ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-WEPAB300

PYTHON BASED TOOLS FOR FRIB LLRF OPERATION AND MANAGEMENT*

S. Kunjir†, D. Morris, S. Zhao
Facility for Rare Isotope Beams, Michigan State University, East Lansing, MI, USA

Abstract

Some Python based tools have been developed at the Facility for Rare Isotope Beams (FRIB) for the ease of operation and management of the low level radio frequency (LLRF) controllers. Utilizing the rich features in Python, some tasks can be easily applied to a whole segment, one type of cryomodule (CM), a specific cryomodule or individual cavities grouped by a complex custom query. The tasks include, for example, 1) testing interface connections between various sub-systems prior to an operational run; 2) setting, checking and saving/restoring parameters during and after an operational run; 3) updating LLRF controller firmware and software during maintenance. With these tools, routine manual tasks are streamlined to achieve significantly greater efficiency in terms of scalability, time, memory and network resources. Considering channel access security, beam on/off status etc., the strategy of choosing either input/output controller (IOC) or Python for the implementation of certain tasks is also discussed in the paper.

INTRODUCTION

The Facility for Rare Isotope Beams at Michigan State University will be a scientific user facility for nuclear physics research [1]. The FRIB superconducting RF (SRF) linear accelerator (linac) includes four types of SRF cavities consisting of 104 quarter-wave resonators (QWR) and 220 half-wave resonators (HWR) hosted in a total of 46 cryomodules (CM). Table 1 shows a summary of different types of systems at FRIB.

Table 1: FRIB SRF Linac Systems

Linac System	System Type	# of CM	# of Cavities
Linear Segment 1 (LS1)	CA	3	12
Linear Segment 1	CB	11	88
Folding Segment 1 (FS1)	CH	1	4
Linear Segment 2 (LS2)	CC	12	72
Linear Segment 2	CD	12	96
Folding Segment 2 (FS2)	CG	1	4
Linear Segment 3 (LS3)	CD	6	48

The FRIB LLRF controllers are used for cavity testing, integrated tests and cryomodule tests. It interfaces with multiple systems including global timing system (GTS), fast protection system (FPS), low level control (LLC) system, experimental physics and industrial control system (EPICS) and the RF amplifier [2].

EPICS is used at FRIB as a preferred control system for interfacing various hardware sub-systems with input output controllers (IOC) and provides access to process variables (PV) via channel access (CA).

LLRF engineers at FRIB perform routine operational, maintenance and troubleshooting tasks before, during and after a run. These tasks require batch processing of PVs, which includes setting a particular group of PVs to a certain value, testing interface between different sub-systems, measuring output levels, saving PV values before or after an event, restoring PV values from historical data and updating firmware/software of LLRF controllers. An example of batch processing of PVs would be turning all amplifiers on or off.

A Python based tool has been developed at FRIB to automate manual operations, consequently saving time and resources, in turn increasing productivity. These tools will be discussed in this paper.

CS-STUDIO

CS-Studio, an Eclipse based collection of tools, is used by operators as an interface for monitoring, controlling and operating sub-systems. Figure 1 shows an example OPI for LLRF controllers used at FRIB. This OPI provides easy access to all parameters of an LLRF controller and has been used for cavity testing, CM testing and CM commissioning. Operators can interact with PVs and better visualize live and historical data in the CS-Studio environment.

Operating and managing over 300 cavities is less efficient, since CS-Studio lacks feature for batch processing. For example, to write a value to the same PV of multiple LLRF controllers, the operator has to open individual OPIs for each LLRF controller manually, that takes a significant amount of time.

Also, in the CS-Studio it is hard to target a group of LLRF controllers defined by a complex custom query. For example, the Save and Restore feature of CS-Studio cannot save or restore PV values grouped by a complex custom query. Reasons shown above are the motivation to develop LLRF expert tools.

FRIB LLRF EXPERT TOOLS

Python offers abundant convenient features for scientific and engineering programming [3]. PyEpics is an interface for the Channel Access (CA) library of the EPICS to the Python programming language. The PyEpics package provides a base EPICS module to Python with methods for

* This work is supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661.
† kunjir@frib.msu.edu.

