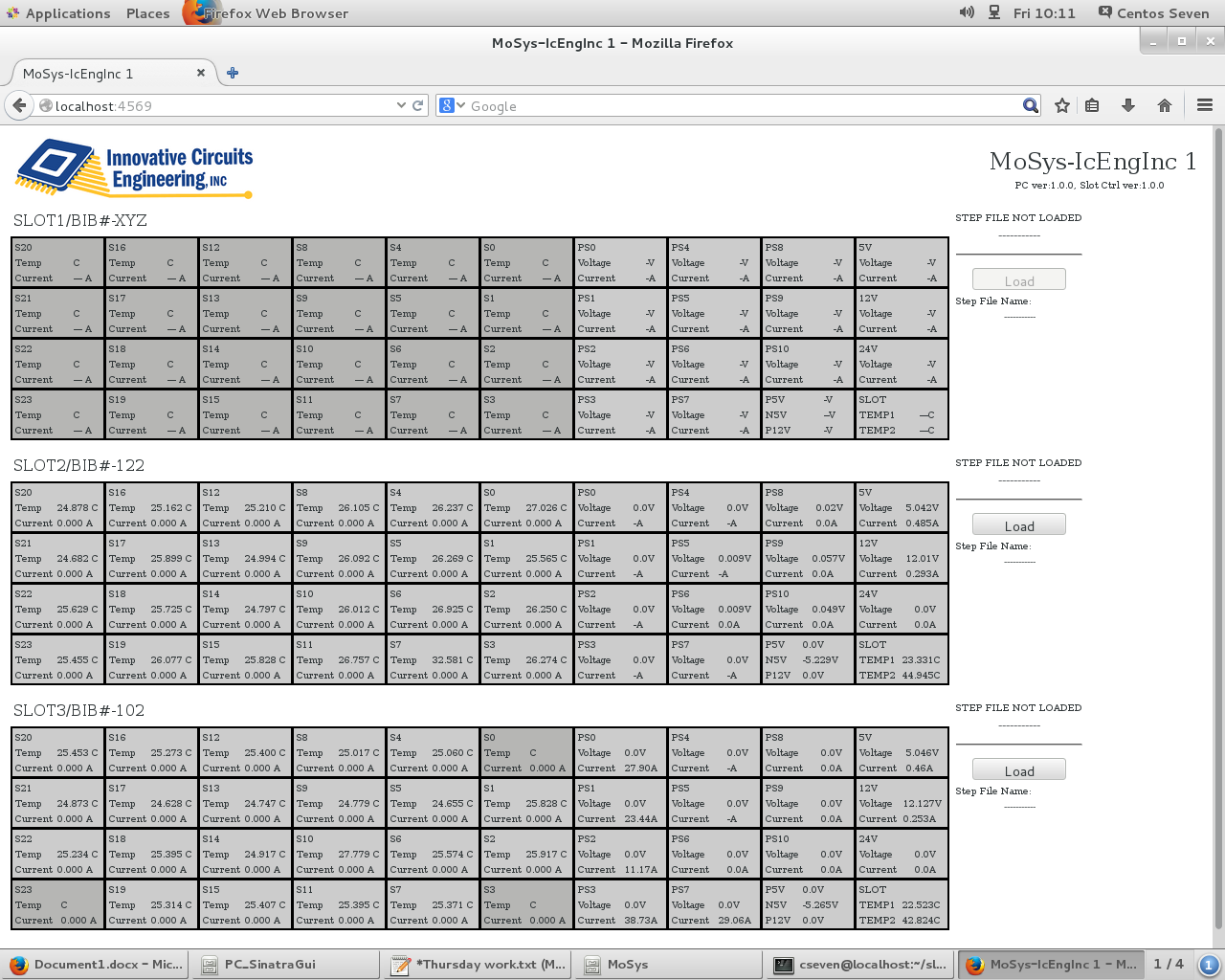
