PIP-II LLRF - Quality Control and HW & SW Documentation

Shrividhyaa Sankar Raman

PIP-II LLRF Final Design Review

July 17, 2024

A Partnership of:

US/DOE

India/DAE

Italy/INFN

UK/UKRI-STFC

France/CEA, CNRS/IN2P3

Poland/WUST

Scope

- Introduction
- QA/QC Scope
- Software Management
  - Redmine Repository
  - GitLab Repositories – for the collaboration
- Documentation
  - FNAL Firmware/Software
  - LBNL Firmware/Software
- QA/QC and Safety
  - General QA Plan
  - Bench Test Procedure
- Process through Final Design
- Summary

Introduction – About me

- Engineer, AD-RF-LLRF
- M.S. Electrical Engineering
- Fermilab experience for 5 years.
- Front-End Software Development
- SOC FPGA - Linux OS/Firmware
- Control Systems Interfaces - ACNET, Labview, EPICS
- Testing and Validation of LLRF Systems.
- Git Repository and Code Management for Front-End Software / FPGA Firmware / VxWorks Projects.
- Code Documentation using Doxygen.

QA/QC Scope

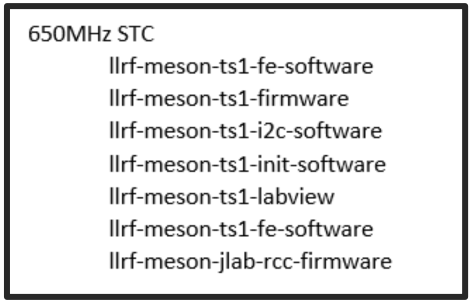
- The LLRF System Quality Control Plan adheres to the Accelerator Systems (L2) and the PIP-II Quality Assurance Plans (DocDB#4805 and DocDB#142 respectively).
- It aligns with the Project Management Plan for the PIP-II Project.
- The LLRF systems required for the PIP-II Project is comprised of both hardware and software deliverables all residing within WBS.121.03.04.
- The LLRF System QA Plan reflects the systems, controls, and measures incorporated by the PIP-II Project to manage, plan, assess, and improve processes to deliver operational and scientific excellence in a consistent environment with minimal risk.
- This LLRF Systems QC Plan (QCP) covers the various acceptance and testing steps required to ensure these deliverables are provided to the project while meeting the required specifications.

Quality Assurance and Reliability of Software

- Software Management processes are crucial for system reliability and maintainability.
- Code Repositories in software management tools such as GitHub and GitLab allow changes in Firmware and Software to be deployed with minimal disruption to operations.
- Documentation of firmware / software is essential for continuous development and improvement.
- Good documentation facilitates wider firmware / software review and evaluation, thereby ensuring quality.

Software Management

- Redmine Git Repository:
  - Redmine is a flexible project management web application.
  - We have a Redmine repository for each Project.
  - Projects have subprojects with their own repositories for the various software / firmware components.
  - An example project is shown below.