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-WEPAB300

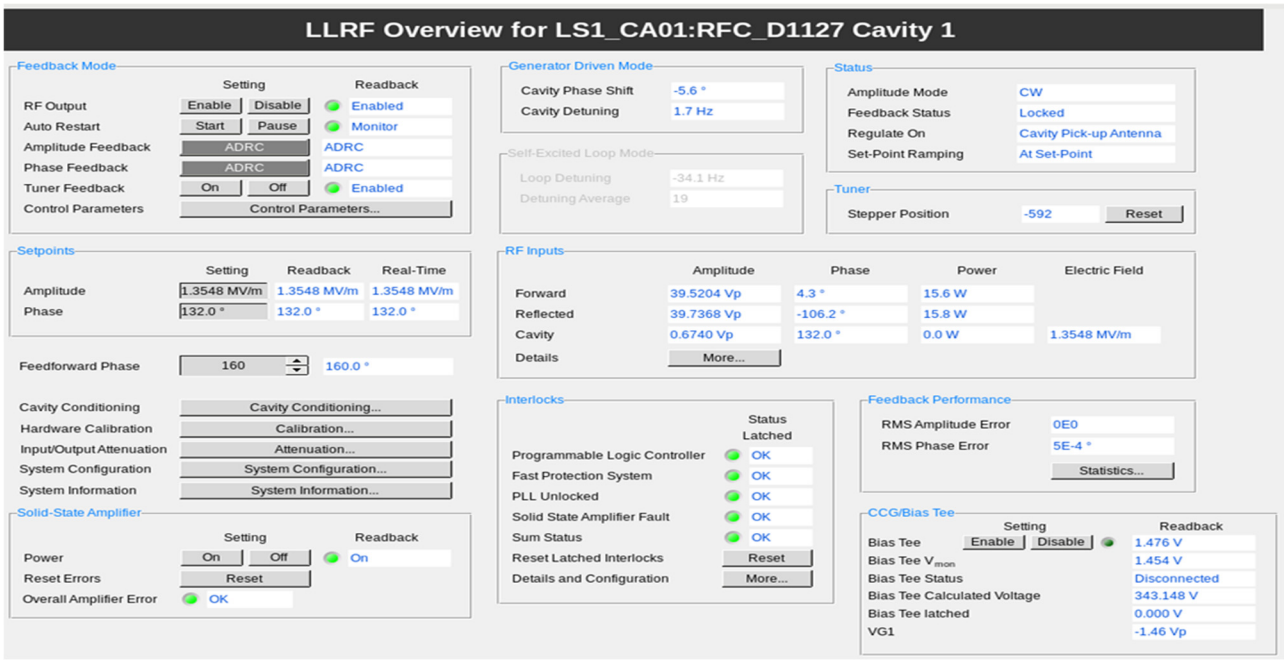

Figure 1: CS-Studio LLRF main OPI screen for a single cavity.

reading from and writing to EPICS PVs via the CA protocol. The package includes a thin and fairly complete layer over the low-level CA library in the ca module, and higher level abstractions built on top of this basic functionality [4]. QtDesigner, included in the PyQt package, is used for rapid prototyping of a graphical user interface (GUI) and its generated files are converted to Python using the pyuic5 module. Figure 2 shows FRIB LLRF Expert Tool GUI. Connecting to EPICS PVs is more expensive than retrieving data from connected PVs. Connection expense can be lowered by not retaining the connection or creating monitors on the PVs [4]. Connection to a larger number of PVs is made through caget_many() and caput_many() for most of the tasks in this tool. PV objects are used where connection to a PV is required to be kept alive to re-use the PV. This approach utilizes less network resources, that results in less congestion on the network while significantly improving speed.

