# BMW I-BUS Project

I'm working on a little project to interface to the I-BUS in my E46. This project will go in between the radio head unit and the CD changer. It will intercept messages meant for the CD changer and only forward messages when the CD changer is selected. When Aux Audio is selected the CD Changer messages will be intercepted and my board will respond to the head unit like it was a CD changer. I've seen a few other devices out there, like the DICE, that allow you to hook an iPod (MP3 player) to the stock radio, but, they require you to disconnect the CD changer. My project will keep the CD changer. I intent to use the R/T steering wheel button meant for the cell phone to select between the MP3 player and the CD Changer. I am thinking about having multiple inputs that can be rolled through with each press of the R/T button. This will allow MP3 player, Satellite radio or HD Radio to be routed in through the stock radio.

At this moment I haven't picked a processor, but, it's going to be a Microchip PIC device with dual UARTs, probably something in the PIC24F series. I have picked an I-Bus interface chip and it's NOT the Melexis TH3122. I'll explain a bit later.

Below is picture of my I-Bus reader that I build just to see what's on the bus and if I can write code to capture the message and interpret them. I used an old board from work, a PIC16F886A and a 4-line LCD Display. I am able to capture and display in hex the IBUS messages. For example the top line is 0x50 0x04 0x68 0x3B 0x28 0x2F. The first byte is the source ID, in this case 0x50 which is the MFL or Steering wheel, the 2nd byte 0x04 is the length meaning there are 4 more byte in the message. The 3rd byte is, 0x68 which is the radio. The 4th byte and 5th byte are the message, in this case 0x3B and 0x28 which means Button Previous was released and the 6th byte is the checksum. So XOR'ing 0x50, 0x04, 0x68, 0x3B and 0x28 equal 0x2F, the checksum value. If you XOR the checksum with the rest of the message you get 0x00. Anything other than zero means the checksum doesn't match and the whole message should be ignored. I have two LEDs on the board, a red one and a green one. I have the red one blink everytime the checksum doesn't match and the green one blink when it does match. I also have a DIP switch to change the mode of the LEDs so that it only blinks when MFL to Radio messages are detected. I am most interested in steering wheel commands. The next command 50 05 68 3B 08 0F for Previous button-Pressed and then Previous button-Released again.
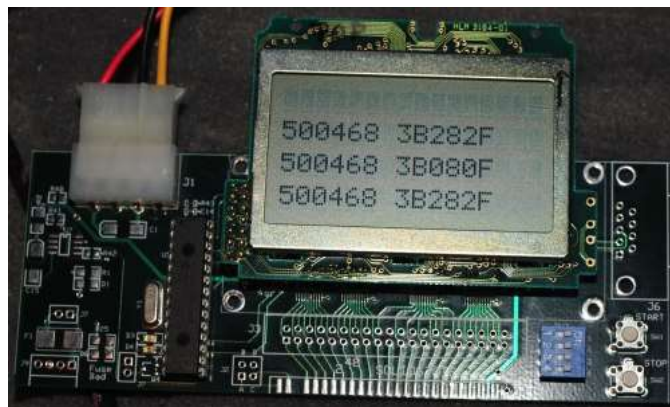
Below is the schematic minus most of the stuff on the board that I didn't use. Pretty standard, PIC microcontrollers, some buttons & switches, some LEDs and voltage regulator. The part you might find interesting is my I-Bus interface. I didn't use the Melexis TH3122 chip which everyone seems to think is the way to go. I looked at that chip and thought, why do I need all this, plus the chip is hard to get and 16 Euros from Rolf Reseler in Germany. Screw that, I can do it with discrete components. I found a three transistor circuit which did the job, but, I didn't like the fact that it drew current from the bus when the bus was high. In a multidrop bus like the I-Bus you should not draw current from the bus when it's floating and passively pulled up. Plus that circuit used two transistors for receive. The first transistor inverts the signal and the second transistor is there to invert it again to get the proper polarity. I thought about it some more and came up with the circuit below. One schottky diode and a pull up resistors, there only $0.16 worth of components. My circuit presents no load to the I-Bus when the bus is high even if my circuit is unpowered, except for minimal diode leakage. When the I-Bus is low the diode is forward biased and pulls the PIC's receive pin low. The 10K resistor pulls the line to +5V when the I-Bus is high. As you can see in the picture above I have built this circuit, written code and it works great. Now this circuit only sits on the bus and snoops. It does not transmit. To add transmit capability I just need to add one more resistor and a transistor. Of course the

