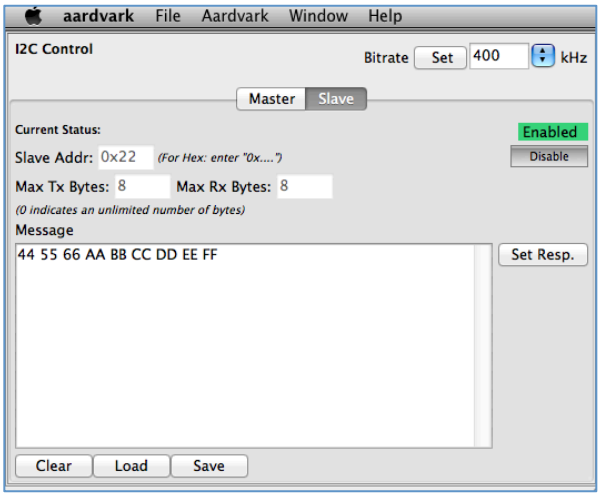
Scope

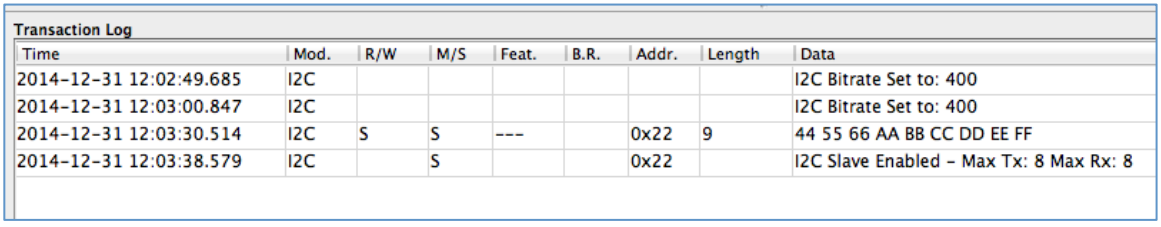
I decided to try the CAN/I2C translation with a Komodo interface for generating CAN commands and an Aardvark adapter to be an off-board I2C slave.

Using the Komodo GUI, I can generate CAN commands to send/receive commands and data to any of the I2C Slave devices. I wanted to use the Aardvark adapter as an off-board I2C Slave.

Using the Control Center Serial software GUI, I setup the Aardvark adapter as an I2C slave with response data. Here are step-by-step instructions:

- Select “Slave” mode
- Set Bitrate to 400 kHz
- Set Slave Addr to 0x22 (I picked an address which is not used on the Activity Board)
- Set Max Tx Bytes to 8, and Max Rx Bytes to 8
- Enter some random data in the Message window “44 55 66 AA BB CC DD EE FF”, and “Set Resp.”
- Select “Enabled”





Time	Mod.	R/W	M/S	Feat.	B.R.	Addr.	Length	Data
2014-12-31 12:02:49.685	I2C							I2C Bitrate Set to: 400
2014-12-31 12:03:00.847	I2C							I2C Bitrate Set to: 400
2014-12-31 12:03:30.514	I2C	S	S	---		0x22	9	44 55 66 AA BB CC DD EE FF
2014-12-31 12:03:38.579	I2C		S			0x22		I2C Slave Enabled - Max Tx: 8 Max Rx: 8

So if I can send a Read command to the I2C device address 0x22, I should be able to read back the data I entered in the Message window.

Now I used the Komodo GUI to send the CAN command.

The CAN/I2C Activity Board User Manual explains how to formulate the proper CAN command to communicate with the I2C devices. It is best to read this section first.

But, basically the CAN ID is made up of the 7-bit node address (I2C address), plus a 4-bit opcode. The opcodes are defined in Table 2 and 3 of the manual. Basically your choices are Read or Write, and with or without a Stop.

To perform an I2C Read with a Stop, the opcode is 1000 (see Table 3 of the CAN/I2C Activity Board User Manual). We've set the I2C address to 0x22 (7-bit address is 010 0010).

Put the opcode and the 7-bit I2C address together to get the CAN ID.