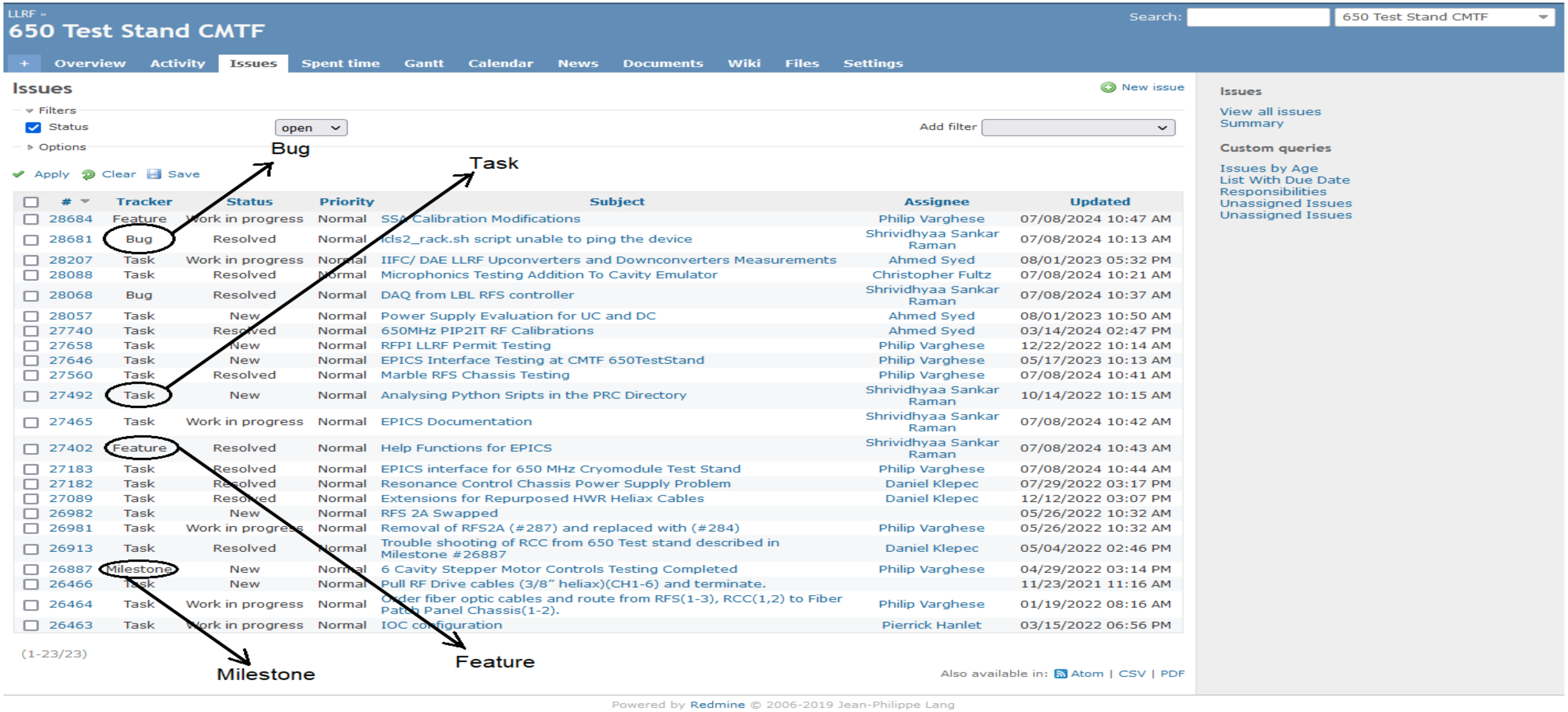


650MHz STC

- llrf-meson-ts1-fe-software
- llrf-meson-ts1-firmware
- llrf-meson-ts1-i2c-software
- llrf-meson-ts1-init-software
- llrf-meson-ts1-labview
- llrf-meson-ts1-fe-software
- llrf-meson-jlab-rcc-firmware

Project Management Tools - Redmine

Redmine is also being used as a project management tool to track Issues, Tasks and Milestones. An Example is as shown below:



□	#	Tracker	Status	Priority	Subject	Assignee	Updated
□	28684	Feature	Work in progress	Normal	SSA Calibration Modifications	Philip Varghese	07/08/2024 10:47 AM
□	28681	Bug	Resolved	Normal	lcls2_rack.sh script unable to ping the device	Shrividhyaa Sankar Raman	07/08/2024 10:13 AM
□	28207	Task	Work in progress	Normal	IIFC/ DAE LLRF Upconverters and Downconverters Measurements	Ahmed Syed	08/01/2023 05:32 PM
□	28088	Task	Resolved	Normal	Microphonics Testing Addition To Cavity Emulator	Christopher Fultz	07/08/2024 10:21 AM
□	28068	Bug	Resolved	Normal	DAQ from LBL RFS controller	Shrividhyaa Sankar Raman	07/08/2024 10:37 AM
□	28057	Task	New	Normal	Power Supply Evaluation for UC and DC	Ahmed Syed	08/01/2023 10:50 AM
□	27740	Task	Resolved	Normal	650MHz PIP2IT RF Calibrations	Ahmed Syed	03/14/2024 02:47 PM
□	27658	Task	New	Normal	RFPI LLRF Permit Testing	Philip Varghese	12/22/2022 10:14 AM
□	27646	Task	New	Normal	EPICS Interface Testing at CMTF 650TestStand	Philip Varghese	05/17/2023 10:13 AM
□	27560	Task	Resolved	Normal	Marble RFS Chassis Testing	Philip Varghese	07/08/2024 10:41 AM
□	27492	Task	New	Normal	Analysing Python Sripts in the PRC Directory	Shrividhyaa Sankar Raman	10/14/2022 10:15 AM
□	27465	Task	Work in progress	Normal	EPICS Documentation	Shrividhyaa Sankar Raman	07/08/2024 10:42 AM
□	27402	Feature	Resolved	Normal	Help Functions for EPICS	Shrividhyaa Sankar Raman	07/08/2024 10:43 AM
□	27183	Task	Resolved	Normal	EPICS interface for 650 MHz Cryomodule Test Stand	Philip Varghese	07/08/2024 10:44 AM
□	27182	Task	Resolved	Normal	Resonance Control Chassis Power Supply Problem	Daniel Klepec	07/29/2022 03:17 PM
□	27089	Task	Resolved	Normal	Extensions for Repurposed HWR Heliax Cables	Daniel Klepec	12/12/2022 03:07 PM
□	26982	Task	New	Normal	RFS 2A Swapped		05/26/2022 10:32 AM
□	26981	Task	Work in progress	Normal	Removal of RFS2A (#287) and replaced with (#284)	Philip Varghese	05/26/2022 10:32 AM
□	26913	Task	Resolved	Normal	Trouble shooting of RCC from 650 Test stand described in Milestone #26887	Daniel Klepec	05/04/2022 02:46 PM
□	26887	Milestone	New	Normal	6 Cavity Stepper Motor Controls Testing Completed	Philip Varghese	04/29/2022 03:14 PM
□	26466	Task	New	Normal	Pull RF Drive cables (3/8" heliax)(CH1-6) and terminate.		11/23/2021 11:16 AM
□	26464	Task	Work in progress	Normal	Order fiber optic cables and route from RFS(1-3), RCC(1,2) to Fiber Patch Panel Chassis(1-2).	Philip Varghese	01/19/2022 08:16 AM
□	26463	Task	Work in progress	Normal	IOC configuration	Pierrick Hanlet	03/15/2022 06:56 PM

