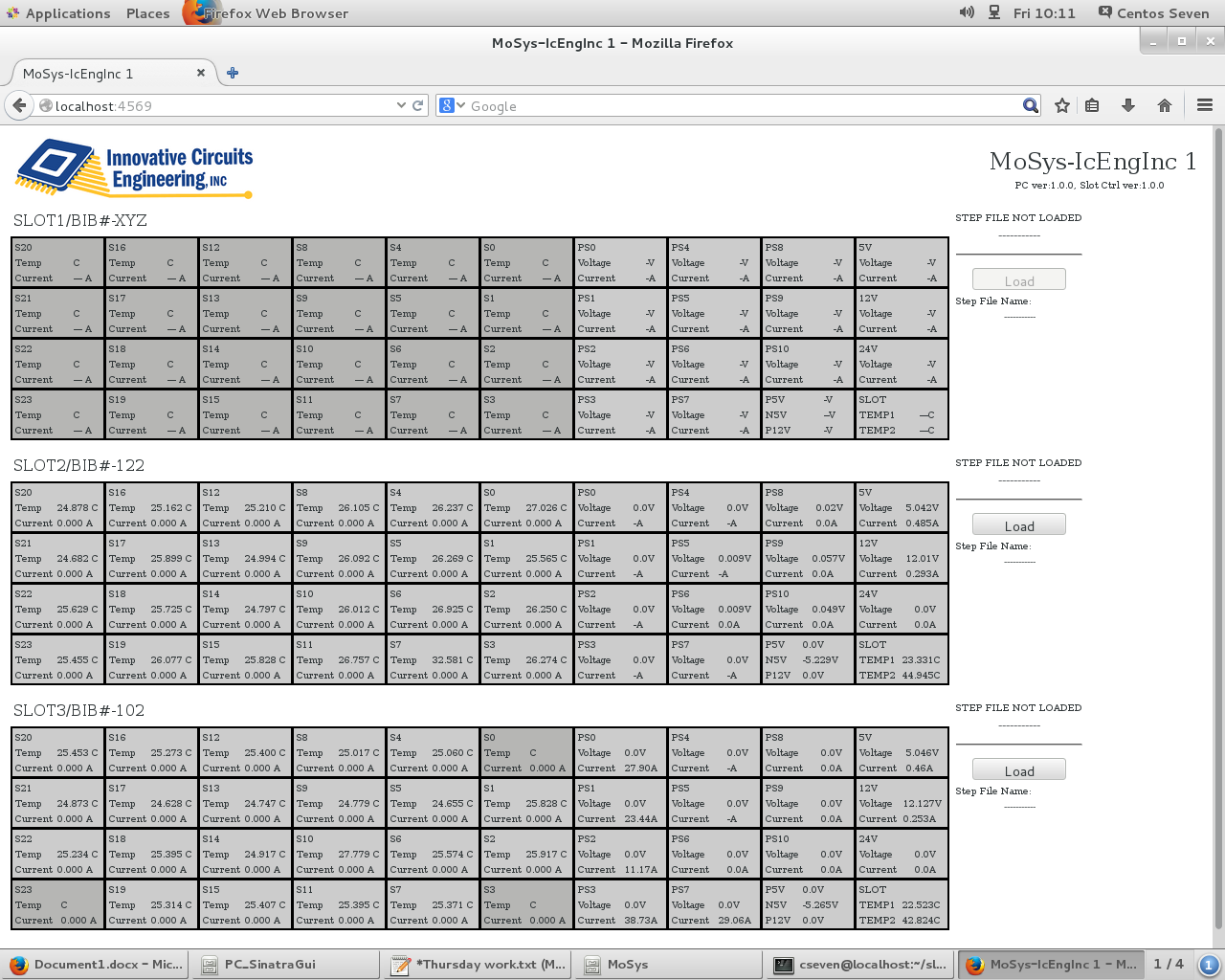
