

Prominence Magnetometer Software

Introduction:

Refer to *ProMag_OpticsandControl* for an overall description of the Prominence Magnetometer (ProMag). The memo *ProMag_Operations* describes how ProMag operates. This memo proposes a top-level design for software needed to make it operate.

ProMag software resides on three machines, *wires* that controls the experiment, *smoke* that controls the PixleVision camera, and *mirrors* that controls the Rockwell camera. Applications to control cameras were originally coded for an evolutionary phase of the Advanced Stokes Polarimeter into SPINOR and have been modified for ProMag due to a different polarization modulation scheme. Applications on *wires* to control the experiment are entirely new. These are ProMag Control and the ProMag GUI.

ProMag Control Application:

Supervises ProMag observing operations such as Map, Calibrate, Setup.
Low-level command of all mechanisms.
Receives commands from keyboard or via TCP/IP from ProMag GUI
Controls all mechanisms as appropriate for operations.

ProMag Control Threads:

Main: Initializes the instrument via direct calls to hardware and setting of software parameters.

- Starts Observe and TCP Command threads.

- Receives commands from the keyboard, calls command parser that performs calls to hardware. Commands need to be defined to control all devices and all Observation commands.

- Sends commands to Observe thread via inter process communication (IPC).

- If TCP Command thread is connected to the ProMag GUI, returns status to the ProMag GUI.

- Shuts down the instrument via direct calls to hardware.

TCP Command: Receives commands via TCP/IP from the ProMag GUI running on the same or another computer and calls the command parser.

Observe: Receives commands from the Main thread.

- Loops through Maps, Calibrations, and Setup operations sending commands to cameras to record data, and to mechanisms to change as appropriate to the observation.

- Sends status via TCP/IP back to ProMag GUI if TCP Command is active

Newport Functions:

Devices are controlled via the Newport software library. Units are degrees or millimeters depending upon whether device is linear or rotary. All functions return immediately. When motion is started devices return 'busy'. If a device has arrived at a previously commanded position and stopped, a second call to the same position returns that position. If the argument is invalid an error is returned. Devices could also return

errors if a limit condition is present or there are other failures. All devices share the following commands.

Init: Initialize the mechanism. May require finding home. Returns 'busy' or current position if done.

Home: Go to the home position and stop. Returns 'busy' or current position if done.

Abort: Stop all motion immediately.

Move_Rel: Move to relative position. Argument is relative amount to move. Returns 'busy' or current position if done.

Move_Abs: Move to absolute position. Argument is absolute position. Returns 'busy' or current position if done.

Run_Rate: Applies only to image rotator. Arguments are starting absolute position and rotation rate. Returns 'busy' or some error.

6733 Functions:

The NI 6733 waveform generator board produces strobes for cameras and voltages for the six ferroelectric liquid crystal (FeLC) units. Commands are via the National Instruments command library. The eight waveforms are generated continuously once started. The FeLCs must not be exposed to DC except for 0 VCD.

Init: Perform any board initialization required and set all voltages to zero.

Abort: Set all outputs to zero immediately.

Modulate: Start sending out waveforms. There are nine arguments - rate in Hz, camera 1 delay in fraction of a cycle (like 0.33), camera 2 delay in fraction of a cycle, V1, V2, V3, V4, V5, & V6, the voltages for the six FeLC units. See timing diagram. Camera strobes are 0 to 5 VDC.

Temperature Controller Functions:

Control the Alpha-Omega temperature controller via the RS-232 port.

Init: Whatever initialization is required for the interface and controller

SetTemperature: Set controller to control at a particular temperature. Argument is temperature in Celcius. Returns current temperature or Error.

Status: Return controller status information such as power level and present time temperature.