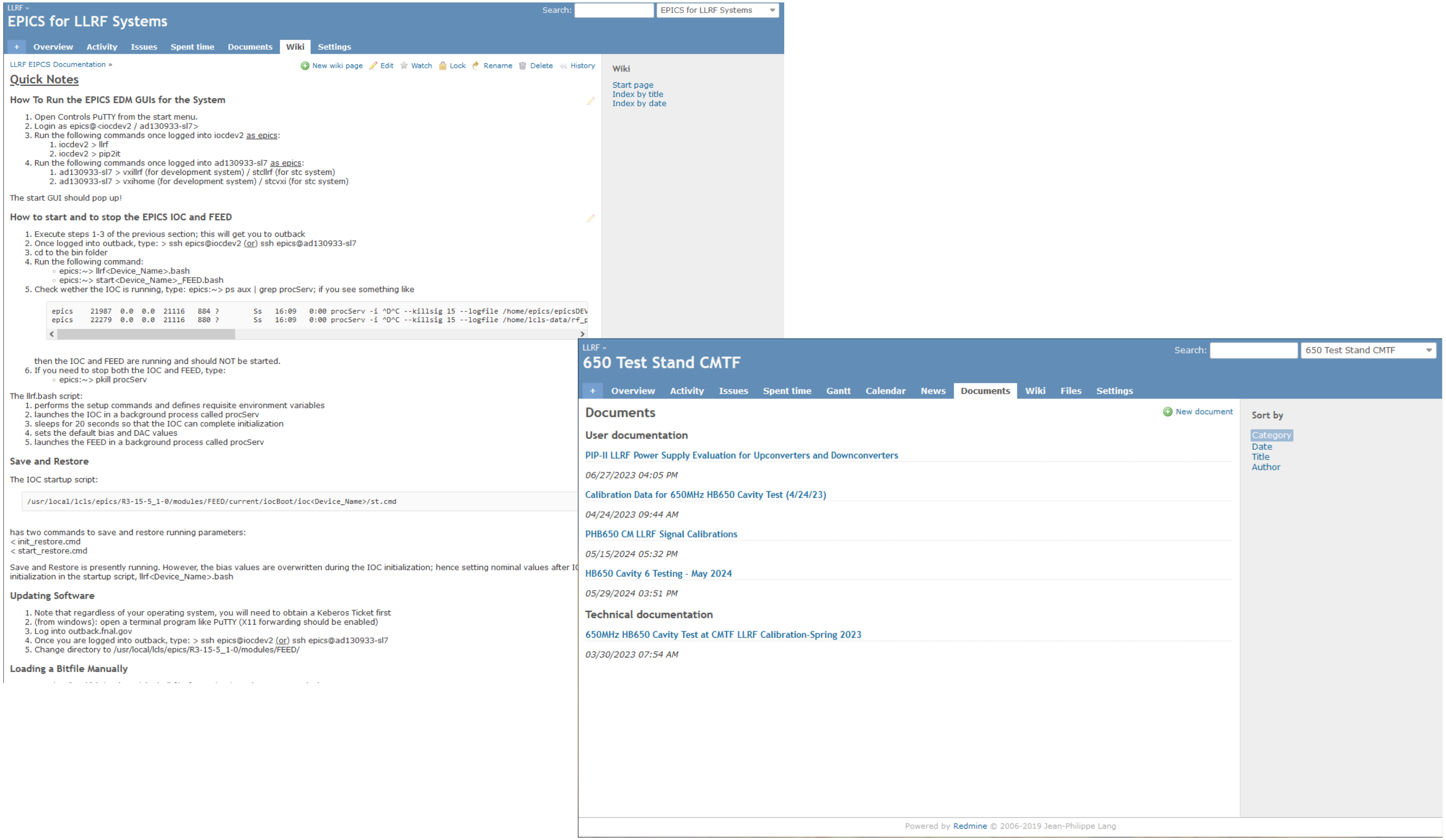
(1-23/23)

Also available in: Atom | CSV | PDF

Powered by Redmine © 2006-2019 Jean-Philippe Lang

Project Management Tool - Redmine

Internal notes and documents are stored in the Wiki and Documents Tabs



Quick Notes

How To Run the EPICS EDM GUIs for the System

1. Open Controls PuTTY from the start menu.
2. Login as epics@<iocdev2 / ad130933-sl7>
3. Run the following commands once logged into iocdev2 as epics:
   1. iocdev2 > llrf
   2. iocdev2 > pip2it
4. Run the following commands once logged into ad130933-sl7 as epics:
   1. ad130933-sl7 > vxillrf (for development system) / stcllrf (for stc system)
   2. ad130933-sl7 > vxihome (for development system) / stcvxi (for stc system)

The start GUI should pop up!

How to start and to stop the EPICS IOC and FEED

1. Execute steps 1-3 of the previous section; this will get you to outback
2. Once logged into outback, type: > ssh epics@iocdev2 (or) ssh epics@ad130933-sl7
3. cd to the bin folder
4. Run the following command:
   - epics:~> llrf<Device_Name>.bash
   - epics:~> start<Device_Name>_FEED.bash
5. Check wether the IOC is running, type: epics:~> ps aux | grep procServ; if you see something like

```
epics    21987  0.0  0.0  21116   884 ?        Ss   16:09   0:00 procServ -i ^D^C --killsig 15 --logfile /home/epics/epicsDEV
epics    22279  0.0  0.0  21116   880 ?        Ss   16:09   0:00 procServ -i ^D^C --killsig 15 --logfile /home/lcls-data/rf_p
```

   then the IOC and FEED are running and should NOT be started.
6. If you need to stop both the IOC and FEED, type:
   - epics:~> pkill procServ

The llrf.bash script:
1. performs the setup commands and defines requisite environment variables
2. launches the IOC in a background process called procServ
3. sleeps for 20 seconds so that the IOC can complete initialization
4. sets the default bias and DAC values
5. launches the FEED in a background process called procServ