microcontroller would then need to do collision detection and hold off, which is one benefit of the Melexis chip, but, I'm not scared of that. I like my circuit, but, for my next board I'm going to want transmit capability so I'm going to use the ON Semiconductor AMIS30600. It's in stock at Mouser for only $1.46, not half way around the world for 10 Euros plus 6 Euros shipping.
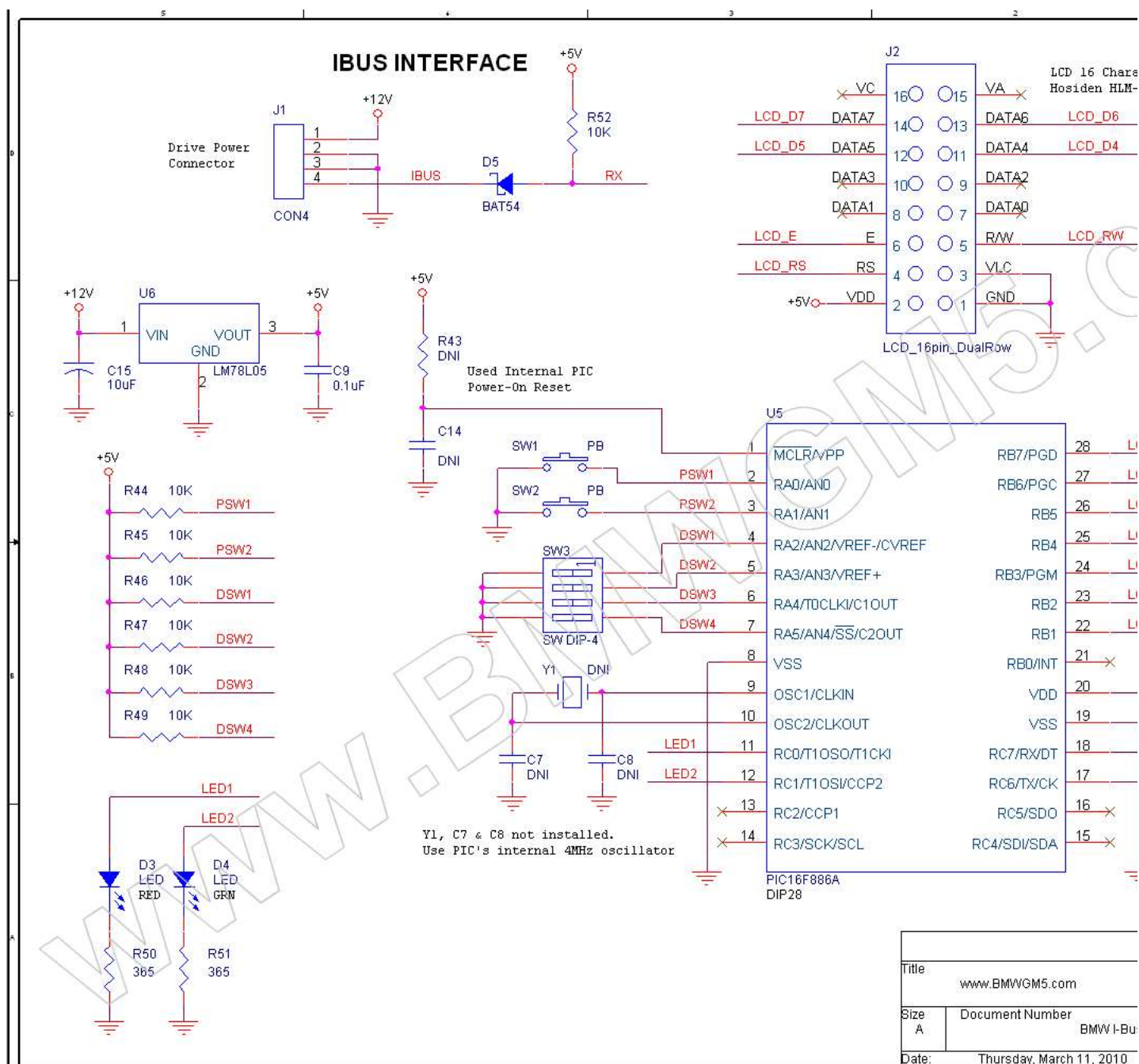
My project has taken many twists and turns. First I wanted to use the stock head unit to interface to multiple audio sources then I wanted to change the head unit to a Pioneer Premier head unit which I could control using the BMW steering wheel controls. Then back to using the stock head unit. Now I'm not sure what I want to do and the project is on hold while I work on some other projects. I might get back to this project when I get some time or when I decide what I want to do with my car audio. In the mean time I'm going to leave this web page up with some of my notes and earlier work.

