K-BUS DATA READER PROJECT

This project was a continuation of the Automobile MP3 Player project. After getting the MP3 player working with the toggle switches, the next obvious step was to try to get the player tied in with the radio controls, specifically, the controls on the steering wheel. I thought it would be great to control functions such as scrolling thorough the songs, and traversing directories with these buttons and eliminating the bank of switches on the console that I added for the MP3 player.

While investigating how the steering wheel buttons were connected to the radio, I discovered the world of K-Bus. The K-Bus is a standard bus protocol for automobile electronics. In other words, it is a way for devices within a car to communicate with each other. On the BMW, there are several devices on this bus. These include the radio, the steering wheel radio button module, the CD changer, the light control module, etc. I was initially only really interested in how I could read the state of the steering wheel buttons, but I got a little caught up in the technology.

I read an article on www.openbmw.org that described some of the details of how the K-Bus (or I-bus as it is called in the case of the BMW bus). This got me started on the road to decoding the bus.

I started by researching electronics that would help me in building my K-Bus extractor. The device I selected was the Melexis TH3122 "K-Bus Transciever with Integrated Voltage Regulator". This part is as close to a perfect solution as you can get. It connects to the the K-Bus, Ground and the cars 12 volts and outputs 5 volts to power other components. The only drawback for the home hobbyist is that it only comes in a surface mount SOIC16 package. I was able to get a "SOIC to DIP" adaptor (don't remember the source right now).

The following is a schematic of the circuit that I put together to translate K-Bus to Serial RS-232. It is simple and most of the parts are fairly common (Digikey is a great source for electronics, but at the time of this writing, they did not carry the TH3122, although they do carry other Melexis parts). I acquired this part as a sample from the company.

Click in the box to get a .bmp image of the schematic.

Of course, you must use it at your own risk. I make no guarantees, and take no responsibility if you screw up your car!

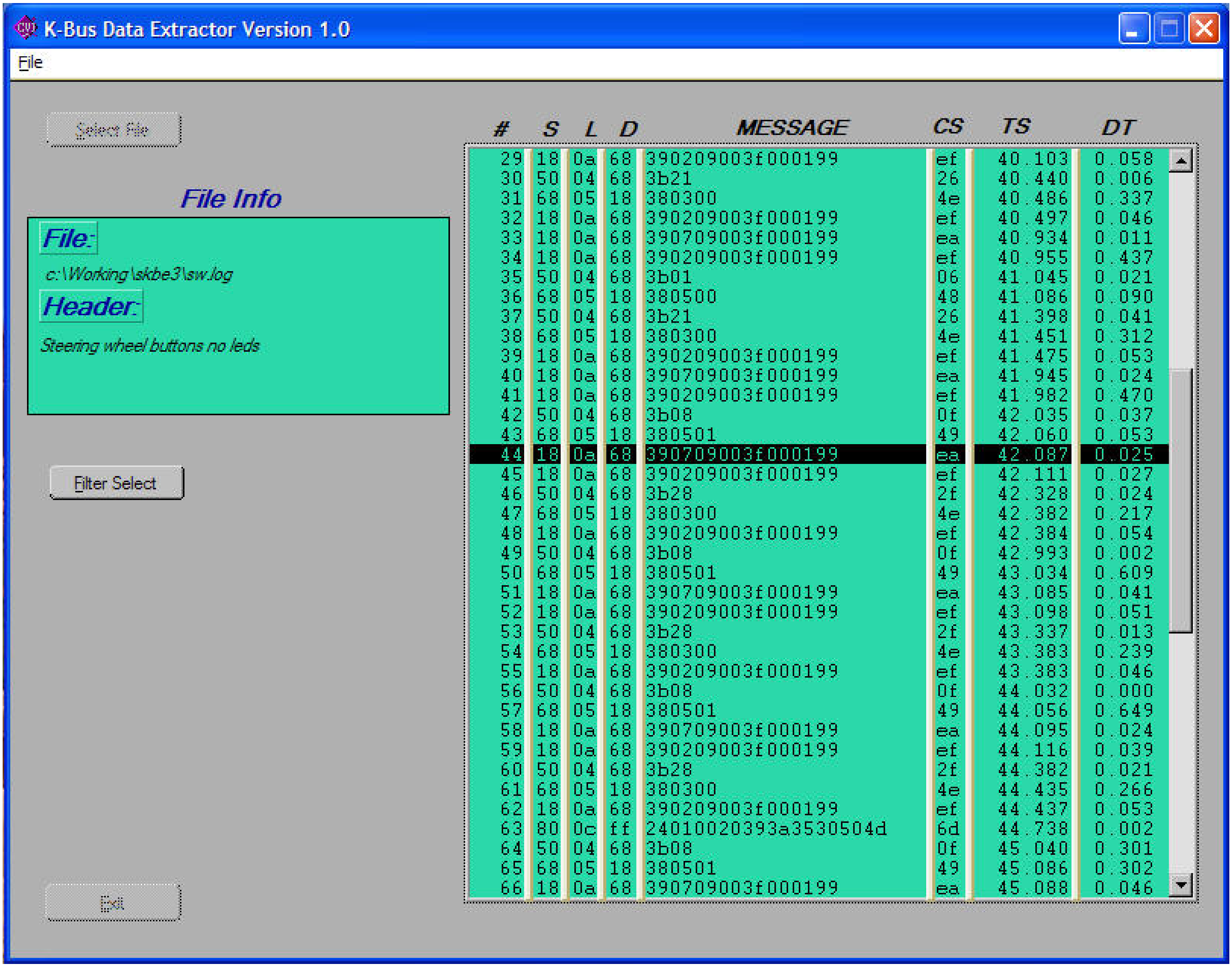
The pictures below show the assembled version of the K-Bus to Serial converter. I assembled it on a standard prototype board and soldered all connections. I suppose this could be done on a PC board from ExpressPCB (I used for the PIC Demo Board), but I was only making one. I tied into the K-Bus where the connection for the mobile phone is. (Under the center console). There was an article on the web about installing a mobile phone that gave the pin-outs for the connector.