Save and Restore

The IOC startup script:

```
/usr/local/lcls/epics/R3-15-5_1-0/modules/FEED/current/iocBoot/ioc<Device_Name>/st.cmd
```

has two commands to save and restore running parameters:
< init_restore.cmd
< start_restore.cmd

Save and Restore is presently running. However, the bias values are overwritten during the IOC initialization; hence setting nominal values after IO
initialization in the startup script, llrf<Device_Name>.bash

Updating Software

1. Note that regardless of your operating system, you will need to obtain a Keberos Ticket first
2. (from windows): open a terminal program like PuTTY (X11 forwarding should be enabled)
3. Log into outback.fnal.gov
4. Once you are logged into outback, type: > ssh epics@iocdev2 (or) ssh epics@ad130933-sl7
5. Change directory to /usr/local/lcls/epics/R3-15-5_1-0/modules/FEED/

Loading a Bitfile Manually

Documents

User documentation

PIP-II LLRF Power Supply Evaluation for Upconverters and Downconverters

06/27/2023 04:05 PM

Calibration Data for 650MHz HB650 Cavity Test (4/24/23)

04/24/2023 09:44 AM

PHB650 CM LLRF Signal Calibrations

05/15/2024 05:32 PM

HB650 Cavity 6 Testing - May 2024

05/29/2024 03:51 PM

Technical documentation

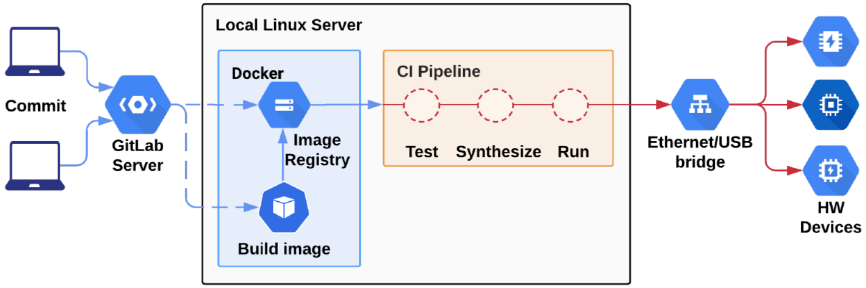
650MHz HB650 Cavity Test at CMTF LLRF Calibration-Spring 2023

03/30/2023 07:54 AM

Powered by Redmine © 2006-2019 Jean-Philippe Lang

Software Management

- GitLab
  - Continuous integration puts a great emphasis on testing automation to check that the application is not broken whenever new commits are integrated into the main branch.
  - Used by LBNL Collaboration.



- GitHub
  - Alternate code repository that Fermilab is moving towards.
  - Similar to GitLab
  - Manages software projects with features like issue tracking and code management.