PV names at FRIB contain the linac system, system type, sub-system type, cavity distance number and the parameter name. The PV naming convention is "LinacSystem_SystemType:SubSystemType_CavityDistance:Parameter". For example, "LS1_CA01:RFC_D1127:EN_CMD_RF" is a PV for enabling RF on the D1127 cavity of the CA01 system type cryomodule that is in LS1 and this parameter is a part of sub-system type RF Controller (RFC). Writing a value 1 to this PV enables RF. Due to this convention, it is easy to target the same PV for selected LLRF controllers, or go through all PVs of one controller.

Device names and parameter names are stored in separate lists to enable flexibility while performing PV actions. This modular approach improves scalability and makes modifications of device and parameter names easy while also saving storage space. Additionally, parameters are divided into set parameter names and readback parameter names. PV names are generated by appending parameter names to device names before execution of a task. The "Output" window displays messages, errors and warnings for the current running task.

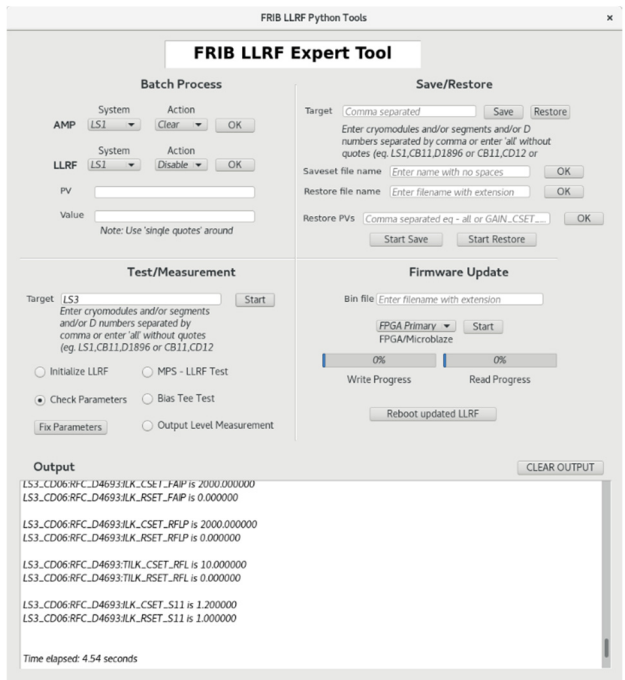

Figure 2: FRIB LLRF Expert Tool.

OPERATION TASKS

Batch Process

Manually enabling RF of multiple cavities of multiple cryomodules in CS-Studio is time consuming. The batch process task within FRIB LLRF expert tool allows the

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-WEPAB300

operator to interact with multiple PVs simultaneously. The operator can select the system type and the action. For example, LLRF actions are enable/disable RF, reset interlocks and write a specific value to a similar group of PVs. The operator can select the system type and enter the PV name, for example, "EN_CMD_RF", and specify a value, in this case, either 1 or 0. This will execute a write to all the operator specified PVs within the selected system type. Batch processing helps the operator save time on manual tasks and gives more control interacting with PVs.

Test/Measurement

Operators perform certain tests before an operational run to verify the status of the subsystems and troubleshoot the problems if there are any. Operators can select specific cavities for these tests by entering the target linac segment name(s), cryomodule name(s), cavity name(s) or a combination of these names, for example, operators can enter "LS3" to select all cavities within LS3 or "CA01" for all cavities within cryomodule CA01 or "DXXXX" to select a specific cavity where X is the cavity number or a combination "LS3, CA01, D5678" or "all" if all PVs are to be selected.

"Initialize LLRF" is used to setup parameters such as cavity type, attenuation, interlocks, control parameters, et al. to initial values before turning on RF for the first time.

Sometimes after an IOC reboot, setpoint values don't match with readback values. "Check Parameters" verifies if setpoints match with their corresponding readbacks and "Fix Parameters" fixes any discrepancies in readbacks by re-writing the setpoints with current values.

Machine protection system (MPS) is one of the many subsystems connected to the LLRF controller and "MPS – LLRF connection test" verifies if the connection is healthy between the two subsystems. This test is performed by tripping certain interlocks on the MPS and verifying the trip status on the LLRF.

