CC3501 weekly report example

Group number: 2
Team members: Hunter Kruger-Ilingworth, Thomas Mehes, Quentin Bouet
Week number: 7

Progress this week:

Task	Who did it?	What were the outcomes?	Who did the peer review?	What did you learn?
Load cell testing	Hunter	Getting an appropriately scaled analogue signal from the loadcell apparatus from the researcher (0-3.3V). This was done using an instrumentation amplifier breakout board from the EE3901 subject. The layout for this part of the schematic was added to the Altium project.	Quentin	Confirmed that a circuit layout was defined for the loadcell. Viewed working loadcell on oscilloscope.
Testing SDI-12 sensor with RS485	Quentin	Tested RS485 with oscilloscope and SDI-12 sensor. Wrote a simple program to send SDI-12 breaks and commands. Built circuit on breadboard with pico. Added ability to receive response from sensor in the code.	Thomas	Oscilloscope confirmed “break” was working correctly with uart_set_break() where it waited for >12 ms before sending message. Confirmed with putty that SDI-12 sensor responses can be obtained.
Added Transceiver to schematic	Thomas	RS485 Transceiver circuit was routed in the schematic. It was based off the working breakout board we have https://core-electronics.com.au/ttl-uart-to-rs485-converter-module.html A voltage divider is included to ensure output to the microcontroller is 3.3V not 5V	Hunter	Corrected the ports on the schematic. Updated the block diagram of the system to the report.
“Conceptional progress”	Laurance	We were informed of many problems, a few solutions and importantly pointed in the direction of progress.	Hunter, Quentin, Thomas	SDI-12 has inverted bit values to UART, RP2040 voltage range isn’t large

				enough to send SDI-12 on its own.

Overall project tracking:

Week number	**Milestones**
4	Confirm project topic
5	Begin Overview and planning
6	Hardware design: Microcontroller, DAC, SD card, flash and usb interface
7	Hardware design: Voltage regulators, loadcell circuit layout and testing, SDI-12 testing and interfacing and Informal check with Laurance
8	Hardware design: finish PCB layout and review to make sure all design rules pass. Submit draft schematic to Laurance for review. Begin Software
9	Implement fixes to the PCB. Final PCB design submitted on Friday to Terence
LR	Software: Begin development that doesn't require hardware testing
10	Hardware: Solder components to PCB and begin interfacing Software: Coding to receive data from SDI12 sensors & I^2C DAC
11	Software: data logging applications including averaging, variable sampling periods and clean exported data.
12	Verify all hardware functionality, perform testing of existing software on the physical board. Polish the software.
13	Implement final bug fixes. Write the report. Demo day during Friday lab.