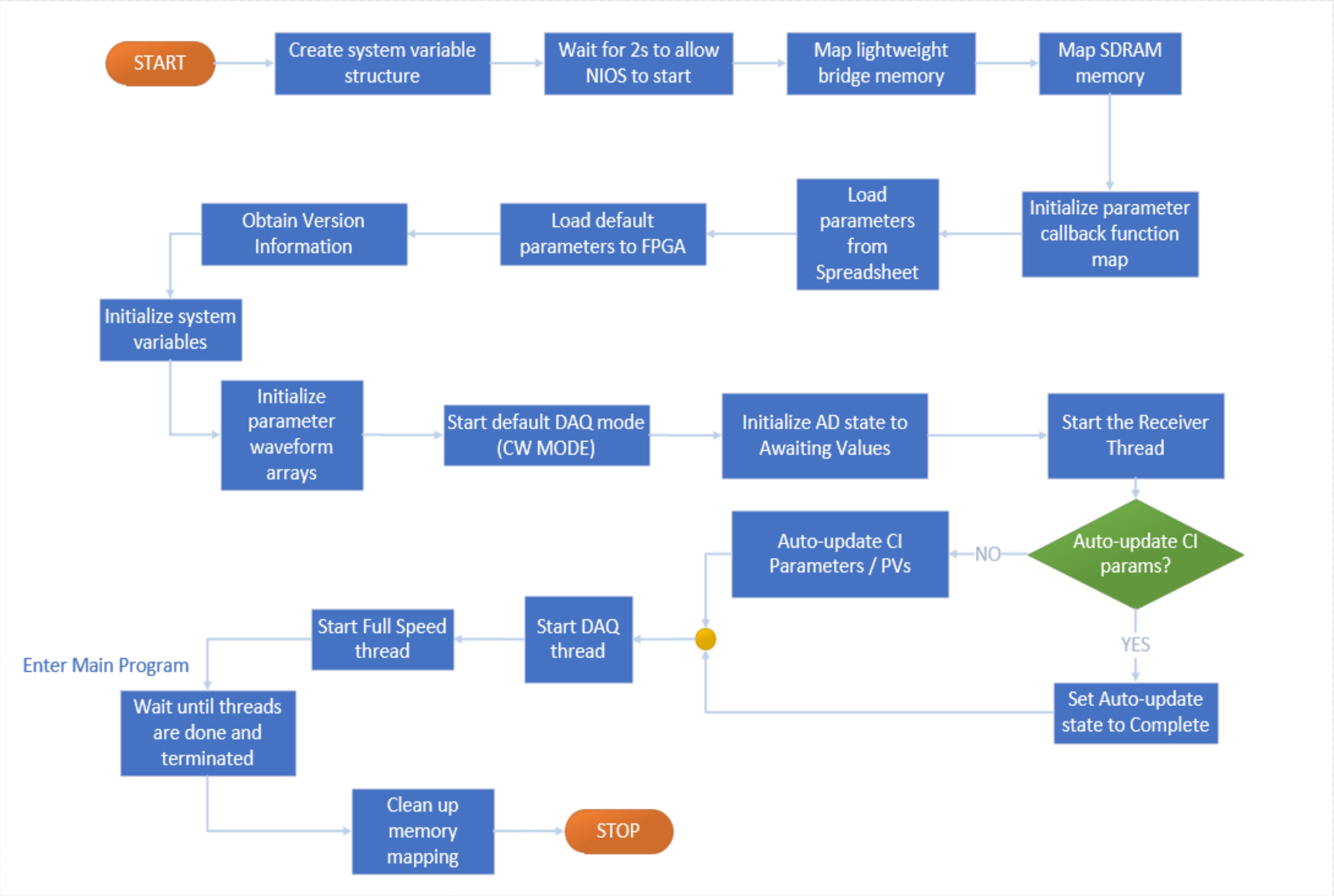
Documentation

- Project Requirements
- Functional and Technical Requirements
- Design Documents
- Schematics, Software Flow-charts Documentation
- Test Procedures for individual hardware / software deliverables
- System Level Test Procedures (User Interface, Parameter / Control Settings, etc.)
- Quality Control (Test Results and Requirements Validation)

SYSTEM	DOCUMENT ID
LLRF FRS	ED0004194
Resonance Control System TRS	ED0005782
4-Channel Upconverter FRS	ED0004189
4-Channel Upconverter TRS	ED0005163
8-Channel Downconverter FRS	ED0004190
8-Channel Downconverter TRS	ED0005166
Mater Oscillator/ Reference Line System FRS	ED0005057

Software Documentation - Flowcharts

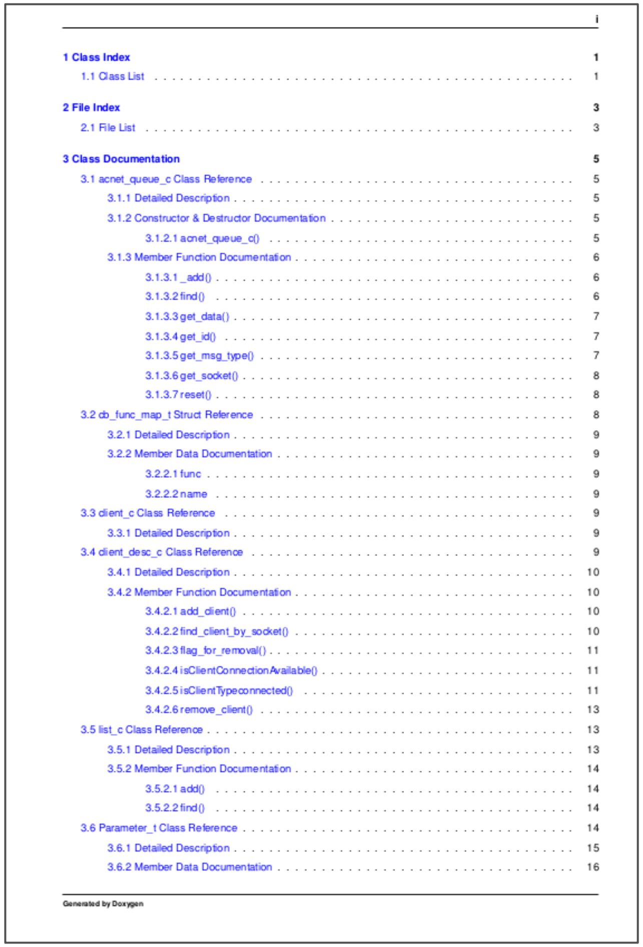
Front-End Initialization Sample


START
Create system variable structure
Wait for 2s to allow NIOS to start
Map lightweight bridge memory
Map SDRAM memory
Initialize parameter callback function map
Load parameters from Spreadsheet
Load default parameters to FPGA
Obtain Version Information
Initialize system variables
Initialize parameter waveform arrays
Start default DAQ mode (CW MODE)
Initialize AD state to Awaiting Values
Start the Receiver Thread
Auto-update CI params?
NO
Auto-update CI Parameters / PVs
YES
Set Auto-update state to Complete
Start DAQ thread
Start Full Speed thread
Enter Main Program
Wait until threads are done and terminated
Clean up memory mapping
STOP

Software Documentation - Doxygen

- Doxygen is a tool for generating documentation from annotated C/C++ sources.