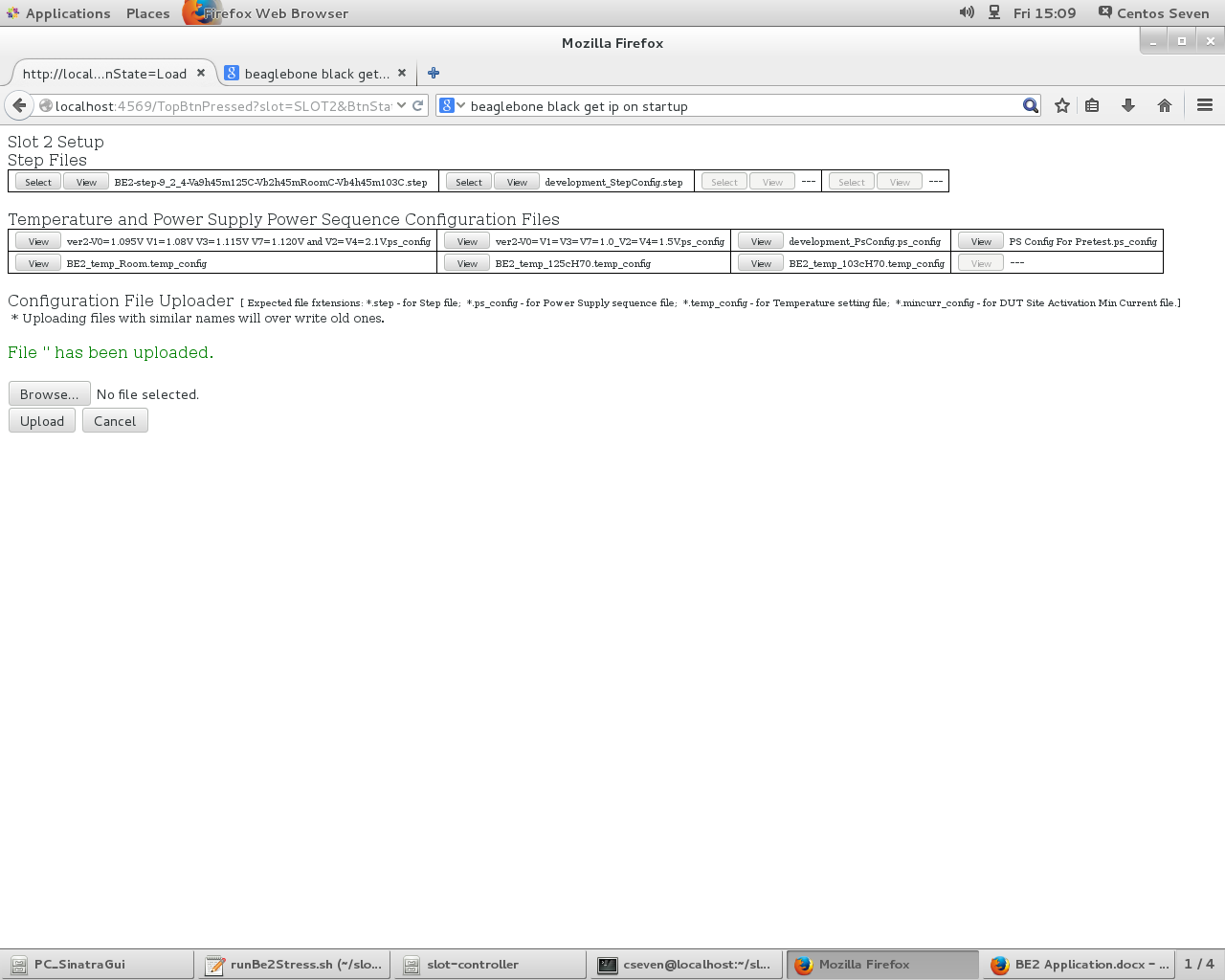
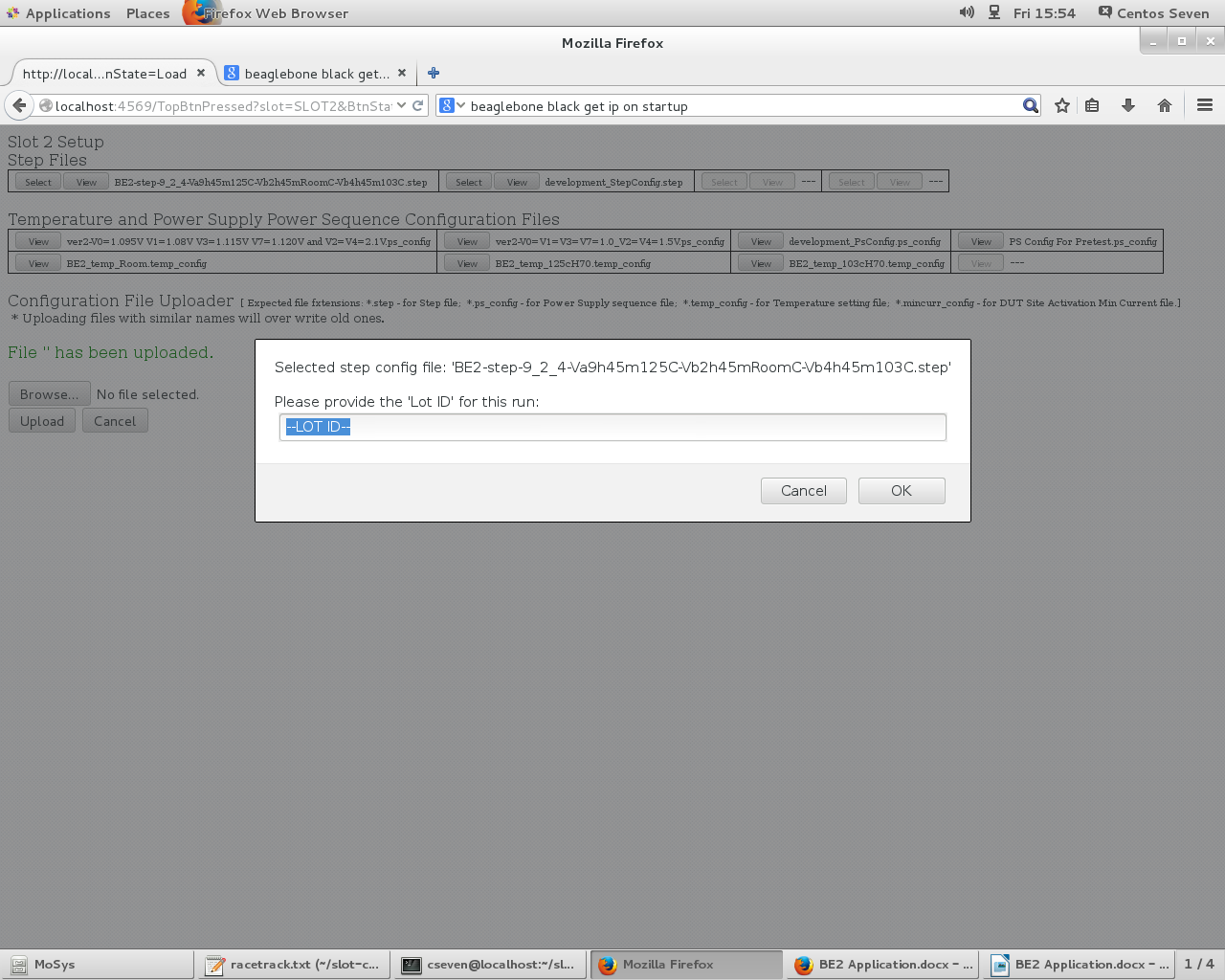
