## WISPR program execution

The WISPR CompactFlash card can have a file named “start” at the top level which will be run by the init process. The WISPR shuts down after the shell script exits.

For the LARA winch we want to execute a detection program with several command line settings for part of each hour. Before running the detection program, we want to check available disk space. Several times a day we also want to run a spectrograph to determine wind and wave conditions at the surface. Also, we sync exact time on the CF2 controller and WISPR boards to gps time.

Instead of running the several programs directly from the WISPR start script, we instead read from the serial port connecting WISPR to the CF2 for 10 seconds and then execute any text as a shell command. Before and after this read/execute loop the script sends markers in an <xml> format to the CF2, and this could include a marker after each program executes if needed.

The following is a snippet from the WISPR /mnt/start shell script:

echo -n "<wispr>" > /dev/ttyBF1

line=""

# -s silent, -t timeout

read -s -t 10 line < /dev/ttyBF1

if [ -n "$line" ]; then

echo "$line"

eval "$line"

fi

echo -n "</wispr>" > /dev/ttyBF1

So, several linux commands can be executed and typically those commands will use the same serial port for textual I/O. The following is a snip of code from CF2 wsp.c which runs the spectrograph program. This code is abstract enough to be pseudo-code as an example:

// format spectrograph command line from basic command + flags + gain + logfile

sprintf( b, "%s %s", wsp.spectCmd, wsp.spectFlag );

if (wsp.spectGain)

sprintf( b+strlen(b), " -g%d", wsp.spectGain );

if (wsp.spectLog)

sprintf( b+strlen(b), " -l %.5s%03d.log", wsp.spectLog, all.cycle );

// start program

if (wspOpen()) raise(1);

flogf( "\nexec '%s'", b );

utlWrite( wsp.port, b, EOL );

// listening for over two minutes, then computation takes a while

if (!utlReadExpect(wsp.port, buf, "RDY", 200)) raise(2);

utlWrite(wsp.port, "$WS?\*", EOL);

if (utlReadWait(wsp.port, buf, 60)) raise(3);

flogf("\nwspStorm prediction: %s", buf);