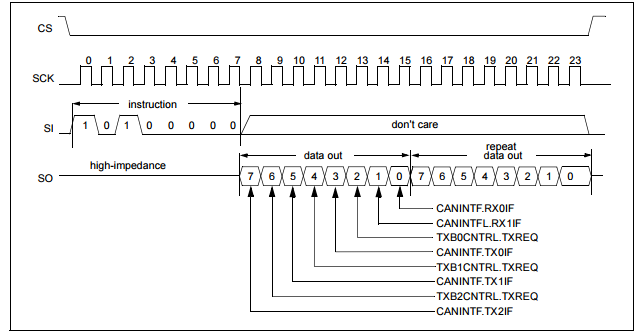
**Why is the CAN Shield not working?**

* I’m currently working with a brand new CAN shield and it’s not operating like it used to.
* Can\_send\_queue.isEmpty() always seems to be true.
* Currently trying to revert to a previous version of the code.
* Checked out: “[**Fixed issue with reaceiving of CAN messages**](https://github.com/UTAT-SpaceSystems/CDH-LaptopInterface/commit/ace69a4e4f592db20d0bb089f5ee1a16347f47e0)” and still no luck.
* I’m getting @MSG\_ERR over and over again.
* I’m sending to ID=32, and an 8-byte message.
* Q: **Why am I only providing CAN.sendMsgBuf only 4 variables when it takes 5?**
* /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* \*\* Function name: sendMsgBuf
* \*\* Descriptions: send buf
* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
* INT8U MCP\_CAN::sendMsgBuf(INT32U id, INT8U ext, INT8U rtr, INT8U len, INT8U \*buf)
* {
* setMsg(id, ext, len, rtr, buf);
* return sendMsg();

}

* **What is ext, rtr?**
* **Rtr = remote transmission request**
* **Ext = extended flag for a longer ID.**
* Okay, so I added in the extra attribute.
* Still getting @MSG\_ERR over and over again
* I’m still seeing the same thing as when I was working with Steven, which is that the first 3 messages fail as SEND\_MSG\_TIMEOUT and then all the subsequent messages fail as TX\_BUF\_TIMEOUT.
* Q: Are the headers on the CAN shield connected properly?
* Probably
* **I switched to a different Arduino and I’m still getting the same issues as before.**
* **Revert the code back even further…**
* Even code from before the merge with my transceiver code doesn’t seem to work.
* **I may need to hook this thing up on the oscilloscope.**
* **Q:** What is the point of the send queue?
* **Code is not entering void loop() now?**??
* For some reason I can’t seem to overwrite the program on the UNO.
* **Forgot make sure that I was using the external editor**
* **I sent a single message properly** and then the subsequent messages followed the same pattern as above which was 3 sending timeouts followed by tx timeouts forever.
* 100 ms delay: same thing happens.
* 250 ms delay: same thing happens.
* **Try resetting the chip after every message?**
* **Q: Is the message being received by the SSM?**
* **Actually never mind, I don’t ever receive a message properly, still the same thing that I had in the beginning.**
* Turned DEBUG\_MODE on for the mcp\_can code.
* Not much extra information here.
* Something to do with the rtr, ext flags?
* Tried running the example program, still no dice.
* Tried increasing the timeout value
* **Was sending ever able to work? If I can get receive to work, that would be good enough.**
* I added a line to the SSM code so that it would transmit often, still no luck receiving anything from the can-bus shield.
* There might be something wrong with the SSM/ wires connected to it?
* Try electronics bus:
* Still nothing.
* **Interesting:** When I connect the can-bus shield to the OBC, it can’t receive any CAN messages.
* Finally got a message while the CBS was plugged in, didn’t see anything received on the CBS side of things.
* **I’m receiving something??**
* How the hell am I getting 315 out of an 8-bit variable?
* LOL, wasn’t using the instances.
* **Still not receiving anything ☹**
* Connecting to the other CBS and a different Arduino didn’t seem to change anything.
* 
* When we read the status from the MCP2515, we get back:
* 0x54 = 01010100
* All the TX REQ bits are 1 meaning that they’re waiting to transmit…
* Make sure the initialization was done correctly.
* Tried slowing spi speed down to 200kHz.
* **I GOT SOME SHIT**
* **External SSM was not playing nicely.**
* **Electronics bus is a fucking beauty.**
* 1 MHz also works, it seems like 2MHz was to high
* Let’s check that.
* 2 MHz also works now…
* Whatever, works now.
* **When running the cabus\_gui, sometimes the program just quits for no apparent reason.**
* Including a “I’m Here” message seems to keep the communication going between the GS and GUI longer, but it still crashes.
* Q: Does it crash when I go to look at the plots?
* Looks fine to me, actually.
* Let’s go back to the master commit now.
* **Decidedly, the changes he made recently are preventing request\_sensor\_data() from being called.**
* Boundaries are located at line