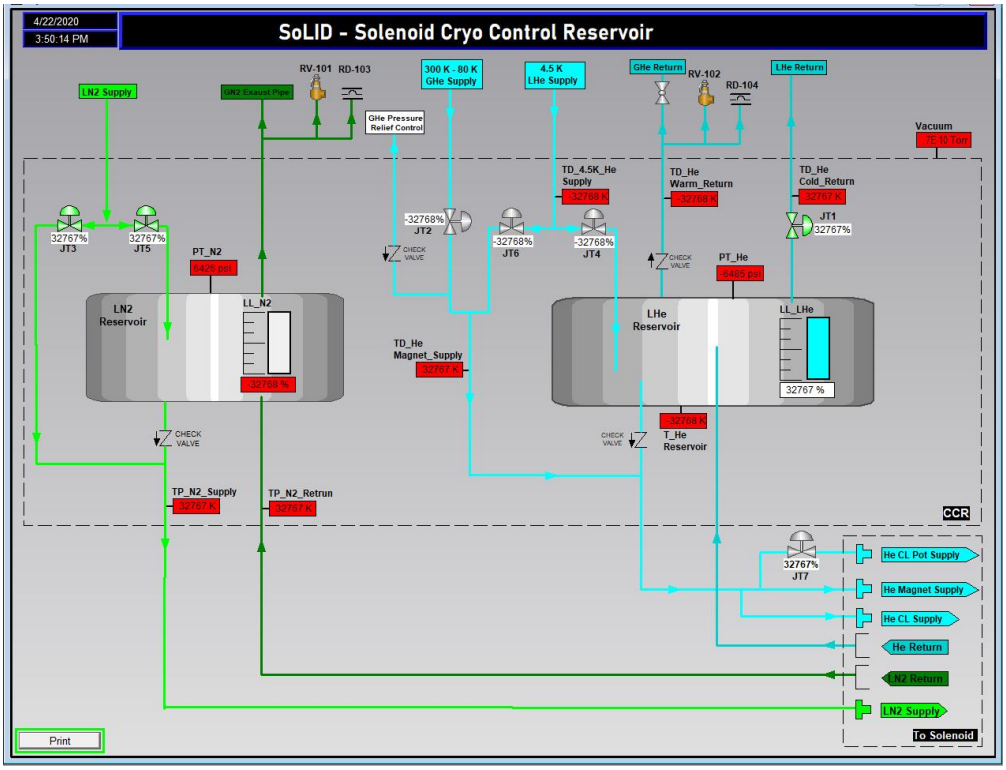
Detector Support Group
Weekly Report, 2020-04-22

Summary

Hall A – SoLID Magnet Controls

Mary Ann Antonioli, Aaron Brown, Pablo Campero, Brian Eng

- Generated AutoCAD drawings:
  - ✶ A00000-16-03-1150 – Linear Voltage Differential Transducers
  - ✶ A00000-16-03-2450 – Cable Diagram for JT Valves
- Completed sketch drawing for the readout and control of the JT Valves
- Developing database to store controls and instrumentation drawings for SoLID
  - ✶ Drawings are databased as paths to the PDF file
- Developed PLC subroutine to read temperature sensors in Cryo Control Reservoir
  - ✶ Subroutine also checks sensors for errors based on their limit and fault readouts
- Developing PLC code to monitor liquid levels inside LN_2 and LHe reservoirs
- Completed *Cryo Control Reservoir* HMI screen
  - ✶ Added temperature sensors and vacuum alarm animations to show sensors readout errors and over/under limit.



SoLID Solenoid Cryo Control Reservoir HMI screen. Sensors are not connected to the PLC system, hence the red indicators.

Hall A – SBS GEM

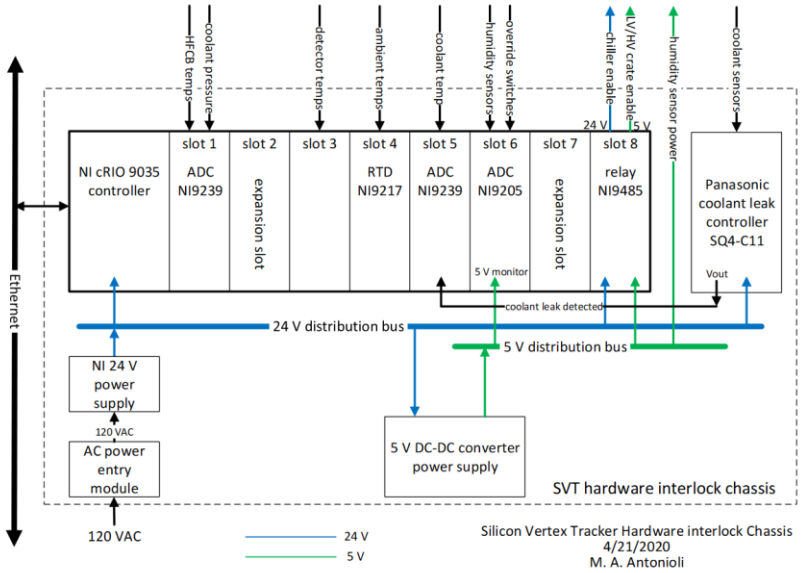
Brian Eng, Marc McMullen

- Generated schematic/notes on I^2C multiplexer board to for its CAD design
- Ordered DSG Gas Flow Sensor PCB; delivery expected in ~one week.

