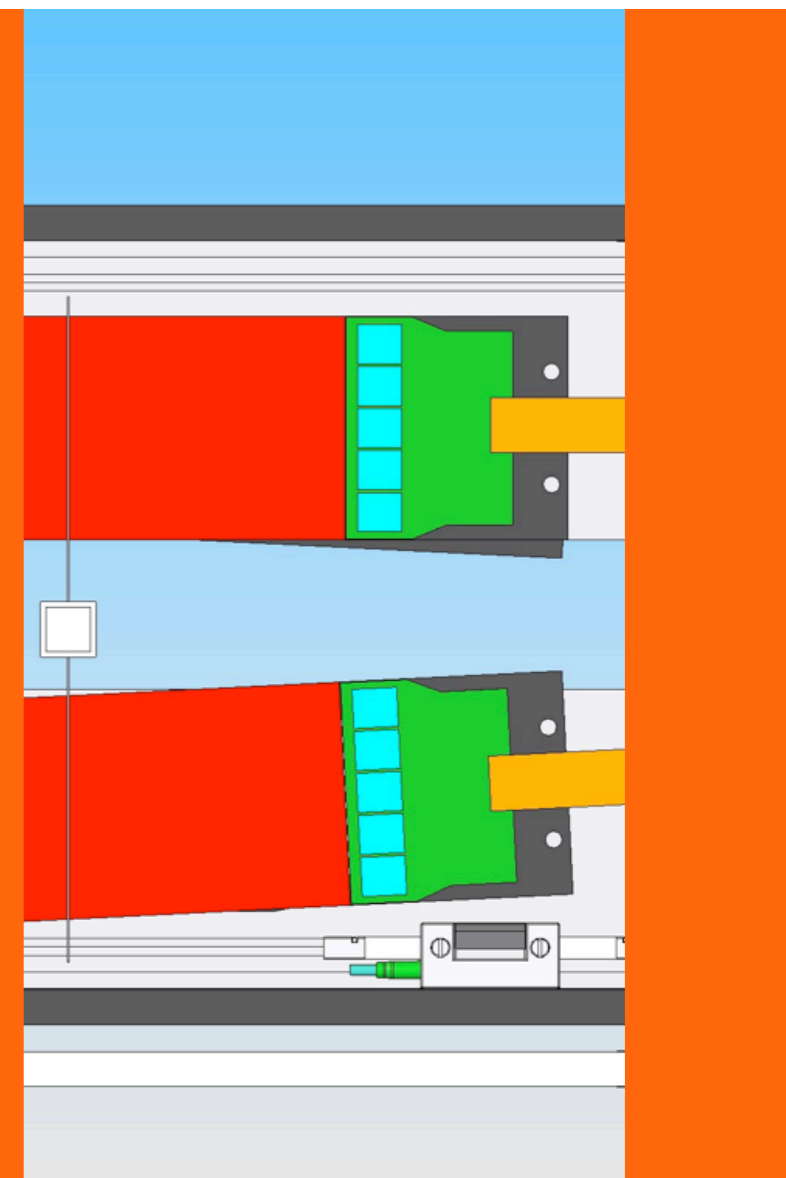# HPS Si Tracker Software: Monitoring, Control, Calibration and Conditions

Tim Nelson - **SLAC**

HPS Software Workshop - JLab

10/17/2011

Required Monitoring and *Control*

- Power: voltages and currents controlled/monitored though CAEN SY527 (RS232)

  - Voltage and current for both HV, LV for 20 Sensors: 80 values

  - Should be easy to integrate with EPICS based monitoring (also used at CDF)

- Cooling:

  - Hybrids have up to four temperature inputs.

  - Anticipate four per *module* (2 hybrids + 2 blocks): 40 values

  - Temperature data can be extracted from SVT datastream (can define with Ryan)

  - Also, $T_{chiller}$, T_{input} / T_{output} of cooling manifolds: 3 values $\Big\}$ Not clear what hardware looks like

  - Dewpoint monitoring?

- Control/Monitoring of SVT positions: 4 values (hardware not defined yet)

- Beam conditions: interlock SVT power on beamline monitoring (is there really software?)

- Monitoring plots: EVIO, EVIO to LCIO + simple driver

Required Calibrations and Conditions

- Will be regular runs that use internal calibration generator on APV25

    - Pedestal, gain, noise, t_0 for each channel: 4 * 12800 channels

    - These will define "calibration periods" that need different constants

- APV25 / DAQ configuration: all APV25 / DAQ settings for each run

- We expect internal alignment to be stable, however:

    - We have so many tracks that it should be possible to determine full alignments of any single run with small errors

    - 20 * 6 constants required: 120 alignment constants. Bow constants? (at most 2)

    - Online monitoring plots will need to use some assumption about tracker position. Will need conditions system designed to serve special "online calibration" data

- Trigger time offset: global timing offset between ECal trigger and SVT for each run

How does all this happen?

- Power/Current monitoring should be relatively easy.

- Someone on DAQ side will need to format SVT temperature data for input to EPICS. Other temperature data probably comes from JLab hardware.

- SVT calibration data is the big consumer

  - Need to define calibration procedure flag calibration runs with a "run type" and generate code to analyze the calibrations and produce constants that can be stored in the conditions/calibration database.

  - Same for alignment, but work on this is well underway thanks to Matt.

  - What kind or run/analysis is required to determine trigger time offset?

Bottom line: this all looks manageable, but there are things to think about, work to do!!