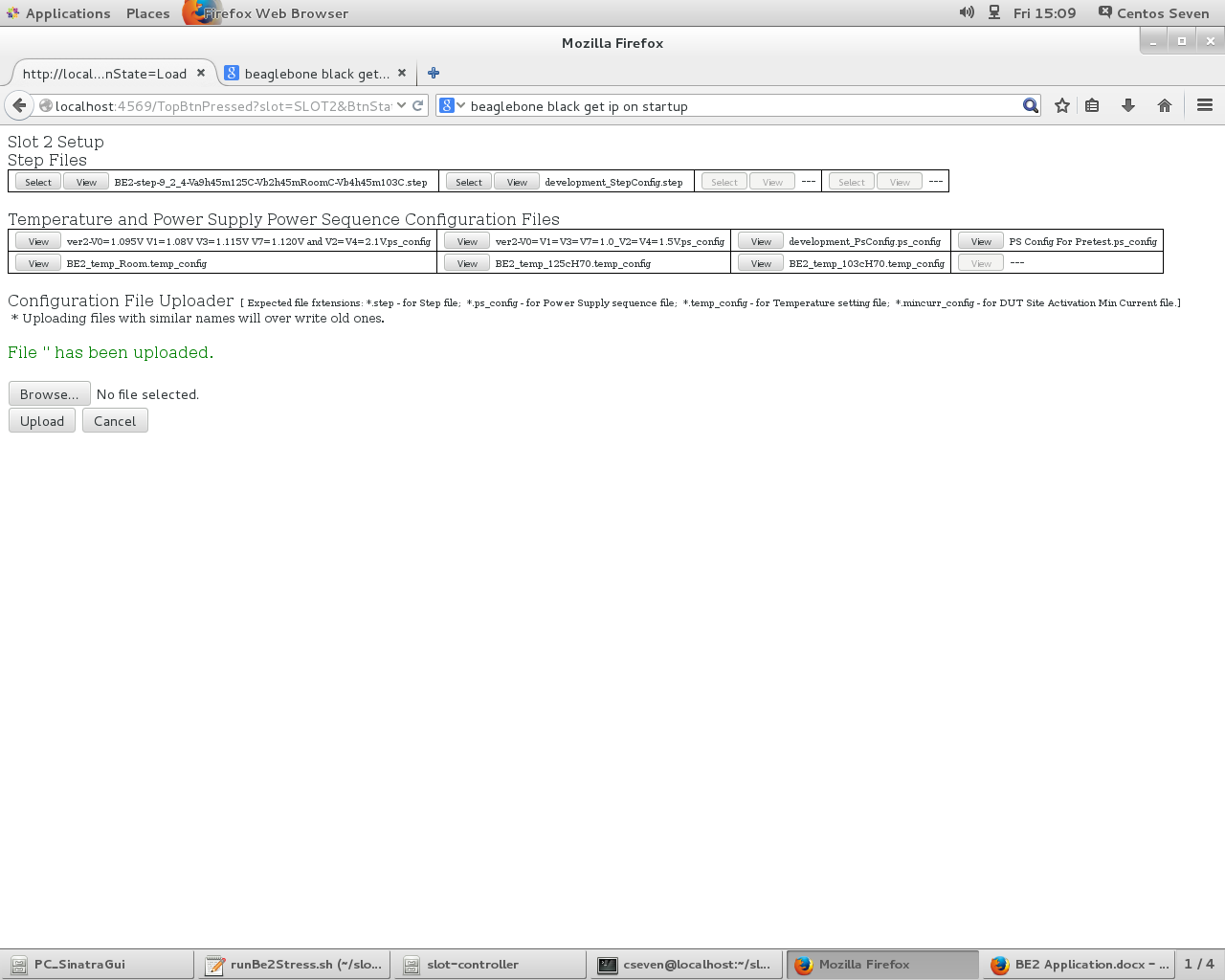
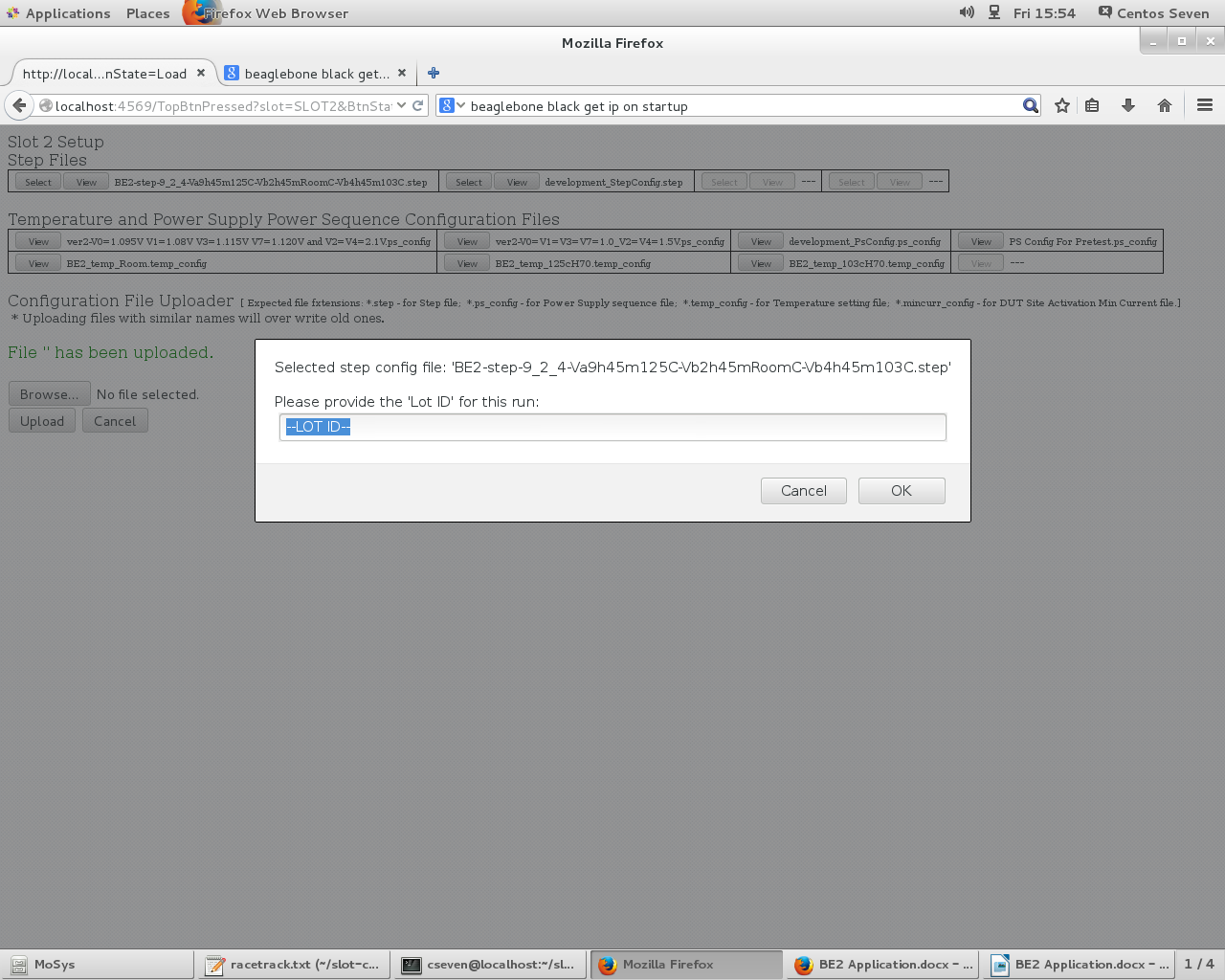
