Expanded RocketDriver GUI Notes and Requirements

Datalogging

* Make this a separate C/C++ program managing the CAN controller for reading messages in at full rate and writing to a dump file. Dump file stores all raw frames.
* Also decode the most recent sensor values and state messages for each object and store in another file (or memory, however you would like to handle the location is up to you). This file will be used by the GUI to look up the state of everything received at a given GUI program refresh rate.

GUI

* There shall be three separate sections of clickable command buttons.
  + Group 1 – Test Mode and Individual Valve Commands. The Test button sends a state command to the stand/rocket to enter “Test” state where it will accept individual actuation commands. The rest of the buttons here all represent every individual controllable output device on the stand/rocket.
  + Group 2 – Primary State Commands. Other than Test, the commands here only tell the stand/vehicle to enter a given state which has a group of actions that happen. These also have hard coded rules based on the system state diagram.
  + Group 3 – Node Status. Each remote node will be wired with an output to the hard reset pin. These buttons require a second confirmation command. Any other information associated with the node, such as MCU temperature for example, should be grouped together with the clickable reset button.
* Visual Indicators
  + There shall be visual indicators for all of the following that are clear and intuitive for non-Avionics team members to read during operations.
    - The state of all commands based on received confirmation from the stand/rocket nodes (i.e. a valve does not indicate on because the GUI button was clicked, but because the remote nodes have reported back that they are in that state)
      * Each Node System State
      * Each valve state
      * Each valve enable/lockout state
    - That states are currently locked out from being entered. This is based on the system state diagram.
    - Data values for all sensors.
    - If data is “current” for all information displayed. We shall determine a system for displaying when information is stale based on timestamps of last update on that information.
    - For when data received is within determined ranges that represent a system state. Consider this as active feedback on the real sensed state of the propulsion system and not just the avionics system. (i.e. the tank pressure transducers have visual indicators for when tanks are depressurized, at low pressure that is still an operational safety concern, and above the threshold of operational pressure to fire the propulsion system).
    - For when data received is outside of the safe operating range of the hardware. This is an emergency warning indicator that the propulsion system is outside of the certified operating ranges.
    - Optional – graphs for sensor data over time. This will depend on the space allotted after meeting the critical requirements. Here are some following examples that would be of particular interest.
      * Thrust and chamber pressure data over time during the engine firing.
      * Tank sensors (temperature and pressure) during filling and pressurization
      * Load cell values for vehicle weight during filling and launch
      * IF HARDWARE IS ADDED IN FUTURE – Load cell readings on tanks during filling and during firing for static fire stand.