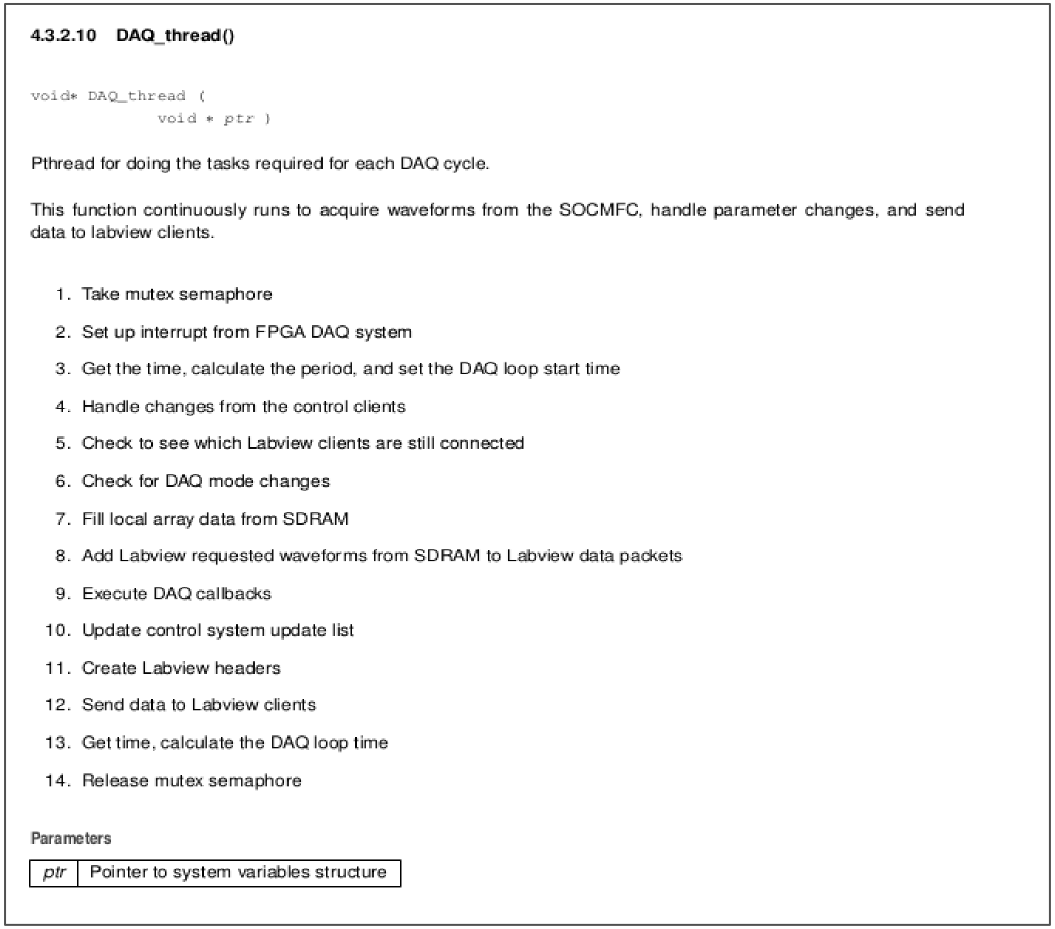
Generates a detailed Software Documentation



i

Generated by Doxygen

Example of a function description



4.3.2.10 DAQ_thread()

```
void* DAQ_thread (
            void * ptr )
```

Pthread for doing the tasks required for each DAQ cycle.

This function continuously runs to acquire waveforms from the SOCMFC, handle parameter changes, and send data to labview clients.

1. Take mutex semaphore
2. Set up interrupt from FPGA DAQ system
3. Get the time, calculate the period, and set the DAQ loop start time
4. Handle changes from the control clients
5. Check to see which Labview clients are still connected
6. Check for DAQ mode changes
7. Fill local array data from SDRAM
8. Add Labview requested waveforms from SDRAM to Labview data packets
9. Execute DAQ callbacks
10. Update control system update list
11. Create Labview headers
12. Send data to Labview clients
13. Get time, calculate the DAQ loop time
14. Release mutex semaphore

Parameters

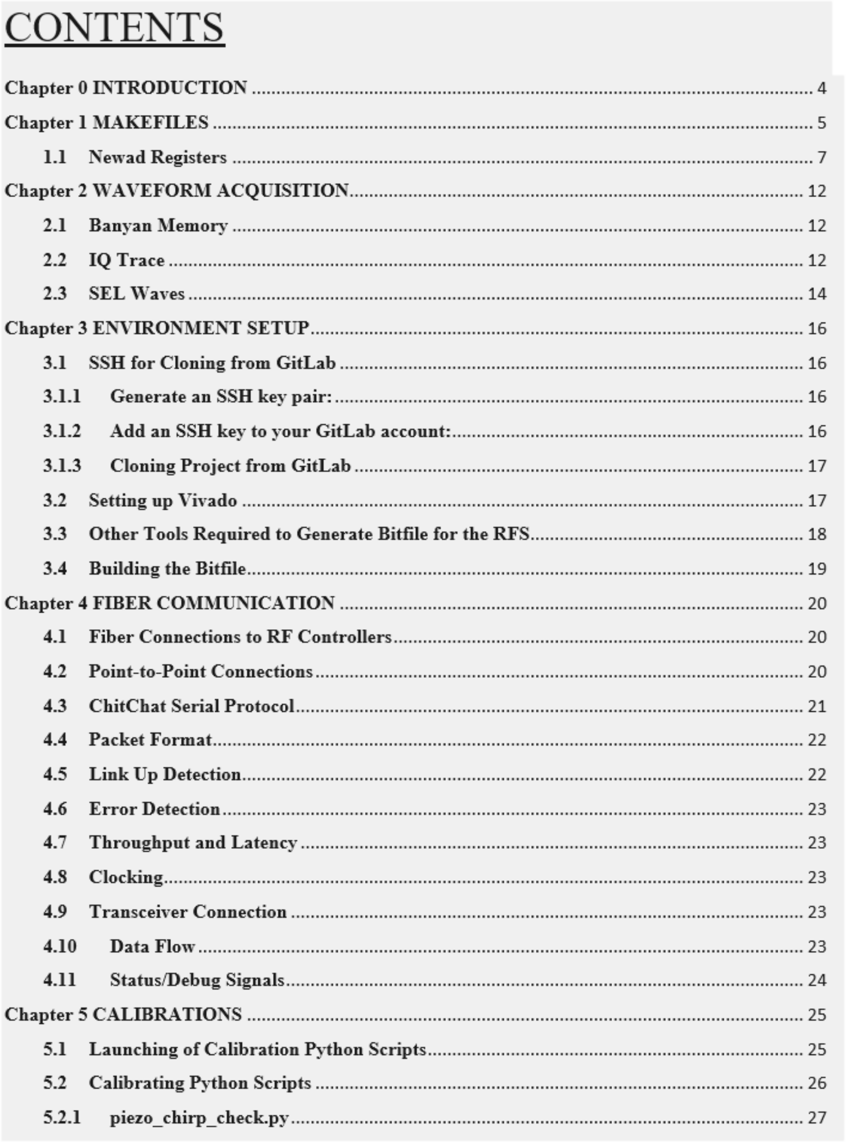
ptr	Pointer to system variables structure