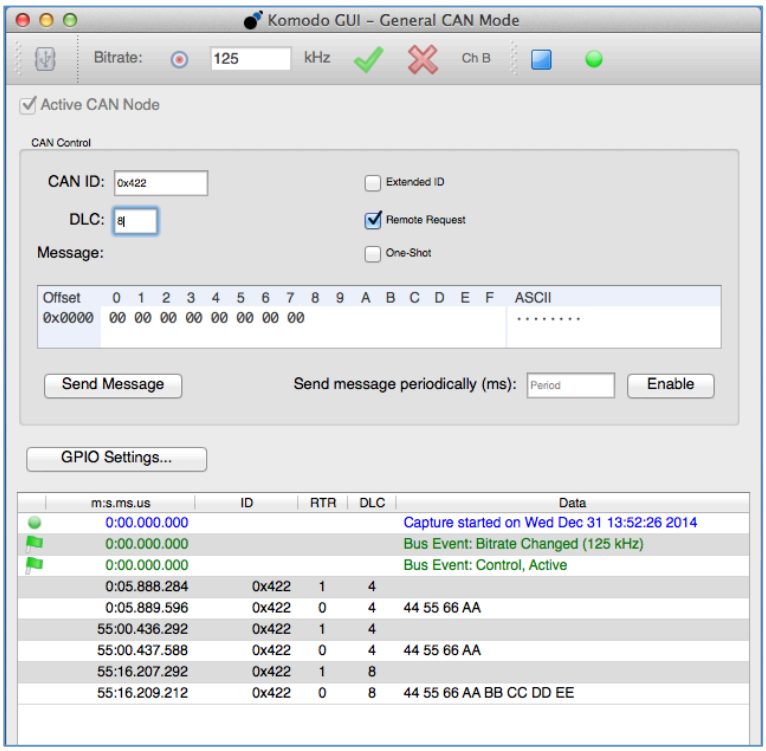
1000 010 0010 or 100 0010 0010

This in Hex is:

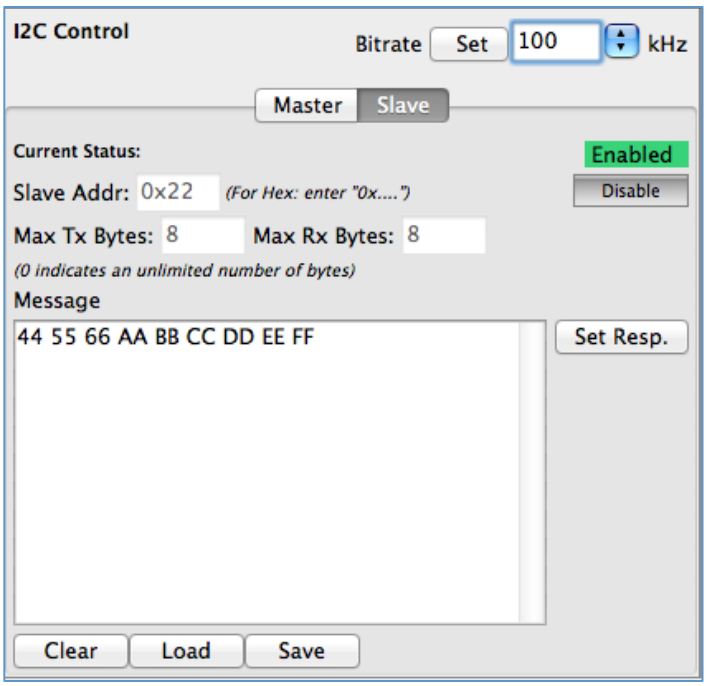
4 2 2 and becomes our CAN ID.

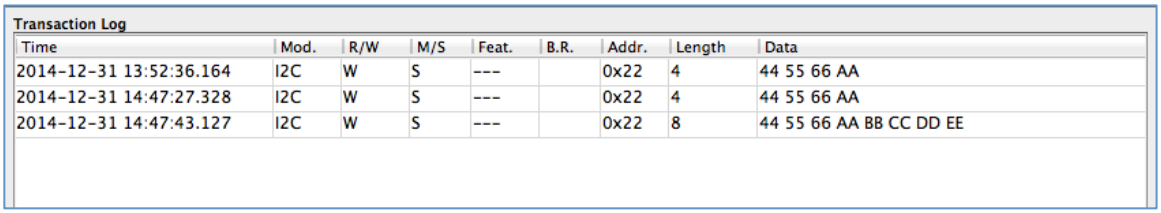
The DLC will be set to the number of bytes we want to read back from the slave. And the Remote Request (RTR) needs to be checked.

Then click "Send Message".



So by clicking Send Message, the Komodo interface sends the CAN message to the CAN bridge which translates it to the I2C bus, and on to the Aardvark adapter which is acting as an I2C Slave. The Aardvark adapter then responds with the number of bytes specified in the DLC. You can see three read requests in both the Control Center Serial and Komodo GUI windows. Two are requests for 4 bytes of data, and one is a request for 8 bytes of data.





Summary

So this is an example of having the Komodo interface send a CAN command to the CAN/I2C Activity Board, and having the Activity Board translate the message to I2C, and send it to the Aardvark adapter that is acting as an I2C Slave. The Slave then responds back, and the message is passed through the Activity Board and converted to a CAN message which is logged on the Komodo interface.

These tools are extremely useful and handy in simulating your environment before you completing your design. They can also be used for debugging your design as well.

About Total Phase

Total Phase manufactures user-friendly, powerful, and affordable USB, I2C, CAN and SPI development tools for embedded systems engineers. Total Phase tools combine best-of-breed features, Windows, Linux, and Mac OS X support, a royalty-free API, and excellent customer support, making them perfect additions to any engineer's toolbox.

Please visit us at http://www.totalphase.com

Copyright © 2015 Total Phase, Inc. All rights reserved.