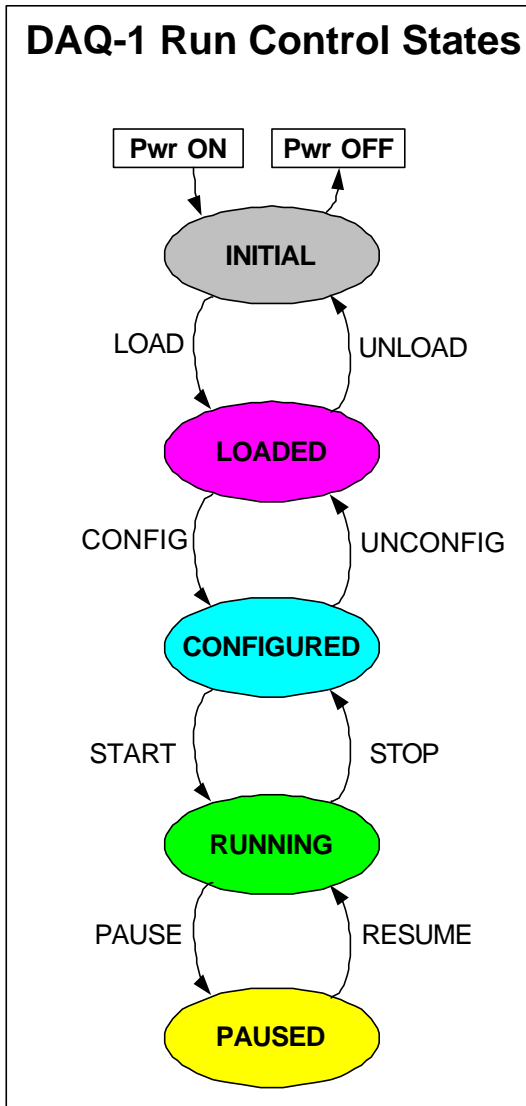


SCT “Calibration”: A Few Ideas

A work in progress...

Present Interpretation of DAQ-1 States



(from Pixel Calibration: A Use Case)

INITIAL to LOADED

Check ROD status registers

LOADED to CONFIGURED

Load and Start Slave DSPs
Configure BOC

CONFIGURED to RUNNING

Setup event trapping
Start calibration event loop

RUNNING to PAUSED

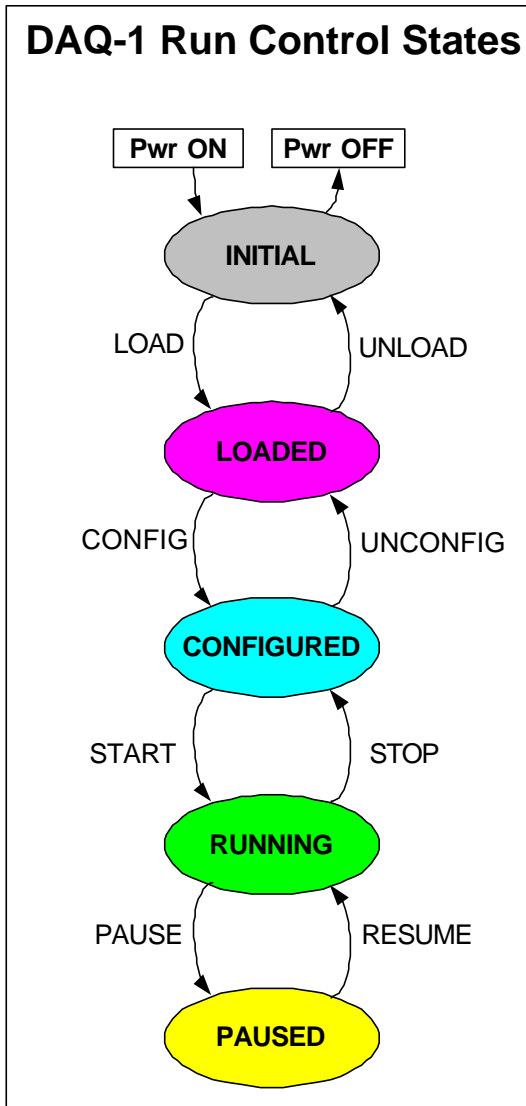
Nothing

Problems with that model

- ❑ Upon entry to the Configured state, surely it is good practice to bring the detector modules to their default operating condition?
- ❑ BOC must be configured before detector module LV is switched on (clock must be present)
- ❑ Detector module configuration and operation are sensible only if suitable power is applied to the modules: certain transitions between DAQ states correspond to transitions between DCS states

=> are we making the most sensible interpretation of these states?