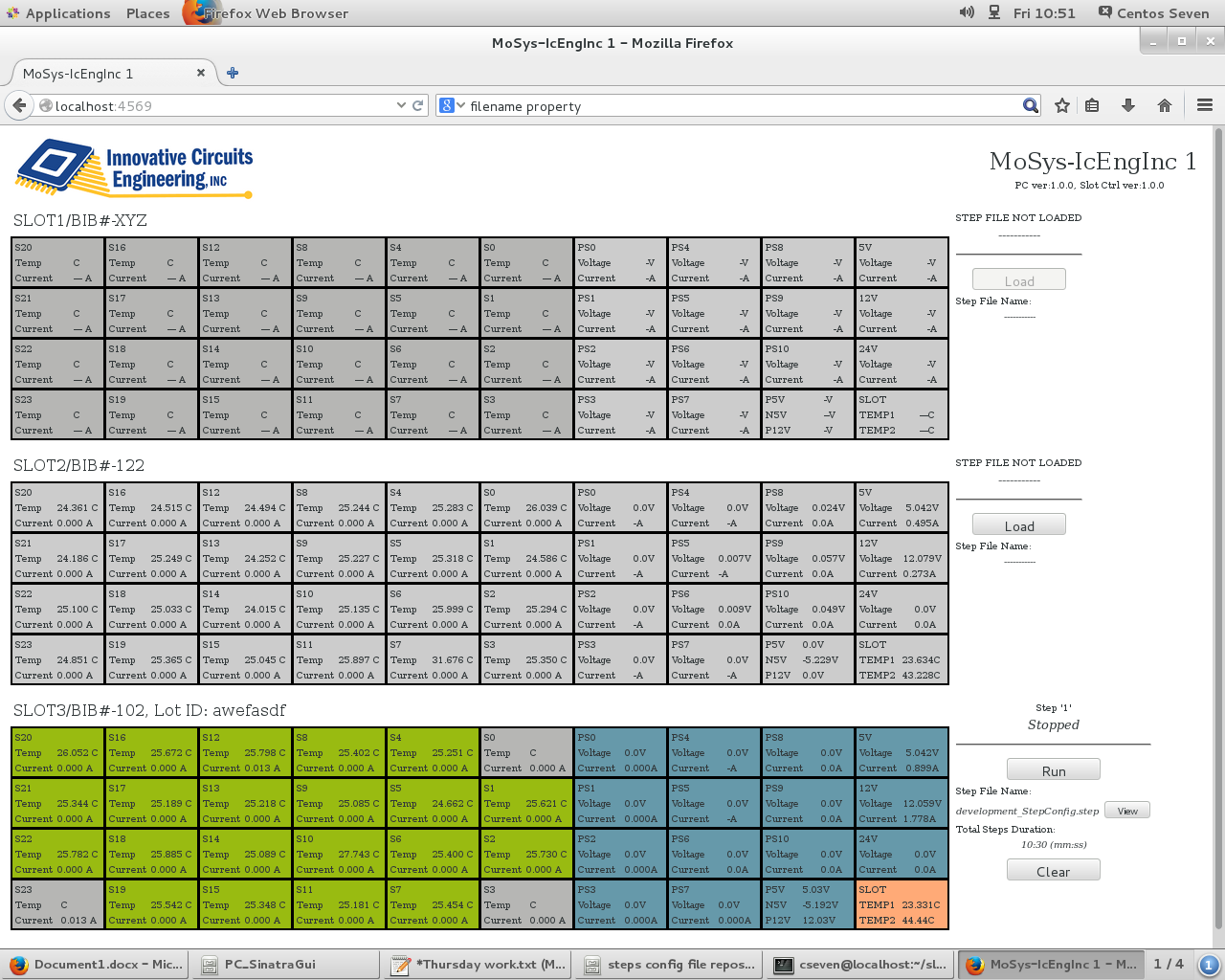
Running the BE2 application from boot up