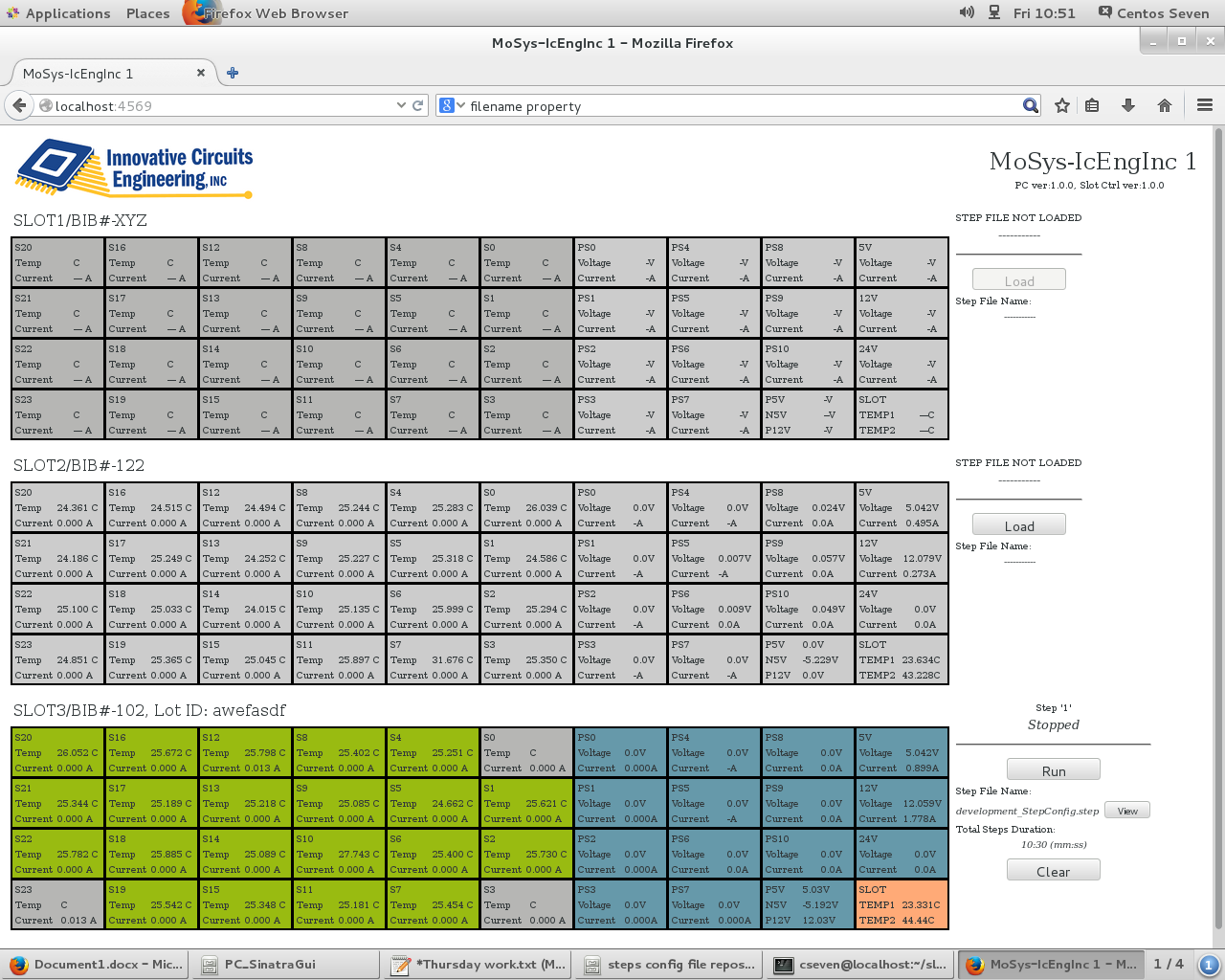
Running the BE2 application from boot up