XXX Add the following pictures: Connector end, circuit board in vehicle.

Assembled Circuit	Another View

The next obvious step was to start collecting data from the bus. The first attempt involved using the standard Windows HyperTerminal program. This was useful in verifying that I could collect data, but it doesn't do a very good job with data that is outside the standard ASCII character set. I had some strange results, but it was very useful initially.

Then I decided to write my own program that would capture data to a file for analysis later. I went through some attempts to use the time between messages to distinguish message from each other. This was somewhat successful and would have been enough to extract the information that I needed. But as I often do, I got obsessed with a particular project and decided to write a generic program to capture, timetag, and filter messages on the bus. The following screen shot shows the results of some data I captured on the bus.



File Info

File: c:\Working\skbe3\sw.log

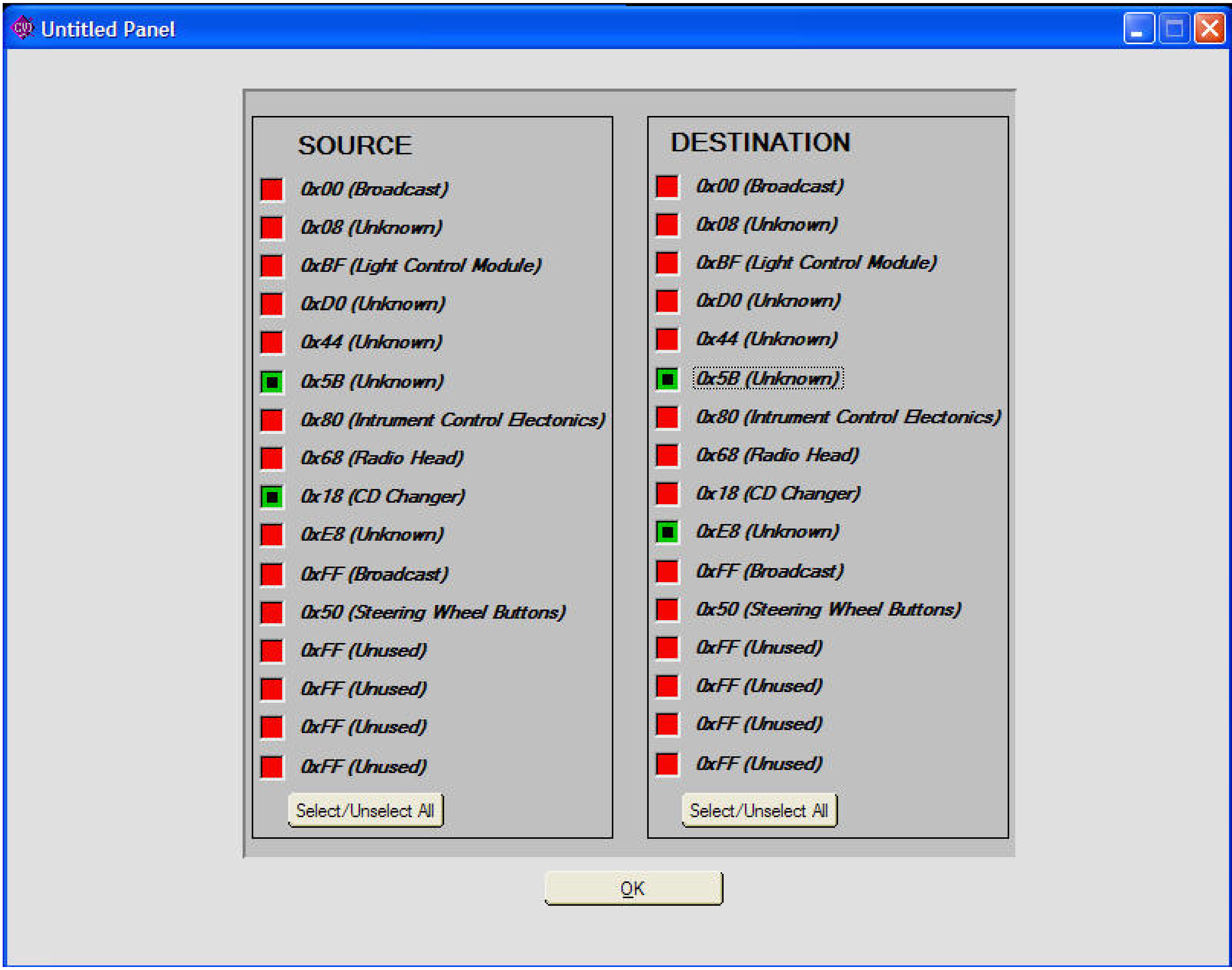
Header: Steering wheel buttons no leds

#	S	L	D	MESSAGE	CS	TS	DT
29	18	0a	68	390209003f000199	ef	40.103	0.058
30	50	04	68	3b21	26	40.440	0.006
31	68	05	18	380300	4e	40.486	0.337
32	18	0a	68	390209003f000199	ef	40.497	0.046
33	18	0a	68	390709003f000199	ea	40.934	0.011
34	18	0a	68	390209003f000199	ef	40.955	0.437
35	50	04	68	3b01	06	41.045	0.021
36	68	05	18	380500	48	41.086	0.090
37	50	04	68	3b21	26	41.398	0.041
38	68	05	18	380300	4e	41.451	0.312
39	18	0a	68	390209003f000199	ef	41.475	0.053
40	18	0a	68	390709003f000199	ea	41.945	0.024
41	18	0a	68	390209003f000199	ef	41.982	0.470
42	50	04	68	3b08	0f	42.035	0.037
43	68	05	18	380501	49	42.060	0.053
44	18	0a	68	390709003f000199	ea	42.087	0.025
45	18	0a	68	390209003f000199	ef	42.111	0.027
46	50	04	68	3b28	2f	42.328	0.024
47	68	05	18	380300	4e	42.382	0.217
48	18	0a	68	390209003f000199	ef	42.384	0.054
49	50	04	68	3b08	0f	42.993	0.002
50	68	05	18	380501	49	43.034	0.609