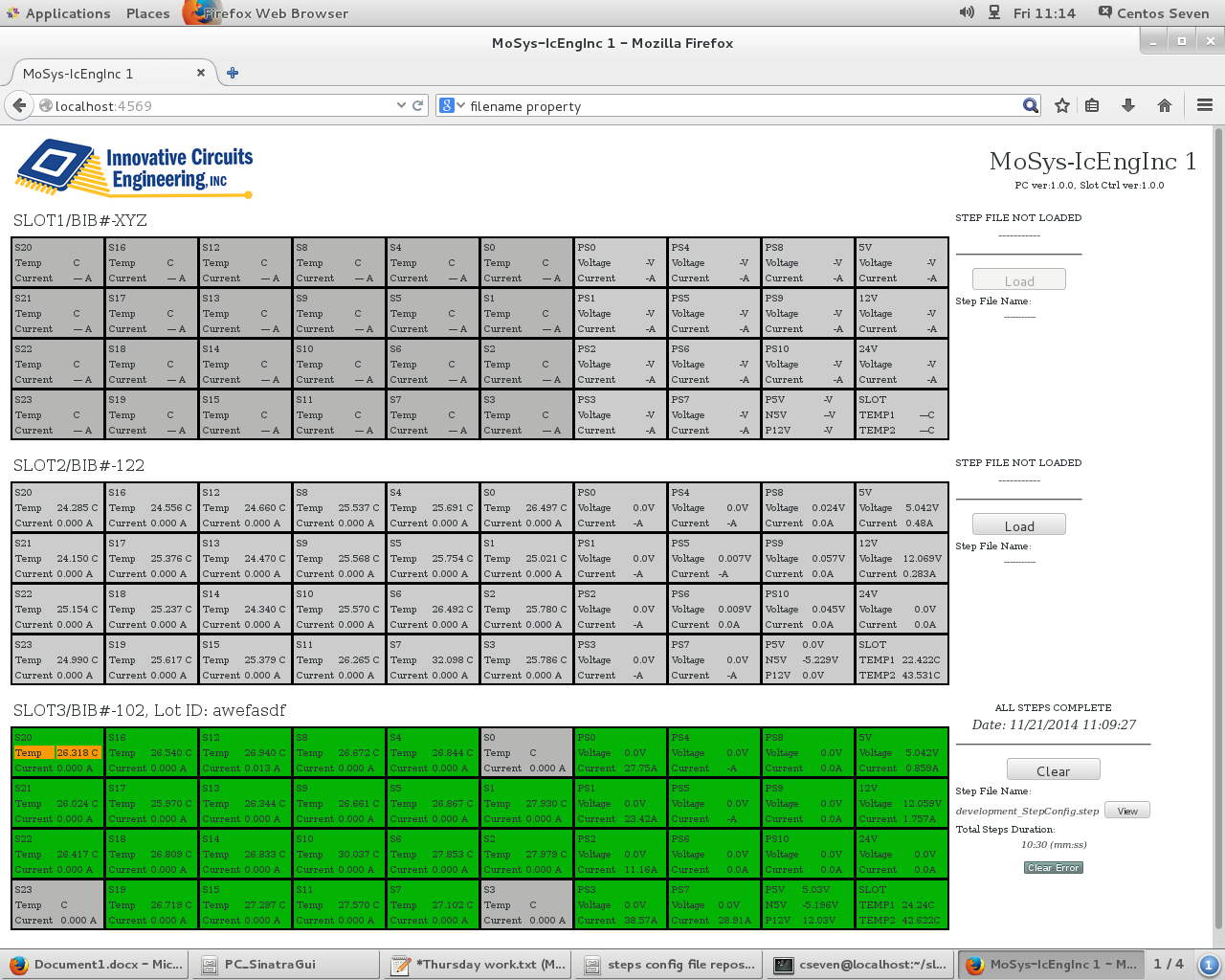
1. Ensure that Ethernet (em1) is "On".
   1. If it's not on, do the follow:
      1. Open the Network settings dialog box through this following menu:
      2. Applications->System Tools->Settings. Then select the "Network" icon.
      3. Once the Network settings dialog box is open, select "Ethernet (em1)" and turn it on by click the Off switch to On.
2. Open a terminal on the PC that runs the BE2 machine.
3. Do the following commands on the terminal without including the quotes:
   1. "cd ~/slot-controller/"
   2. ". runBe2Stress.sh"
4. Once the messages on the terminal given by the last command is read and a key is pressed as instructed, the terminal will close.
5. A browser will open, and there should be a tab label "Mosys-IcEngInc 1", or what ever the name is given the machine.
6. Wait for about a minute for the browser tab to show a similar display below:
7. Once tab shows the display, lots can now be run on the system.