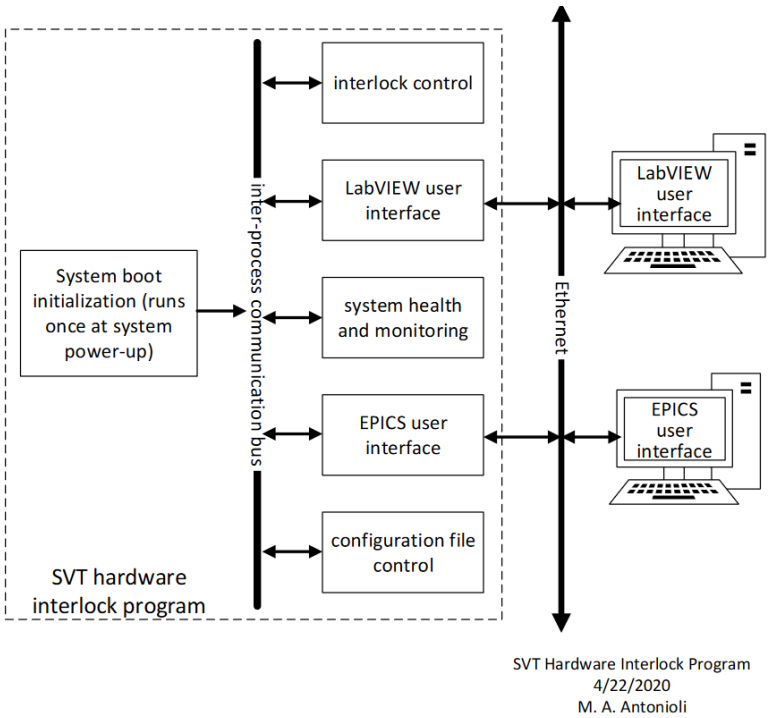
Hall B – SVT

Mary Ann Antonioli, Peter Bonneau, Marc McMullen

- Received Interlock System cable disconnect parts
- Generated technical documentation for Hardware Interlock System



System diagram of SVT Hardware Interlock System



Network and Processes diagram of SVT Hardware Interlock System

Hall C – CS-Studio Screen (CSS) Development

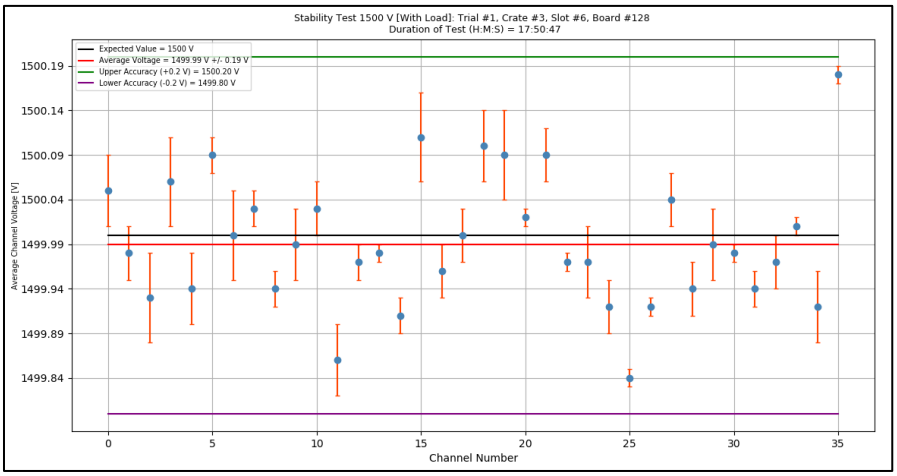
Mary Ann Antonioli, Aaron Brown, Pablo Campero, Tyler Lemon

- Developing *HMS Cryo* and *HMS Q1 PSU Setup* screens
- Developed Python program that runs from a CSS screen that creates test versions of converted screens with local PVs.
- Developing database to store completed CSS-BOY screens
- Wrote DSG note detailing CSS screen development; note posted to DSG website