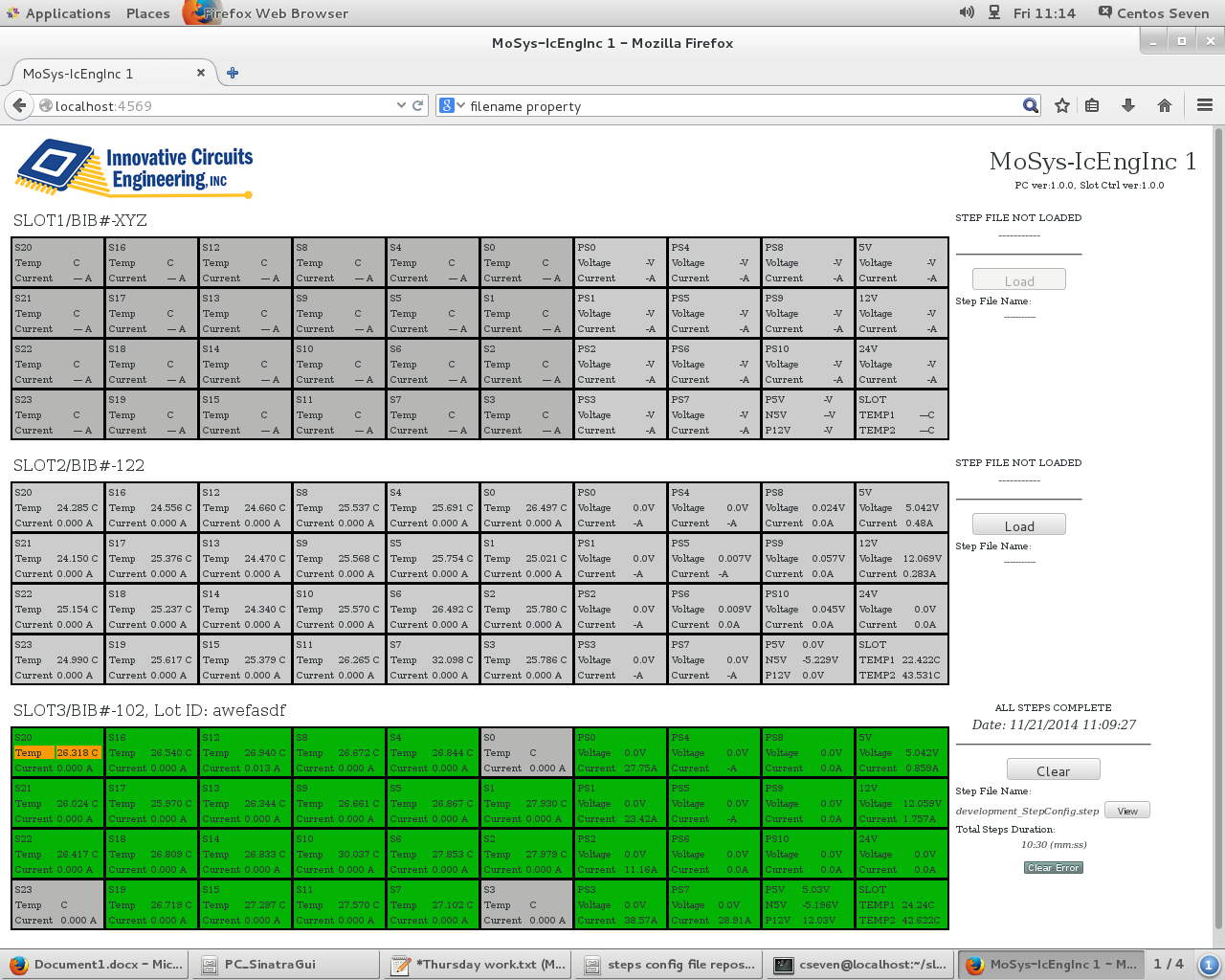
1. Ensure that Ethernet (em1) is "On".
   1. If it's not on, do the follow:
      1. Open the Network settings dialog box through this following menu:
      2. Applications->System Tools->Settings. Then select the "Network" icon.
      3. Once the Network settings dialog box is open, select "Ethernet (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/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.