Alternate Interpretation of DAQ-1 States



INITIAL to LOADED

Load Configuration Data
Configure BOC
Load and Start Slave DSPs

LOADED to CONFIGURED

Request DCS moves to STANDBY
Send initial configuration to detector modules


CONFIGURED to RUNNING


Request DCS moves to RUN
Enter Interactive mode

RUNNING to PAUSED

Nothing

INITIAL ↔ LOADED

- 
- Load Configuration Data
 - Read status register (HPI)
 - Read reply buffer (HPI)
 - Check status registers (HPI)
 - Mask all formatters (Prim)
 - Load slave code (Prim)
 - Start slaves (Prim)
 - Set read/write register fields (including BOC registers) (Prim)
 - Enable all formatters (Prim)

- 
- Reset board (HPI)
 - Check status registers (HPI)
 - Tell Master DSP that slaves are stopping (Prim)
 - Stop slaves and reset them (Prim)
 - Request DCS moves to OFF (if at higher state)

LOADED ↔ CONFIGURED



- Request DCS moves to STANDBY (if at lower state)
- Send initial configuration to detector modules



- Request DCS moves to STANDBY (if at higher state)

CONFIGURED ↔ RUNNING



- Request DCS moves to RUN (if at lower state)
- Enter RUN - INTERACTIVE mode (for “calibration” tests)



- Send initial configuration to detector modules
(No change in DCS state – leave at RUN ready for further tests)

