The FRIB Linac consists of:

- Linear Accelerator (Linac)
- Tuners
- Resonators
- Beam Delivery System
- Quads (Seg1-3)

All these PVs are accessible using the Experimental Physics and Industrial Control System (EPICS) channels. IOC driver handles read/write actions to these PVs.

Python Application:

- Python offers several convenient features for scientific and engineering programming. The Python EPICS package (PyEpics) is useful to interact with EPICS channel access PVs.
- Qt Designer is the Qt tool helpful in designing and building graphical user interfaces. Utilities from PyQt library helps to generate Python files.
- Built-in widgets and forms
- XML to format to store design files
- 'translate' module to convert to C++ code
- 'translate' module to convert to Python code
- Several Python scripts have been developed for RF commissioning at FRIB. "FRIB RF Expert" application lets users perform mass action to apply to multiple LLRF and amplifier systems. This application serves as main file and all developed Python scripts are nested under it.

System:

- Selection based on cavity number, cryomodule number, system type or Linac segment type. Script searches for keyword in database where all device names for LLRF controller and Amplifier are located.
- Action: Options to turn On/Off, clear reset interlocks for LLRF controller and amplifier.
- PV: Manually input PV name and Value.
- Initialize LLRF: Useful to setup parameters such as cavity type, attenuation, interlocks, control parameters etc. to initial values before running RF for first time.
- Check Readbacks: useful to make sure all set-points and readbacks match and there is no discrepancy in values.
- MVS-LLRF test: Useful to run test between LLRF controller and Machine Protection System (MPS) to ensure any combination of system and action is selected, it shows a pop-up message to verify user actions.
- Output window shows device names, PV names and value has changed.
- Python script is an easy way to prototype any state machine quickly and test the logic.
- Once developed and tested, script's logic can be transferred to State Notation Language (SNL) and implemented on IOC driver. To provide most channel access security, it is recommended to implement state machines on IOC driver.

State Notation Language (SNL):

- Allows programming of sequential state-oriented operations to run in the IOC.
- Can implement complicated algorithm, Implement complex closed loop control schemes
- C code can be embedded as part of the sequence
- Channel Access security
- Watch for likely fault modes that are hard to detect via alarms
- Requires IOC restart for changes
- Coordinate control of multiple devices
- File access can be implemented as part of the sequence

Future Work:

- Transitioning this application to State Notation Language (SNL) and implement on IOC driver to add channel access security.