Hall C – CAEN Testing

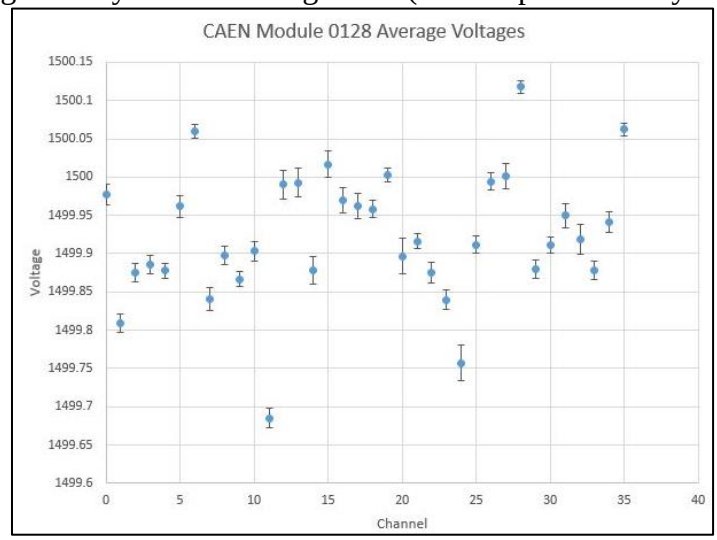
Aaron Brown, George Jacobs

- Analyzing stability test data using Python
  - ✶ Plots below shows average channel voltages with load for Board #128
  - ✶ Error bars in plots are standard deviation for that channel during test



Python plot of each channels' average voltage during stability test.

- Analyzing stability test data using Excel (for comparison to Python results)



Excel plot of each channels' average voltage during stability test.

HDice – fsNMR

Peter Bonneau, Tyler Lemon

- Completed development of Stage 2 of fsNMR program
- Wrote DSG note detailing fsNMR development; note posted to DSG website

Engineering

Mindy Leffel

- Populated one BPM boards (total of eight of ten completed)

DSG R&D – EPICS Data Logger

Aaron Brown

- Developing R program to plot data directly from database
- Debugging issues with data's timestamp
  - ✶ There is a 4 hour time offset when querying the database using R
  - ✶ Changing the session time zone to EST instead of UTC had no effect

DSG R&D – RICH

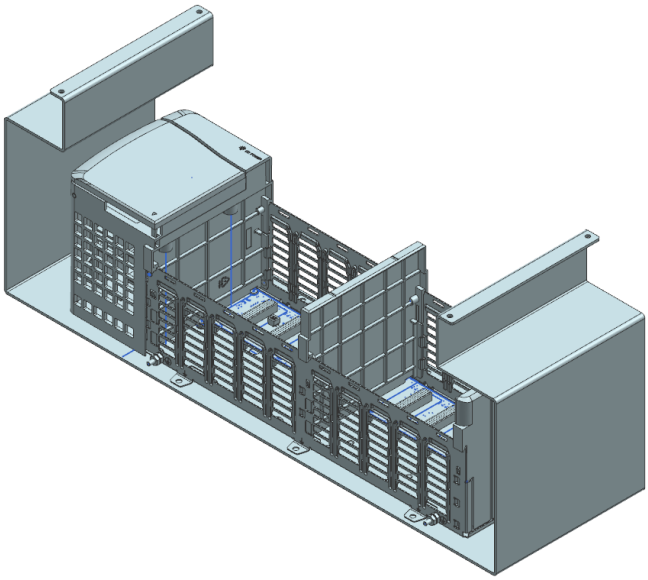
Peter Bonneau

- Researched new Sensirion SHT35-DIS-F2.5KS integrated temperature & humidity sensor. As compared to the SHT85 sensor, the SHT35 has:
  - ✶ Same accuracy: +/-1.5% RH and +/-0.1°C
  - ✶ Smaller size: 2.5 x 2.5mm^2 foot print in a surface mount package
  - ✶ Enhanced signal processing with two distinctive user-selectable I^{2}C addresses

DSG R&D – NX11 Modeling

Brian Eng

- Creating PLC rack mount in NX 11 using its Sheet Metal program.



NX11-modeled PLC bracket with PLC model to show how PLC will fit in bracket.

DSG R&D – MSELV Chassis

Peter Bonneau, Tyler Lemon, Marc McMullen

- Repositioned RMC components and optimized trace routing.
- RMC review completed
- Tested Raspberry Pi's signal generation using different GPIO (general purpose input/output) libraries to investigate whether it could be used as chassis controller
  - ✶ Timing of generated square wave measured using oscilloscope with average frequency of signal calculated over 1000 samples.
  - ✶ Python GPIO library's timing for frequencies greater than 100 Hz is inconsistent
  - ✶ *pigpio* library can consistently generate correct signals up to 500 kHz.
    - Issues at 1 MHz may be due to data type of input to program that generates signal.

	Python GPIO Library		*pigpio Library*	
Set Frequency	**Average**	**Std Dev**	**Average**	**Std Dev**
1 Hz	0.9987 Hz	0.0002205 Hz	1 Hz	0 Hz
10 Hz	9.967 Hz	0.01135 Hz	10 Hz	0 Hz
100 Hz	**97.53 Hz**	**0.1921 Hz**	100 Hz	0.001179 Hz
1 kHz	0.8264 kHz	0.003415 kHz	1 kHz	0.0001819 kHz
10 kHz	3.559 kHz	0.005209 kHz	10 kHz	0.009437 kHz
100 kHz	5.176 kHz	0.146 kHz	**100.5 kHz**	**1.026 kHz**
1 MHz	0.005482 MHz	0.0003155 MHz	2.107 MHz	0.032010 MHz

Table containing results of Raspberry Pi signal generation tests using Python GPIO and *pigpio*