Running Lots

To run lots, a few configuration files must be provided in the "~/slot-controller/steps config file repository" folder, where files contain pertinent information for the BE2 machine will use.l

* The following files with given extension must be present along with proper information provided within each file:
  + \*.step - This file contains the recipe to run the steps on a lot. Generally, it contains which \*.ps\_config, and \*.temp\_config files to use per step; how low a step is to run under the conditions listed on the steps, step name.
  + \*.ps\_config - This file contains the Power Supply configuration for a step. It generally has the nominal settings, and the trip points and tolerance points for voltages and currents.
  + \*.temp\_config - This file contains the temperature settings for a step. It generally has the nominal temperature settings along with the trip points and tolerance points.
  + \*.minCurr\_config - This file is not used, but it must be present at its current file format since the application expects it.
* Once the expected files listed above are available, to run a lot, click on the "Load" button for the slot on which the lot is suppose to run and do the following:

1. A page of file listings will open like what's displayed below, then select the \*.step file for the lot is to execute on when running.
2. A pop-up dialog box will open, and provide the Lot ID for the run.
3. Press OK on the pop-up dialog box.
4. A similar screen below will then display, depending on the slot selected:
5. If a DUT is grayed out, but there's a DUT on it, clear the lot by pressing the "Clear" button, and re-load the step-configuration file. If it stays gray but it's not suppose to be gray, the DUT is faulty.
6. The "Clear" button clears the uploaded step file
7. The "Run" button starts all the steps to its completion.
8. Run the corresponding racetrack script:
   1. ~/TDBI/be2stress INT<XX>,<YY>,<ZZ> <BIB#> <LOT ID> Test, where:
      1. <XX> - is an interger representing the hours of how long step 1 is configured to run. Usually, if <XX> is set to 9, step 1 is configured to run for 9hours with added padding of 45 mins, so basically step 1 is set to run for 585 minutes.
      2. <YY>, and <ZZ> represents the number of hours for step 2 and step 3 consecutively, and the steps are configured to add 45 mins per step.
      3. <BIB#> - is actually the burn-in-board number that is written on the board in which the software is to run on.
      4. <LOT ID> - the Lot identifier of the run.
9. If the slot is not running, a solid yellow bar of light will be on - on the corresponding slot on the machine.
10. If the machine is running, a solid green bar of light will be on - on the corresponding slot on the machine.
11. If the run encountered an error, a red bar of light will be blinking along with a beeping sound.
12. The beeping sound can be silenced for 5 minutes by pressing the right button.
13. The error on the slot can also be cleared by clearing the step configuration.
14. If there are tolerance faults while running, DUTS will have an orange bar indicating the fault. If the fault is over temperature tolerance or under temperature tolerance, the data on the temperature display will be orange. The left side will stay orange indicating that a fault occurred on that DUT. If the right side is orange, it means it is currently at a fault.
15. A small "Clear" button with grayish-green color will also show if faults become present. Pressing this button will clear the display of faults. New faults may show, and the button will become available again.
16. Once the run is complete, a green bar of light will be blinking, and it will have a similar image given below:



1. Once the run is complete, log records must be copied to the FTP server 192.168.1.4, username: Mosys, password: mosys1234. The log files for the run are dropped in the "BE2\_LogFiles/YYYY-MM-DD-run" folder, where YYYY-MM-DD is the date when the lots were executed.
2. Copy the racetrack log files also in the save ftp folder. The folder to be copied is found at "~/MoSys/", and the folder name is in M<YY><MM><DD><xxx> format where <YY>,<MM>,<DD> is the date when racetrack was executed, and <xxx> could be INT or BI. For example, if racetrack was run on 12 Nov 2014 and completed, a folder in "~/MoSys/" to be copied into the ftp folder will be "M141112INT" or "M141112BI".