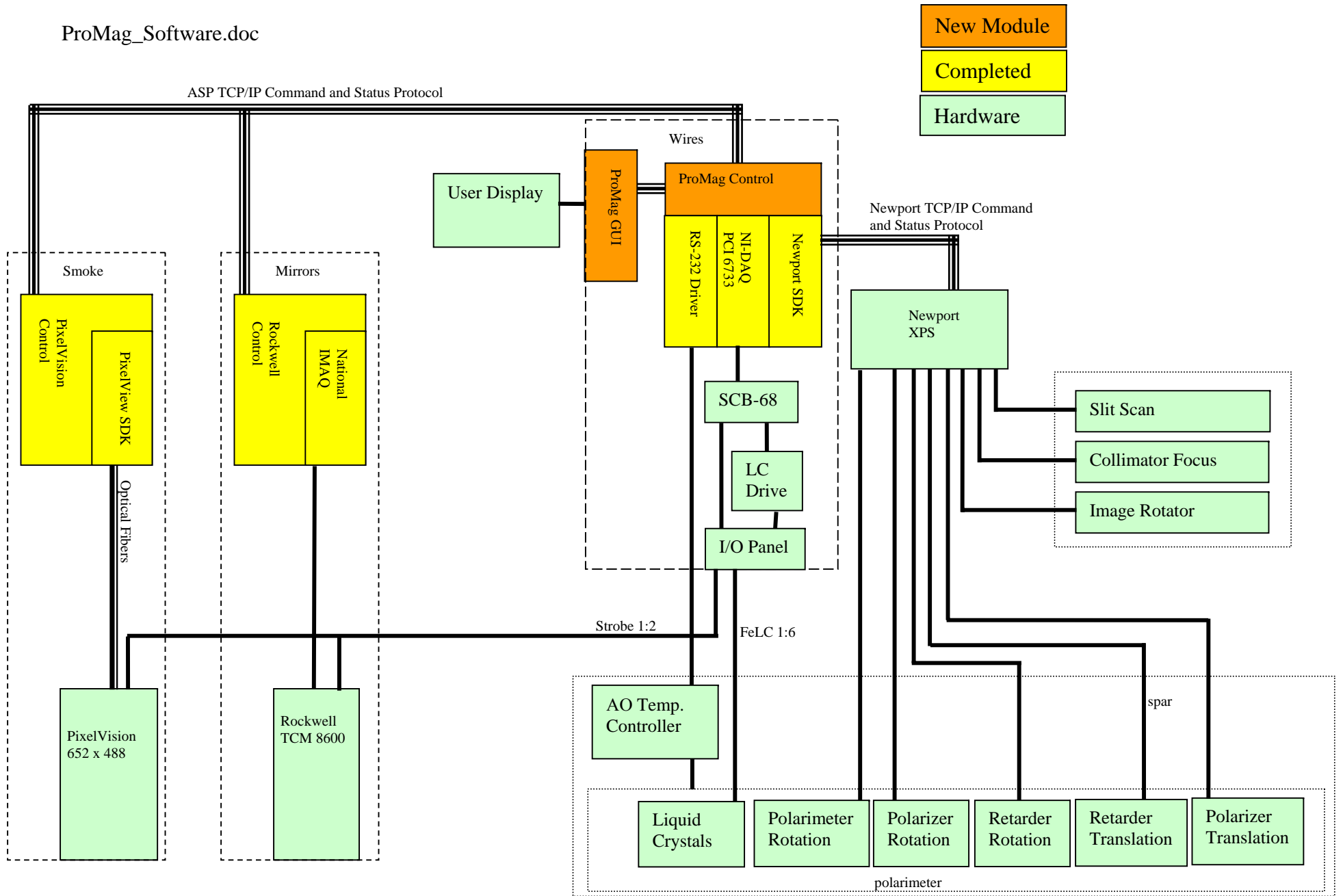
ProMag GUI Application:

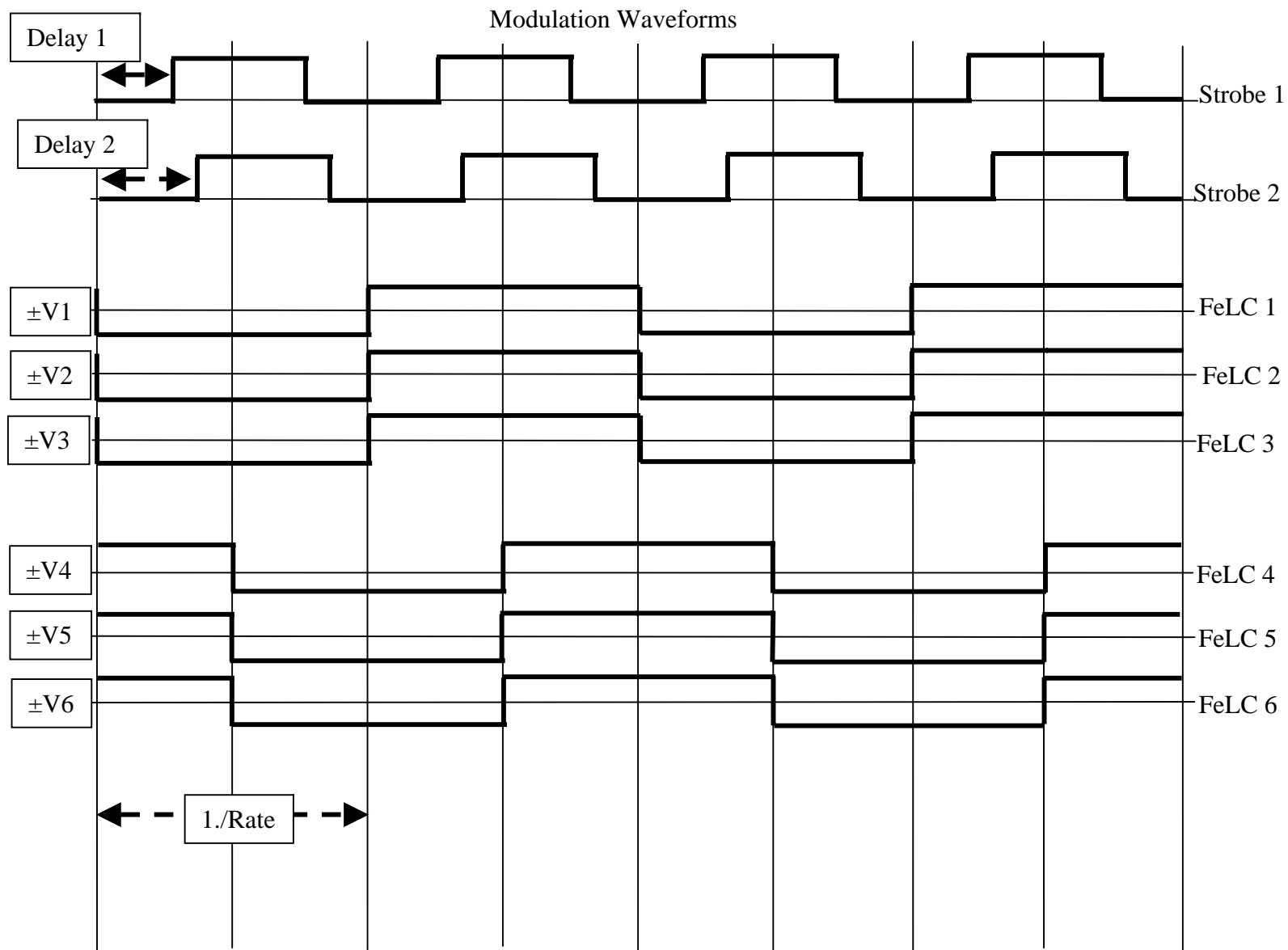
Graphical interface to ProMag Control.

Widgets provide graphical control equivalent to keyboard commands.

Status of operations and mechanisms is displayed.

May run on ProMag Control computer or elsewhere.





-End-