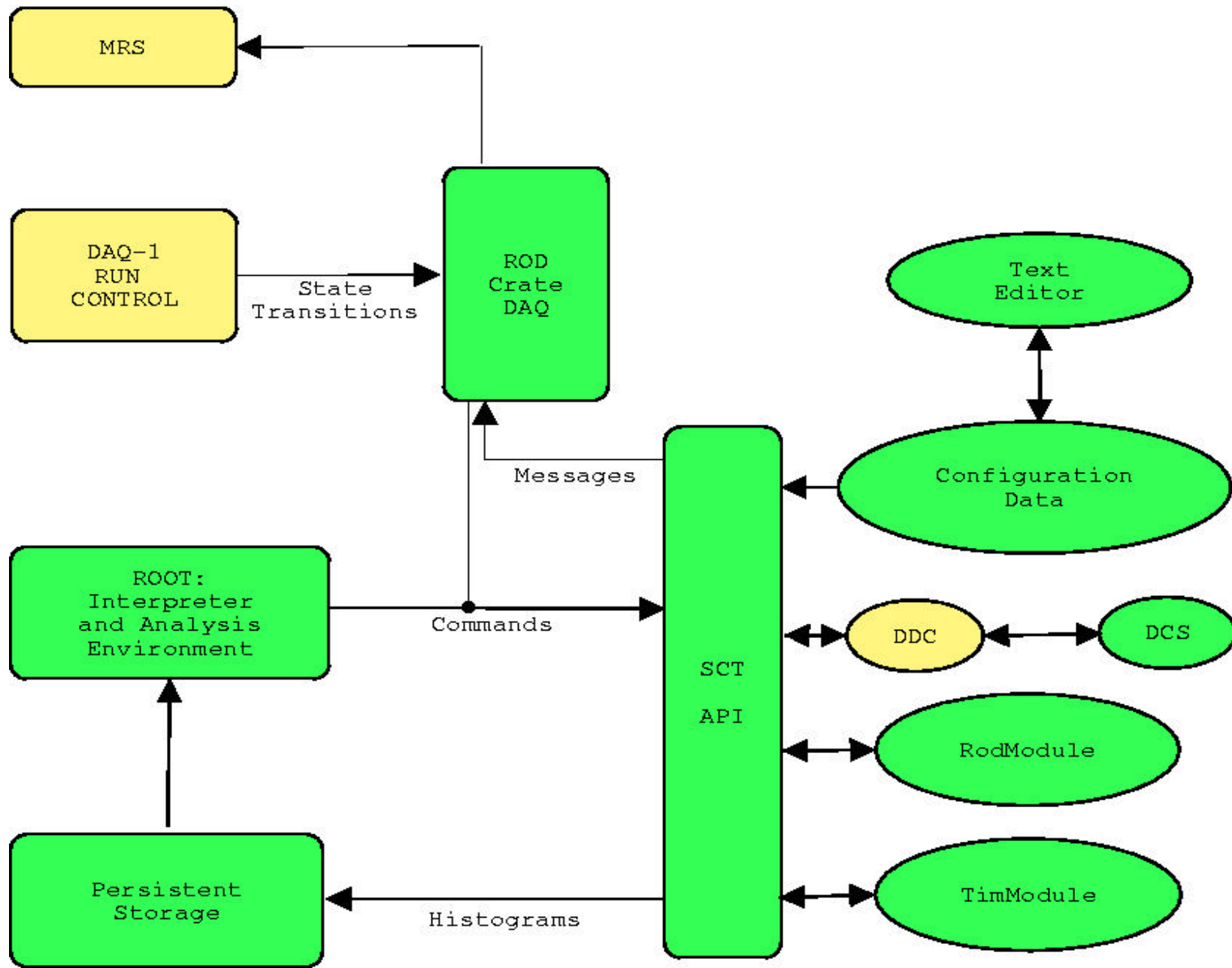
RUNNING ↔ PAUSED

↓ No action

↑ No action

Interim Analysis and Archival

- ❑ Is there a simpler place to start than IPC?
- ❑ What's wrong with binary files (direct dump of histogram memory) stored within a suitably structured file system, eg.
RODDATA/run/test/scan/rawhist.bin
- ❑ Configuration information, DCS data and fit results may be treated similarly.
- ❑ The resulting binary files could be analysed on or off line.



Communication Model

SCT API

This is similar to Paulo's "Pixel Function Library" but, at least initially, with less high level functionality. To begin with we would propose to leave operations such as the ResponseCurve (cf. pixel calibration) and TrimScan (cf. pixel threshold equalisation) to be performed as a sequence of individual scans controlled from the interpreter.

SCT API – Required Functionality

- ❑ To be able to read in system and module configuration and calibration information from persistent storage.
- ❑ To be able to vary the parameters of a module's registers, from the system level down to the level of one chip.
- ❑ To configure trigger parameters (eg. select calibration strobe)
- ❑ To configure BOC parameters
- ❑ To configure DCS parameters (eg. VCSEL and PIN voltages)
- ❑ To configure and execute scans
- ❑ To retrieve and save scan data
- ❑ To be able to define arbitrary groupings of modules and treat them differently when changing parameters, executing scans etc.

SCT API – Indexing Modules

We may need to use several methods to reference each module. SCTAPI should include a suitable translation facility to accommodate this:

- By module serial number
 - e.g. 20220330200012
- By module location
 - barrel/row/number
 - disk/ring/number
- By readout channel
 - partition/crate/rod/channel

(It would be good practice to copy the module serial number and location information to ROD such that the histograms are labelled at source.)

Other ideas

- ❑ For diagnostic purposes, it would be useful to develop a scheme for the persistent storage, retrieval and manipulation of primitive lists.
- ❑ We would then request the ability to load and executed saved primitive lists from our command interface or script interpreter.
- ❑ I know, this sounds like TestStand, but I believe we will need a similar facility for use with the SBC.

- ❑ Does the DAQ need to react to certain DCS conditions, such as power supply trips? Do we need an Alarm Handler?