Log Files

When a lot is set to run it will generate a log file, and the file will be found in this directory:

"~/slot-controller\_data/steps log records"

The file has the following format:

"iceLog\_brd<Board #>\_lot<User provided lot name>\_time<YYYYMMDD\_mmss>.log"

Where :

* <Board #> is the board number given by MoSys.
* <User provided lot name> - is the lot name given by the user.
* <YYYYMMDD\_mmss> - is the date and time of lot uploaded to the slot.

If the file is larger than 10meg, the software will split the file into 10meg pieces. For example if a log filename "iceLog\_brd122\_time20141113\_155059\_lotTS2-20141113.log" is over 40meg, a list of files will be found with names like:

"iceLog\_brd122\_time20141113\_155059\_lotTS2-20141113\_Partaa"

"iceLog\_brd122\_time20141113\_155059\_lotTS2-20141113\_Partab"

"iceLog\_brd122\_time20141113\_155059\_lotTS2-20141113\_Partac"

"iceLog\_brd122\_time20141113\_155059\_lotTS2-20141113\_Partad"

"iceLog\_brd122\_time20141113\_155059\_lotTS2-20141113\_Partae"

Each file will be at most 10meg in size.

Setting Up Steps

To skip steps 2 and 3, put in 0 (zero) value for step 2's "4,TIME,STEP TIME,M," row, and step 3's "4,TIME,STEP TIME,M," row. Doing so will not even call power supply sequencing once step 1 is complete, and completes the process there after.PC Setup

1. When setting up the PC with Centos 7, make sure to install Apache HTTP Server.
2. Once Centos is installed and the PC is running, open a terminal on the PC and get a copy of the ruby scripts that will run the PC by using git clone command (without quotes) at the home directory:
3. Go inside the slot-controller folder:
   1. "cd ~"
   2. "git clone <https://github.com/icenginc/slot-controller>"
   3. "cd slot-controller"
4. To configure the IP address of the Slot Controllers in which this PC will use, follow the instruction defined on the document titled, "PC Configuration To Communicate with the Slot Controller"
5. Setup the "runBe2Stress.sh" file.
   1. Edit the file runBe2Stress.sh, and provide the IP addresses of the slots which the PC will communicate to.
6. Get the needed files to run Ruby scripts:
   1. Install 'git'
      1. sudo yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel
   2. Install 'ruby'
      1. sudo yum install ruby
      2. sudo yum install gcc g++ make automake autoconf curl-devel openssl-devel zlib-devel httpd-devel apr-devel apr-util-devel sqlite-devel
      3. sudo yum install ruby-rdoc ruby-devel
      4. sudo gem update
   3. Install 'rack'
      1. gem install rack
   4. Install 'grape'
      1. gem install grape
   5. Create the Port2Interface.so file
      1. pushd ./lib/BBB\_GPIO2 Interface Ruby
      2. ruby extconf.rb
      3. make
      4. popd
   6. Install the listed files by running them on the terminal
      1. gem install sinatra
      2. gem install sinatra-contrib
      3. gem install rest-client
   7. Per MoSys, install a mail server:
      1. see http://www.krizna.com/centos/setup-mail-server-in-centos-6/#postfix
   8. Per MoSys, install tcl tk.
      1. sudo yum install tcl.x86\_64 tk.x86\_64
   9. To setup the static IP address of the PC by going through these menu items: Applications->System Tools->Settings. Click on the Network icon, select "Ethernet (em1)" then click on the "Gear" icon to open the "Wired" dialog. On the left panel, clicking on IPv4 will give you access on how to setup the PC's static IP address. Provide the PC's IP address there.
   10. For the Racetrack setup, ->reboot the PC<- after adding the following entries to the /etc/modprobe.d/blacklist.conf file:  
         
