2024-02-26 RHD

**Status of Antonio Feed Controller Update**

The impetus for doing an update to the Antonio Feed controller is the lockup of the accelerometer and subsequent need to reboot the feed controller to recover.

**Accelerometer**

The accelerometer has an internal hardware state machine that takes care of timing and sequencing of taking measurements and sending the results via the I2C bus. If the results are not read in a timely manner, new measurements are made, sometimes resulting in the state machine locking up.

The existing feed controller uses software timers to determine when to read the accelerometer and that can get out of sequence with the accelerometer timing.

The accelerometer is intended to use an interrupt of the controller to ensure the prompt reading of measurements and keeping the controller and accelerometer in lockstep. This was implemented as a separate piece of code and tested for several days without lock up.

**Code Integration**

Before integrating the new interrupt-driven accelerometer interface, the existing code was compiled. Unfortunately, compiling the existing code was not successful due to missing library files in the area of the disk file system. Legacy libraries were added and compiling attempted again. No luck.

The whole development suite was removed and reinstalled with legacy libraries selected from the beginning. Still missing library files were flagged. Quite a bit of time was spent drilling down into the library to find the missing pieces and many layers deep were dead ends. Also as noted in copyright notices, these libraries came from the early days of Unix development in universities. The troubles with these libraries are probably why years ago Microchip replaced them with a whole new set of libraries and one has to dig deep to even find these legacy libraries.

**Code Rewrite**

After wasting too much time trying to resurrect the existing code and noting many areas of very shaky timing due to the software timing scheme, a rewrite of the whole controller code was started. Due to lack of documentation and the very confusing comments in the existing code, many of the features of the controller are not well understood.

Fortunately, datasheets for the vacuum pump sorted out some of the pass-through commands that were not documented in the existing code at all. A trip to HCRO and logic analyzer traces of the interaction between the controller and the rim box helped sort out many of the protocol issues.

**Auto-Start**

How auto-start is supposed to work is the biggest unknown and the roadblock to finishing the code rewrite. The existing auto-start depends on a full-blown disk file system that is very flexible, but also very complex to implement for the file system itself and the message parser.

**Proposal**

The commands for setting and reading back individual auto-start parameters will be implemented as the existing protocol. For reading back of stored auto-start parameters, I propose that a new command be implemented that reads back an annotated list of the settings. This would allow a very simple store/fetch of the NV memory to be used instead of a full-blown disk file system.

If there is need to multiple sets of auto-start parameters, then I propose determining the number of sets desired and implementing memory slots to match with a readback command. The slots could even be named with another set of commands if desired.

**Summary**

The intension is that the protocol used for the automated query of the feed controller be implemented exactly the same so that the existing monitoring system can continue as is.

If the manual interaction with the feed controller for auto-start settings can be modified to eliminate the dependance on the disk file system, it would be much easier to implement and test the code, and possibly be easier for the operators to interact with.

**Added by AP:**

There are a number of scenarios the feed control board needs to handle. Below is a list of possible scenarios and how the feed control board should behave.

1st Normal operation:

* Controller is powered up with feed at room temperature (TC above 250k) and start mode in automatic, this is the state when you want to cool down a feed newly installed in an antenna.
  + Run diaphragm pump for 60 min (this should be a variable to set time in minutes 10 – 240 min)
  + Start turbo pump on standby RPM 18,000 for 60 min (this should be a variable to set time in minutes 10 – 240 min).
  + Spin up turbo pup to full RPM 90,000 for 60 min (this should be a variable to set time in minutes 10 – 240 min). After time has passed check if turbo power (p316) is below 20W if so move to next step otherwise restart timer for this stage.
  + Start cryocooler with operation set to fixed power control, set power to 70W run until feed temp is below 130K (TC) then go to next step. (the temp and the power for this step should be a variable that we can change too.)
  + When below the 130k threshold change the cryocooler control to temp control loop, and slowly increase the allowed power from setpoint above (70W) to max which is 240W. (the increase in allowed power should also be a variable. Eg. x Watt per 10 min)
  + Now we are stable and running.
* Controller is powered up with feed cold (recover after power outage) TC below 150K and last state of cryocooler was ON.
  + Check vacuum system is ON
  + Check cryocooler is ON (allow for max power of 240W)
  + Cooldown feed

Fault shutdown situations:

* When cryocooler is on, temp sensor A5 and A6 monitor the temperature, should the temperature exceed XX degC, shut down the cryocooler and provide error code.
* When cryocooler is on, check the vacuum system, if the turbo pump shuts down the system also needs to shut down the cryocooler.
* Any error on turbo pump, too hot, too much power, etc. needs to be caught and shown in the feed firmware error log / state.