51	18	0a	68	390709003f000199	ea	43.085	0.041
52	18	0a	68	390209003f000199	ef	43.098	0.051
53	50	04	68	3b28	2f	43.337	0.013
54	68	05	18	380300	4e	43.383	0.239
55	18	0a	68	390209003f000199	ef	43.383	0.046
56	50	04	68	3b08	0f	44.032	0.000
57	68	05	18	380501	49	44.056	0.649
58	18	0a	68	390709003f000199	ea	44.095	0.024
59	18	0a	68	390209003f000199	ef	44.116	0.039
60	50	04	68	3b28	2f	44.382	0.021
61	68	05	18	380300	4e	44.435	0.266
62	18	0a	68	390209003f000199	ef	44.437	0.053
63	80	0c	ff	24010020393a3530504d	6d	44.738	0.002
64	50	04	68	3b08	0f	45.040	0.301
65	68	05	18	380501	49	45.086	0.302
66	18	0a	68	390709003f000199	ea	45.088	0.046

- S = Source Address
- L = Length Byte (Actually Length -2)
- D = Destination Address
- MESSAGE = Body of K-Bus message
- C = Checksum of message (XOR of all previous bytes)
- TS = Timestamp of message relative to start of program
- DT = Delta Time - Time between messages

The program will read back the stored data using a filter as shown below. The green means that a message with that attribute will be displayed, red means it will not.

I don't know what all the codes mean yet, I only know what codes showed up when recorded data. I got some info from a web article that I read but I don't remember where I got it from.



SOURCE

- 🟥 0x00 (Broadcast)
- 🟥 0x08 (Unknown)
- 🟥 0xBF (Light Control Module)
- 🟥 0xD0 (Unknown)
- 🟥 0x44 (Unknown)
- 🟩 0x5B (Unknown)
- 🟥 0x80 (Intrument Control Electonics)
- 🟥 0x68 (Radio Head)
- 🟩 0x18 (CD Changer)
- 🟥 0xE8 (Unknown)
- 🟥 0xFF (Broadcast)
- 🟥 0x50 (Steering Wheel Buttons)
- 🟥 0xFF (Unused)
- 🟥 0xFF (Unused)
- 🟥 0xFF (Unused)
- 🟥 0xFF (Unused)

DESTINATION

- 🟥 0x00 (Broadcast)
- 🟥 0x08 (Unknown)
- 🟥 0xBF (Light Control Module)
- 🟥 0xD0 (Unknown)
- 🟥 0x44 (Unknown)
- 🟩 0x5B (Unknown)
- 🟥 0x80 (Intrument Control Electonics)
- 🟥 0x68 (Radio Head)
- 🟥 0x18 (CD Changer)
- 🟩 0xE8 (Unknown)
- 🟥 0xFF (Broadcast)
- 🟥 0x50 (Steering Wheel Buttons)
- 🟥 0xFF (Unused)
- 🟥 0xFF (Unused)
- 🟥 0xFF (Unused)
- 🟥 0xFF (Unused)