Documenting LBNL Code

- Weekly meetings are held with the team at LBNL, to review firmware/software structure and details to develop complete documentation
- These sessions are mainly aimed at real-time debugging, code walkthroughs and documentation updates.
- They provide benefits such as enhanced understanding of firmware logic, immediate issue resolution, and continuous improvement of documentation.
- It also provides training to FNAL staff to be able to take ownership and provide for future development and code maintainance.

Documenting LBNL Code

- The document generated is constantly updated with new concepts discussed every week and better understanding of the code.
- A picture of some of the topics covered in this draft is shown below.



CONTENTS

Quality Control Plan

- Quality Control Plan is a very important document that ensures production quality.
- To maintain the quality of production following steps will be incorporated for the production process.
  - Inspection of the raw material and components, rejection, and replacement
  - Appropriate Storage
  - Physical inspection at every stage of sub systems assembly
  - Physical and electrical testing for quality check
  - Qualification of complete chassis
  - Functional testing after all the qualification tests
  - QC Document Finalized#: PIP-II-doc-5496

Bench Testing Procedure

- Control systems interface
  - Verify all PVs on EPICS screens are communicating with the chassis appropriately. Read and write functions for PVs should be checked on bench and prior to usage of chassis at PIP-II.
- Software functionality
  - Key features such as calibration / control settings, waveform acquisition and protection features such as RF overdrive / Quench detection.
- Drive Signal testing for timing, amplitude and jitter.
- Triggers and monitors using oscilloscope by connecting to monitor channels.
- Automation using scripting languages maybe helpful for consistent and repeatable testing of various modules.

Safety

- The system shall abide by all Fermilab ES&H (FESHM) and all Fermilab Radiological Control Manual (FRCM) requirements including but not limited to:
  - Electrical Safety
    - FESHM Chapter 9110 Electrical Utilization Equipment Safety
    - FESHM Chapter 9160 Low Voltage, High Current Power Distribution Systems
    - FESHM Chapter 9190 Grounding Requirements for Electrical Distribution and Utilization Equipment
  - Radiation Safety
    - FRCM Chapter 8 ALARA Management of Accelerator Radiation Shielding
    - FRCM Chapter 10 Radiation Safety Interlock Systems
    - FRCM Chapter 11 Environmental Radiation Monitoring and Control
  - General Safety
    - FESHM Chapter 2000 Planning for Safe Operations
    - Follow LOTO Procedures

Process through Final Design

- Support EPICS API for SoCFPGA architecture.
- Support the transition from Redmine repositories to GitHub.
- Incorporate LBNL / LCLS-II scripting and code management features to Fermilab tools.
- Develop adequate documentation of all the tools, procedures and code base including system level information.
- Develop test benches to validate all code components.

Summary

- Currently using Redmine as the code management repository.
- Documenting all the steps, observations and results, throughout the project are an essential part of project completion.
- Follow QA/QC Plan to adhere to all Quality procedures for all LLRF Systems.
- IOC/ EPICS interface for PIP-II controls.

Thank You

Backup Slides

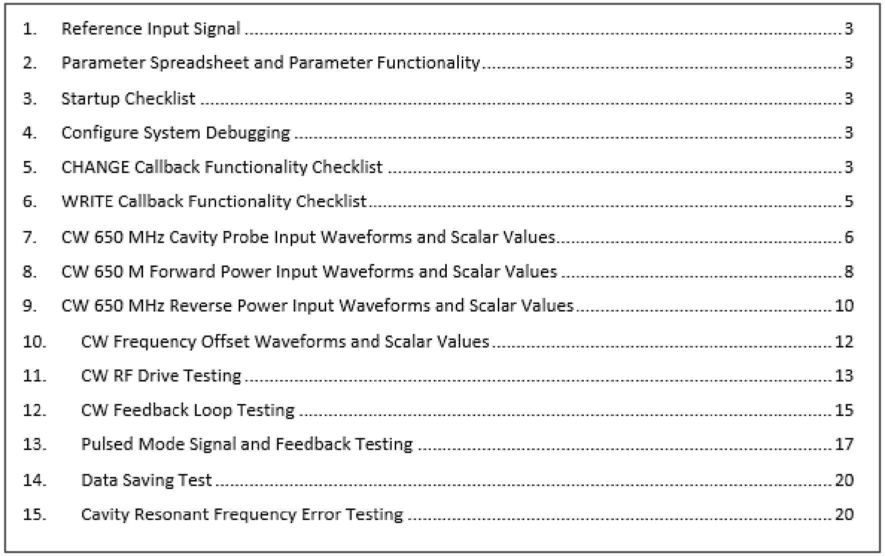
PIP-II LINAC Code Organization

- PIP-II LINAC LLRF Systems are organized into seven projects areas:
  1. RFQ
  2. Bunchers
  3. HWR
  4. SSR1
  5. SSR2
  6. LB650
  7. HB650
- While there is a lot of commonality in firmware and software content, they may need to be distinct due to differences in tuners and resonance control.