"Bias Tee Test" checks the status of bias tees by verifying the voltage readback during on and off state. "Output Level Measurement" displays output attenuation for selected cavities.

MANAGEMENT TASKS

Save/Restore

Data archiving enables operators in analysing historical data and retrieving it at a later point in time helps in recreating the operating environment. The user can enter a target to select desired set of cavities and enter a name for the saveset file. The saveset file contains only values of the PVs and is stored as a compressed text file consequently saving significant storage space. The day, month, year and time is suffixed to the file name for easy classification. The restore operation gives the operator enhanced control over the selection of specific cavities, system types, cryomodules and PVs. If the operator desires to restore only a particular PV or PVs, the operator can enter the individual PV names and these PV values will be restored for the selected target cavities.

Firmware Update

During maintenance, LLRF controllers are updated with the latest firmware and software for FPGA or Microblaze. The LLRF engineer enters the file name and selects the desired system. Upon successful completion of the update process, the controllers are rebooted for changes to take effect.

PERFORMANCE IMPROVEMENT

This tool has significantly improved performance in terms of time taken to perform specific tasks and in turn has caused enhanced operator productivity. Figure 3 shows a comparison between time taken for tasks when performed manually and when performed using Python. Performing tasks in Python is, on average, 20 times faster than performing the same tasks manually in CS-Studio.

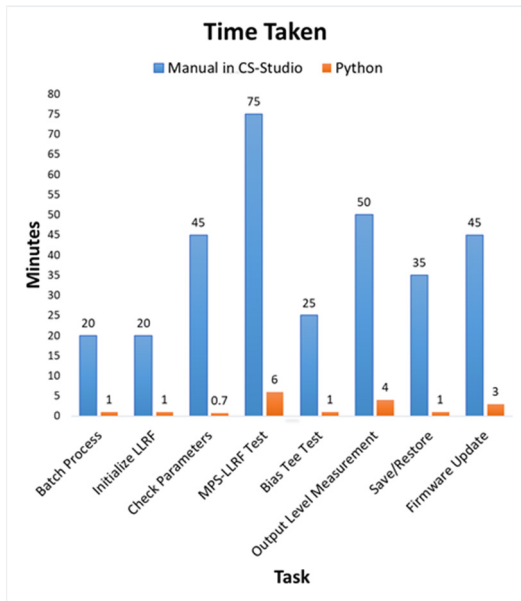

Figure 3: Performance comparison.

CONSIDERATIONS

Python provides fast iterations and great flexibility for developing useful tools. Due to this reason it is prone to mistakes. To avoid any potential adverse effects for running the Python scripts an "Acceptable Script Use on FRIB Controls Network" plan was developed at FRIB that prohibits using scripts for safety functions or when beam power is more than 2 W. It also defines the roles and responsibilities of all parties e.g. developer, reviewer, system owner and user.

Python cannot bypass channel access security rules. A script that involves writing to certain restricted process variables can only be executed by the users who have the privilege. An example of this is the auto-start feature for turning on cavities, a Python script was first developed and tested and then translated to the state notation language (SNL) to be run on the IOC, so regular users can use the feature without violating channel access security rules.

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-WEPAB300

REFERENCES

[1] J. Wei *et al.*, "The FRIB Superconducting Linac - Status and Plans", in *Proc. 28th Linear Accelerator Conf. (LINAC'16)*, East Lansing, MI, USA, Sep. 2016, pp. 1-6. doi:10.18429/JACoW-LINAC2016-MO1A01

[2] D. G. Morris *et al.*, "RF System for FRIB Accelerator", in *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*, Vancouver, Canada, Apr.-May 2018, pp. 1765-1770. doi:10.18429/JACoW-IPAC2018-WEXGBF3

[3] PyQt v5.13 reference guide, http://www.riverbankcomputing.com

[4] PyEpics: Epics Channel Access for Python, https://cars9.uchicago.edu/software/python/pyepics3/.