       blacklist ftdi\_sio  
       blacklist usbserial

Slot Control Setup

1. Get a copy of the ruby scripts that will run the system by using git clone command (without qoutes) in a BBB linux terminal: "git clone <https://github.com/icenginc/slot-controller>"
2. cronjob must execute the needed processes in the background in order for the PC can communicate to the controller, this must be setup by running "crontab -e" (without the quotes) on the slot-controller terminal. An editor will open and these lines must be added in the editor then save and close the editor.
3. The lines that are to be added are the three following lines without the quotes:
   1. "\*/1 \* \* \* \* bash /var/lib/cloud9/slot-controller/lib/DRbSharedMemory/runSharedMemory.sh"
   2. "\*/1 \* \* \* \* bash /var/lib/cloud9/slot-controller/BBB\_GrapeForPcListener/runBoardGrape.sh"
   3. "\*/1 \* \* \* \* bash /var/lib/cloud9/slot-controller/BBB\_Sampler/runSampler.sh"
4. Get the needed files to run Ruby scripts:
   1. If the PC you intend to use for setting up BeagleBone Black is Windows base, ensure to have the proper drivers loaded, and the drivers can be found at BeagleBone Black's web-page: <http://beagleboard.org/getting-started>
   2. Hook-up a BeagleBone Black via USB on a PC, and open a browser tab on this link: <http://192.168.7.2:3000/ide.html>
   3. Open a terminal tab, and run "\curl -sSL https://get.rvm.io | bash -s stable --ruby" without the quotes. It will ask the user to run another script (which the author didn't capture since it's only one time run) to request an ID code from the server.
   4. Once the script which requests for an ID from a server is executed, execute all these scripts in order inside the "slot-controller" directory:
      1. gem update
      2. aptitude install ruby1.9.1-dev
      3. gem install rack
      4. gem install grape
      5. apt-get install libsqlite3-dev
      6. gem install rest-client
      7. \curl -sSL https://get.rvm.io | bash -s stable --ruby
      8. pushd ./lib/BBB\_GPIO2 Interface Ruby
      9. ruby extconf.rb
      10. make
      11. popd
      12. gem install beaglebone
      13. # Set the time zone  
          ln -sf /usr/share/zoneinfo/America/Los\_Angeles /etc/localtime
5. Setup the IP address of the BBB/Slot-Controller:
   1. Using vi editor, make sure to have the following items in the /etc/network/interfaces  
      auto eth0  
       iface eth0 inet static  
       address 192.168.1.212  
       netmask 255.255.255.0  
       network 192.168.1.0  
       gateway 192.168.1.2
   2. The IP address given on the sample above is 192.168.1.212, and new IPs can be provided instead.
6. Setup the HDMI so Mode 7 will be available for the GPIO pins by doing the following terminal shell commands that are surrounded by quotes (make sure that the SD card is in place):
   1. "mkdir /mnt/card"
   2. "mount /dev/mmcblk0p1 /mnt/card"
   3. "vi /mnt/card/uEnv.txt"
   4. Then insert the following line on the editor vi (no surrounding quotes), then save and exit the editor.  
      "optargs=quiet capemgr.disable\_partno=BB-BONELT-HDMI,BB-BONELT-HDMIN"
   5. "umount /mnt/card"
   6. Reboot the board - "shutdown -r now"
7. To configure the IP address of the PC and the Power Supply in which this slot controller will use, follow the instruction defined on the document titled, "Slot Controller Configuration To Communicate with the PC".

PC Configuration To Communicate with the Slot Controller

For the PC to have the IPs of the Slot Controllers, same in order for the PC to know the board number that are provided by MoSys, Location of the system, and System ID, these information must be placed on the file "Pc\_SlotCtrlIps.config" file that sits in the "slot-controller\_data" folder of the PC.

Slot Controller Configuration To Communicate with the PC

For the Slot Controller to have the IP of the PC, same in order for the Slot Controller to know IPs of the power supplies it will use, provide and update the file:

"./slot-controller\_data/BBB\_configuration files/ethernet scheme setup.csv"

This can be done by doing ssh to the board, or through USB connection when setting up the board.

To do ssh, on a PC terminal, execute "ssh [root@192.168.1.211](mailto:root@192.168.1.211)" if the board that is to be accessed is on the IP address 192.168.1.211. Once the connection is in place, go to the slot-controller subdirectory by executing this command: "cd /var/lib/cloud9/slot-controller". Just go to the other directories listed as desired.