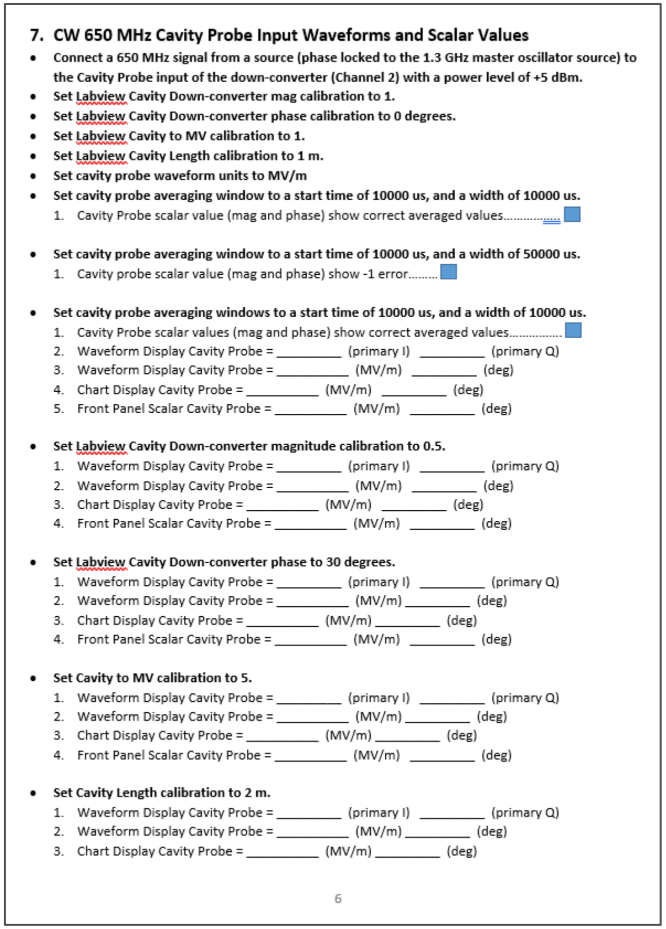
System Level Testing

STC650 Test Stand LLRF System Checklist

Checklist Items



Sample of a test procedure



7. CW 650 MHz Cavity Probe Input Waveforms and Scalar Values

- **Connect a 650 MHz signal from a source (phase locked to the 1.3 GHz master oscillator source) to the Cavity Probe input of the down-converter (Channel 2) with a power level of +5 dBm.**
- **Set Labview Cavity Down-converter mag calibration to 1.**
- **Set Labview Cavity Down-converter phase calibration to 0 degrees.**
- **Set Labview Cavity to MV calibration to 1.**
- **Set Labview Cavity Length calibration to 1 m.**
- **Set cavity probe waveform units to MV/m**
- **Set cavity probe averaging window to a start time of 10000 us, and a width of 10000 us.**
  1. Cavity Probe scalar value (mag and phase) show correct averaged values................ □

- **Set cavity probe averaging window to a start time of 10000 us, and a width of 50000 us.**
  1. Cavity probe scalar value (mag and phase) show -1 error......... □

- **Set cavity probe averaging windows to a start time of 10000 us, and a width of 10000 us.**
  1. Cavity Probe scalar values (mag and phase) show correct averaged values............... □
  2. Waveform Display Cavity Probe = ________ (primary I) ________ (primary Q)
  3. Waveform Display Cavity Probe = ________ (MV/m) ________ (deg)
  4. Chart Display Cavity Probe = ________ (MV/m) ________ (deg)
  5. Front Panel Scalar Cavity Probe = ________ (MV/m) ________ (deg)

- **Set Labview Cavity Down-converter magnitude calibration to 0.5.**
  1. Waveform Display Cavity Probe = ________ (primary I) ________ (primary Q)
  2. Waveform Display Cavity Probe = ________ (MV/m) ________ (deg)
  3. Chart Display Cavity Probe = ________ (MV/m) ________ (deg)
  4. Front Panel Scalar Cavity Probe = ________ (MV/m) ________ (deg)

- **Set Labview Cavity Down-converter phase to 30 degrees.**
  1. Waveform Display Cavity Probe = ________ (primary I) ________ (primary Q)
  2. Waveform Display Cavity Probe = ________ (MV/m) ________ (deg)
  3. Chart Display Cavity Probe = ________ (MV/m) ________ (deg)
  4. Front Panel Scalar Cavity Probe = ________ (MV/m) ________ (deg)

- **Set Cavity to MV calibration to 5.**
  1. Waveform Display Cavity Probe = ________ (primary I) ________ (primary Q)
  2. Waveform Display Cavity Probe = ________ (MV/m) ________ (deg)
  3. Chart Display Cavity Probe = ________ (MV/m) ________ (deg)
  4. Front Panel Scalar Cavity Probe = ________ (MV/m) ________ (deg)

- **Set Cavity Length calibration to 2 m.**
  1. Waveform Display Cavity Probe = ________ (primary I) ________ (primary Q)
  2. Waveform Display Cavity Probe = ________ (MV/m) ________ (deg)
  3. Chart Display Cavity Probe = ________ (MV/m) ________ (deg)

6