**Debugging Electronics Bus**

1. **Get S&R Working with new COMS, CDH boards.**

* Last night I wasn’t able to get the CDH and COMS boards talking over CAN.
* This is probably an issue involving CAN resistors.
* I’m trying it again now to verify that it doesn’t work at the moment.
* The code on the OBC is in fact reaching the CAN handler, so it must be working right?

**Q:** Is the SSM receiving any CAN messages?

**A:** NO

* **Sometimes the OBC will stop responding over JTAG, the best thing to do here is just power cycle the OBC.**
* Time\_manage seems to be taking an overly long amount of time on the bus.
* Still not receiving anything over CAN
* Let’s try replacing his 180R with a 120R one.
* Replaced with a 110 ohm resistor
* Still nothing
* Resistance between CANH and CANL is currently 40 ohms
* I removed the 110 ohm resistor on the COMS PCB, so now the resistance between CANH and CANL is 60 ohms
* Sending from the IT5 CDH to an old SSM board doesn’t work.
* This leads me to think that there’s something wrong with CAN on the OBC side of things.
* Let’s try removing the CAN resistors on the OBC.
* **I still had can\_check\_general( ) commented out :/**

**NOTE: request\_sensor\_data( ) is not currently working.**

* Ok, it looks like the problem is mostly fixed now by inserting break statements into one of the case statements in the SSM. (for decode\_can\_command())

**Sending and receiving with new COMS and CDH works now.**

1. **Get all boards communicating over CAN using the new Motherboard.**

* Pin 75 = 3v3\_OBC.
* 77 = 3v3 for COMS
* 79, 80 = GND
* I’m currently just trying to program the COMS SSM while it’s plugged into the motherboard but I’m having a lot of difficulties.
* I suspect that the OBC is interfering with the programming lines somehow, perhaps I need to set them up as inputs when I’m not using them.
* These are pins 110, 109, 108
* Tried setting up the pins on the OBC as inputs so that they wouldn’t be driven.
* Still couldn’t get the device ID.
* Disconnected OBC
* Able to get device ID now
* Something wrong with connection between OBC and SSM in terms of the reprogramming lines.
* **I separated the reprogramming connections between the OBC and the other boards.**
* I can now read the device signature of the COMS and EPS boards from the PC104 header.
* Let’s try plugging in the payload board now.
* **Sending and receiving works with the new electronics bus :D**

1. **New Initialization / SPI code for SSMs, and payload science collection.**

* **Make sure that each function that uses slave selects drives it high at the end.**
* It does
* **Make a function that periodically checks to make sure that only one slave is selected at any given time.**
* **How to deal with COMS\_TEMP?**
* COMS only has one sensor, simply turn the SS of UHF off while getting temperature data, and then turn it back on at the end.
* **Port expander needs a different slave mode than almost everything else.**
* Port\_expander\_read/\_write save and restore the spi settings.

Rahman’s SPI Settings (02/16/2016):

|  |
| --- |
| void spi\_initialize\_master(void) |
|  |
|  | { |
|  | MCUCR = MCUCR & 0b01111111; |
|  | SPCR = 0b01011111; |
|  | return; |
|  | } |

* Port\_expander\_read and port\_expander\_write now both save and restore the SPI configuration now.
* **Need to set the data direction of each port depending on the SELF\_ID**
* **Done**
* Pins 25-28 = GPIOB0-3, 1-4 = GPIOB4-7
* Pins 17-21 = GPIOA0-4, 22-24 = GPIOA5-7
* EXP\_RST should be high or low? High for normal operation.
* EXP\_RST = pin 22 on the SSM

PORT EXPANDER 000:

* SS\_HUM = GPA4
* SS\_TEMP = GPA3
* SS\_ACCEL = GPA2
* SS\_PRESH = GPA1
* SS\_PRESL = GPA0
* **No other pins on port expander 000 are used**

PORT EXPANDER 001:

* Heater1-5 = GPIOB0-4
* Valve1a,1b = A0,1
* Valve2a,2b = A2,3
* Valve3a,3b = A4,5
* Valve4a,4b = A6,7
* Finished writing code for SS1\_set\_high( ) and SS1\_set\_low( )
* **I’m not able to get pressure values from the payload, it’s likely that there’s something wrong with the port expander code.**
* **I’m also not able to use UART on payload if TXD is an output on the EPS SSM.**
* **(Need these to be inputs on COMS, EPS)**
* **\*\*Can’t set this as an output when everything is put together because then the other SSMs aren’t reprogrammable.**
* **COMS SSM Needs 1500 KHz in order to program it (also make sure that the cable is nice and snug on both ends)**
* COMS is giving me trouble trying to program it from the laptop interface.
* Could try removing COMS RST button?
* MUST disable UART if you want to be able to program SSMs together.

**\*\* Did something to the recent SSM commit to fuck up COMS.**

**(currently on the commit when I got the RF front end working)**

**CAN communication with OBC still not working?**

**CANL seems to have been disconnected on the OBC board**

**Okay, take two:**

* I just noticed that the CAN transceiver on the payload SSM is backwards.
* Alright, I soldered back on the CAN resistors for the OBC and COMS and flipped the transceiver on the payload PCB so that it’s right side up.
* **Let’s try to get CAN communication working between OBC and COMS now.**

**Remember: To validate that a particular configuration works, first test that each SSM on the bus is programmable, then test to make sure that the sending and receiving program with the groundstation still functions normally (both housekeeping packets and normal response packets are received by the groundstation indicating full duplex communication)**

* Radio communication appears functional
* Still not receiving any packets from the satellite.
* **Code reached store\_current\_tc() indicating that a telecommand was received from the COMS SSM**
* Let’s try modifying the programs that are running and trying again (particularly, I noticed that getTickCount() disables interrupts which could (in theory) fuck with receiving CAN messages.
* I have time\_manage and coms running at the same time as opr and hk
* Not sure why but sometimes the groundstation will go a long time without receiving an ACK from the satellite.
* The COMS SSM is not receiving CAN messages sent by the OBC.
* Let’s investigate pins on the OBC involved in CAN.
* CAN0 looks fine unless there’s something wrong with that transceiver.
* Currently I can’t read the device ID for the payload if I only plug in COMS and PAY into the OBC
* Is CANRX connected on the COMS CAN transceiver?
* Yes
* I’m going to try using the old CDH board.
* Still nothing received
* Did I fuck up the port which manages CAN on the COMS SSM?
* I did, but correcting the code doesn’t seem to have fixed my problems.
* Somehow can\_check\_general() got commented out again.
* **I’m able to downlink packets pretty successfully now (housekeeping) but I’m not really getting any telecommands from the COMS SSM.**
* **Need to exit debugging mode to allow enough time for full duplex communication**
* **Works now.**
* Merging with master broke PUS communications
* Looking at the COMS pin definitions
* Reset is currently high which is good.
* DDRD = 0x6F doesn’t make any sense but I can make it this way for now.
* (Should be 0x63)
* COMS\_TEMP\_SS changed but it was incorrect before. (PC4 now)
* **It seems that the COMS SSM will only communicate over radio if it’s alternate SPI pins are set as outputs.**
* **Rule: COMS is the only one that gets to do this, must turn this off if you want to communicate with the other SSMs.**
* I plugged in EPS so that all SSMs are now connected, and I’m still getting communication with the satellite over radio.
* Next: Test to make sure all SSMs are programmable (when comms is turned off on the COMS board)
* All 3 SSMs are programmable from the laptop interface ☺