PC Setup

* **\*\*Install Centos with GNOME Desktop option** ~30 minutes
* **After Centos installs, proceed to do a software update** ~30 minutes

1. When setting up the PC with Centos 7, make sure to install Apache HTTP Server.
   * 1. sudo yum install httpd mod\_ssl
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:
   * 1. First install git:

sudo yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel

1. Go inside the slot-controller folder:
   1. "cd ~"
   2. "git clone <https://github.com/icenginc/slot-controller>"

\* You need to provide a github account login info

Notes: To show git command verbose:

“GIT\_CURL\_VERBOSE=1 git clone <https://github.com/icenginc/slot-controller>"

Notes: If clone doesn’t work, may need to increase buffer size:

“git config –global http.postBuffer 524288000”

* 1. "cd slot-controller"

1. 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"
2. 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.
3. Get the needed files to run Ruby scripts:
   1. Install 'ruby'
      1. sudo yum install ruby
      2. sudo yum install gcc gcc-c++ 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 --system
      5. sudo gem update
   2. Install 'rack'
      1. gem install rack
   3. Install 'grape'
      1. gem install grape
   4. Install ‘curlftpfs’

yum install glib2-devel

yum install fuse-devel

yum install libcurl-devel

wget http://sourceforge.net/projects/curlftpfs/files/latest/download

tar xvzf curlftpfs-0.9.2.tar.gz

cd curlftpfs-0.9.2

./configure

make

make install

**\*\***If above wget doesn't work, download curlftpfs-0.9.2.tar.gz from website: <http://sourceforge.net/projects/curlftpfs/files/>

It will download to folder ~/Downloads

From there:

tar xvzf curlftpfs-0.9.2.tar.gz

cd curlftpfs-0.9.2

./configure

make

make install

* 1. Create the Port2Interface.so file
     1. pushd ./lib/”BBB\_GPIO2 Interface Ruby”
     2. ruby extconf.rb
     3. make
     4. popd
  2. Install the listed files by running them on the terminal
     1. gem install sinatra
     2. gem install sinatra-contrib
     3. gem install rest-client -v ‘1.7.2’
  3. Per MoSys, install tcl tk.
     1. sudo yum install tcl.x86\_64 tk.x86\_64
  4. Setup em1 (Network Interface Controller) as a static IP: 192.168.121.1

Gateway: 192.168.121.0

Netmask: 255.255.255.0

* 1. Setup em2 (NIC) as DHCP (should have IP Address similar to 192.168.1.x)
  2. 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

\* If blacklist.conf file doesn’t exist, create a new file using:

Ex: sudo vim blacklist.conf

* 1. Add crontab jobs by typing:

Ex: crontab –e

Add the following lines:

\*/1 \* \* \* \* bash /home/be2/slot-controller/PC\_SinatraGui/runSinatra.sh

\*/1 \* \* \* \* bash /home/be2/slot-controller/lib/DRbSharedMemory/runPcSharedMemory.sh

\*/1 \* \* \* \* bash /home/be2/slot-controller/scripts/find6MonOldFiles.sh

* 1. Have Apache run at startup:

sudo chkconfig --levels 235 httpd on

* 1. Add missing folders:
     1. /mnt/ftpMosys

Ex: in /mnt -> sudo mkdir ftpMosys

* 1. Add missing file: setFtpMosys
     1. /etc/rc.d/init.d/setFtpMosys

Add:

sudo mount -a

sudo mount /mnt/ftpMosys

* 1. Edit /etc/fstab file and add:

curlftpfs#Mosys:mosys123@192.168.1.4/ /mnt/ftpMosys fuse allow\_other,rw,user,nonempty,noauto 0 0

* 1. Turn off iptables:

sudo service iptables stop

sudo chkconfig iptables off

* 1. Turn off firewalld:

sudo systemctl disable firewalld

sudo systemctl stop firewalld

* 1. Copy “slot-controller repository” folder (which contains step files and config files for the slot run) into the folder “slot-controller\_data”
  2. Also copy the Pc\_SlotCtrlIps.config file from USB or previous Centos PC to the folder slot-controller\_data
  3. Setup a Samba connection to the ftp:
     1. Connect to a network
     2. smb://ftp\_server/mosys
     3. Login: innovative/ice
     4. Password: ice1234
* Reboot PC

1. Add BBB IP Address to list of known hosts by SSHing into each BBB one at a time. A prompt will ask you if you want to add this IP Address to the list of known hosts. Type “y” or “yes”

Example: ssh root@192.168.121.2

1. May need to change some file path naming (due to different pc names in filepath). You can check for errors in /var/spool/mail/*PC\_NAME*

\* where *PC\_NAME* is the name of the pc (ex: be2-3, be2-4..etc)