My project has morphed into a Steering wheel control to Pioneer control project. The deal is I'm about to rip out the stock head unit and put in a Pioneer Premier DEH-P670MP head unit, but, I really want to have steering wheel controls. The Pioneer head unit comes with a CXC3173 wireless IR remote control with 13 buttons. I have mapped out those buttons into NEC Device and Buttons codes in the table below. The forth column the steering wheel IBUS message that will trigger it. For more on NEC IR protocol check out this data sheet on the Celadon web site. IR Protocols

Button	Device	Button	IBUS
Volume +	0xAD/0x52	0x0A/0xF5	50 04 68 32 11 1F(Vol+)
Volume -	0xAD/0x52	0x0B/0xF4	50 04 68 32 10 1E(Vol-)
Band	0xAD/0x52	0x12/0xED	
Mute	0xAD/0x52	0x0C/0xF3	50 04 68 3B 80 27(Dial)
Up	0xAD/0x52	0x40/0xBF	
Down	0xAD/0x52	0x41/0xBE	
Left	0xAD/0x52	0x42/0xBD	50 04 68 3B 08 0F (Prev)
Right	0xAD/0x52	0x43/0xBC	50 04 68 3B 01 06(Next)
Function	0xAD/0x52	0x19/0xE6	
Audio	0xAD/0x52	0x0D/0xF2	
CD	0xAD/0x52	0x1E/0xE1	
Pause	0xAD/0x52	0x58/0xA7	
Tuner	0xAD/0x52	0x1C/0xE3	

Regarding the IBUS message. IBUS is a 9600 baud Even parity, 1 stop bit serial communication channel. The first byte is the source address. In the case of 0x50 that means MFL or steering wheel. The second byte is the length of the rest of the message including checksum. The checksum is calculated by xor'ing the third byte through the last message byte, not including the checksum. The third byte is the destination address. In the case of 0x68 this is the Radio. The next two bytes are the message or button code. The sixth byte is the checksum as calculated by the sender. Use the receivers calculated checksum to verify that there was no error in transmission.

Button	IBUS Message (Hex)
Volume +	50 04 68 32 11 1F
Volume -	50 04 68 32 10 1E
Next (Pressed)	50 04 68 3B 01 06
Next (1 Second)	50 04 68 3B 11 16
Next (Released)	50 04 68 3B 21 26
Previous (Pressed)	50 04 68 3B 08 0F
Previous (1 Second)	50 04 68 3B 18 1F
Previous (Released)	50 04 68 3B 28 2F
R/T	50 03 C8 01 9A
Dial (Pressed)	50 04 C8 3B 80 27
Dial (1 Second)	50 04 C8 3B 90 37
Dial (Released)	50 04 C8 3B A0 07

Destination C8 is Telephone.

Now I find out that the back of my Pioneer head unit has a wired remote in which uses resistance to activate functions

Function	Resistance (Ohms)
Source (CD-Tuner)	0.4 - 2K
Toggle Mute	2.5 - 4.5K
List (Disp)*	5 - 6.5K
Seek-Up (hold-FF)	7 - 9K
Seek-Down (hold-Rew)	9.5 - 13K
Volume Up	13.5 - 18.5K
Volume Down	19 - 29K
Select (no effect)	33.6K
Mode (FM1, FM2, MW)	37.5 - 88K

* List(Disp) Changes the display in FM mode (frequency, station name, station type)
In MP3 mode it toggles between: track title, artist name, file name, folder name, etc.

http://www.mp3car.com/vbulletin/input-devices/91373-pioneer-wired-remote-control-4.html

The above resistance go from tip to ground.

It looks like a 50K pot should be about right. It should be capable of handling all functions. I'm going to try the Microchip MCP4019-50 which is a 50K, 128 step rheostat (fancy name for a pot with only two pins)

When ring is connected to ground (or below 2.4V) the Seek-Up and Seek-Down buttons change to select Next & Previous radio presets.

I have posted a couple web pages on my stereo install.

Front left door speaker location.

Factory BMW 10-channel Amp

Pinout of the Pioneer IP-Bus Connector
http://stereophonik.com/gutterslide/IP-BUS_Hack/

Cool web page on CD Changer Protocols
http://www.mictronics.de/?page=cdc_proto#Pioneer

Back to BMWGM5.com

Got door locks problems? Doors refuse to unlock? It could be your General Module. Check out my web site for more information. www.bmwgm5.com