1. Check if localhost:4569 is up and running
2. Run the BE2 script in a Terminal:
   * 1. cd ~
     2. cd slot-controller
     3. bash runBe2Stress.sh

-> Look for any errors/warnings in the Terminal

\* If localhost:4569 is running but no communication to the BBB is present:

1) check if iptables and firewalld is turned off (step p, q above)

2) check that port 9292 is open using host 0.0.0.0

netstat –lnp | grep 9292

3) check that json is version v1.8.1 and v1.7.7

gem list

* If incorrect versions, uninstall incorrect version and install correct versions

Info:

Check that json v1.8.1 and v1.7.7 is installed:

gem list <- This command will show all the installed modules

To uninstall a module: sudo gem uninstall MODULE\_NAME

To install a module: sudo gem install MODULE\_NAME

The PC communicates to the BBB using Grape on port 9292

rackup needs to use host (0.0.0.0) and port 9292

check by typing: netstat –lnp | grep 9292

make sure port 9292 is bound to 0.0.0.0 and not 127.0.0.1

Search for words:

grep –rnw . –e “WORD”

Change PC domain name in /etc/host

Slot Control Setup (On the Beagle Bone Black, BBB)

1. 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.
2. 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"
3. Download the ruby codes that will run the BE2 application for the slot controller:
   1. Install cloud9

Ex: curl -L https://raw.githubusercontent.com/c9/install/master/install.sh | bash

Or

#### Ex: Installation (https://github.com/c9/core/blob/master/README.md)

Follow these steps to install the SDK:

git clone git://github.com/c9/core.git c9sdk

cd c9sdk

scripts/install-sdk.sh

To update the SDK to the latest version run:

git pull origin master

scripts/install-sdk.sh

Please note that Cloud9 v3 currently requires Node.js 0.12 or 0.10.

sudo yum install epel-release

sudo yum install nodejs

sudo yum install npm

* 1. "cd /var/lib/cloud9"
  2. "git clone <https://github.com/icenginc/slot-controller>"
  3. "cd slot-controller"

1. 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>

**The following commands are executed using the cloud9 IDE on the BBB**

* 1. Open a terminal tab, and run "sudo \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.
  2. Once the script which requests for an ID from a server is executed, execute all these scripts in order inside the "slot-controller" directory (/var/lib/cloud9/slot-controller):
     1. sudo gem update
     2. sudo aptitude install ruby1.9.1-dev
     3. sudo gem install rack
     4. sudo gem install grape
     5. sudo apt-get install libsqlite3-dev
     6. sudo gem install rest-client
     7. sudo \curl -sSL https://get.rvm.io | bash -s stable --ruby
     8. sudo pushd ./lib/BBB\_GPIO2 Interface Ruby
     9. sudo ruby extconf.rb
     10. sudo make
     11. sudo popd
     12. sudo gem install beaglebone
     13. # Set the time zone  
         sudo ln -sf /usr/share/zoneinfo/America/Los\_Angeles /etc/localtime

1. After installing all items above, make sure json version is 1.8.1
   * 1. Check by:

sudo gem list

* + 1. Uninstall by: (for example: json 1.8.3 is installed)

sudo gem uninstall json –v ‘1.8.3’

* + 1. Install by:

sudo gem install json –v ‘1.8.1’

1. Setup the IP address of the BBB/Slot-Controller:
   1. **The below code (IP Address Setup) is outdated.**

\*The new BBB code uses the DIP switches on the Slot board in order to configure an IP Address

Using vi editor, make sure to have the following items in the /etc/network/interfaces  
auto eth0  
 iface eth0 inet static  
 address 192.168.121.2  
 netmask 255.255.255.0  
 network 192.168.121.0  
 gateway 192.168.121.0

1. Setup the HDMI so Mode 7 will be available for the GPIO pins by doing the following terminal shell commands that are surounded 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"
2. 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" folder of the PC.

Slot Controller Configuration To Communicate with the PC (Configure on BBB)

**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:

"/var/lib/cloud9/slot-controller/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@192.168.121.2" if the board that is to be accessed is on the IP address 192.168.121.2. 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.

\* The PC local NIC IP Address will always be:  **192.168.121.1**