

# DEVELOPMENT OF EPICS ACCELERATOR CONTROL SYSTEM FOR THE IAC 44 MeV LINAC

Anthony Andrews<sup>1,2\*</sup>, Y. Kim<sup>1,2,3</sup>, C. Eckman<sup>1,2</sup>, P. Buaphad<sup>1,2</sup>

C. O'Neill<sup>2</sup>, B. Berls<sup>2</sup>, K. Folkman<sup>2</sup>, J. Ralph<sup>2</sup>, and M. Khandaker<sup>1</sup>

<sup>1</sup>Department of Physics, Idaho State University, Pocatello, ID 83209, USA

<sup>2</sup>Idaho Accelerator Center, Idaho State University, Pocatello, ID 83201, USA

<sup>3</sup>Thomas Jefferson National Accelerator Facility, Newport News, VA 23606, USA

## Abstract

The Idaho Accelerator Center (IAC) of Idaho State University (ISU) has been operating nine low energy accelerators [1]. Since fall 2012, the ISU Advanced Accelerator and Ultrafast Beam Lab (AAUL) group has been working to develop a new EPICS system to control 51 magnet power supplies and 2 GigE CCD cameras for the IAC 44 MeV L-band linac. Its original control system was fully analog, which had several limitations in reproducibility and stability during the accelerator operation. This paper provides details of the groups team effort and accomplishments in developing a new EPICS system to control Prosilica GigE CCD cameras and several magnet power supplies including Lambda EMS, Lambda TCR, and TDK-Lambda ZUP power supplies [2–5]. In addition, other useful tools such as the save and restore function used for this linac operation are described.

## INTRODUCTION

To improve the reproducibility and stability of the 44 MeV linac, the control system has been upgraded with EPICS. Unlike the previous analog control system, EPICS provides a means to retrieve high quality pictures from GigE CCD cameras, current readout of magnet power supplies from the control room, stable current control that is not affected by humidity or temperature variations, and a save/restore function to restore operational conditions of the linac. As shown in Fig. 1, the control system for the 44 MeV has three different types of magnet power supplies: TDK-Lambda ZUP, Lambda EMS, and Lambda TCR power supplies. There are 32 TDK-Lambda ZUP power supplies, 15 Lambda EMS power supplies with embedded IEEE 488 controllers, 3 Lambda EMS power supplies with external RSTL controllers, and 1 Lambda TCR power supply with the external RSTL controller<sup>1</sup>. All Lambda EMS and Lambda TCR magnet power supplies were connected to a shared MOXA terminal server with RS485-RS232 conversion interfaces, and the TDK-Lambda ZUP magnet power supplies were connected to the second MOXA terminal server with RS485 interfaces [6]. Then, the terminal servers and Prosilica GC1290 GigE CCD cameras were connected to the EPICS server

\* Mail: andranth@isu.edu

<sup>1</sup>The EPICS system for these power supplies is detailed in references [7–9]

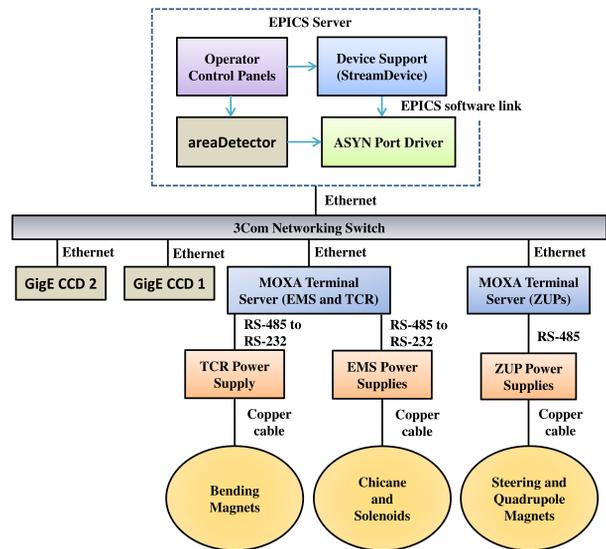


Figure 1: Experimental setup of EPICS server and hardware for the 44 MeV linac.

via an isolated network using Ethernet cables and a 3Com network switching hub. After that, a means to communicate with the power supplies and cameras was developed by using three EPICS modules (ASYN, StreamDevice, and areaDetector) supplied with the synApps package [8]. Then, device support applications were manually programmed for the power supplies and the module areaDetector was used to control one of the CCD cameras. Finally, the current of the power supplies and various parameters of the CCD camera were controlled by manipulating process variables (PVs) using MEDM Operator Interface (OPI) panels.

## DEVICE SUPPORT APPLICATIONS

To control the magnet power supplies and CCD cameras, the first step was to download and install the correct software to communicate with them. Then, in the case of the power supplies, the second step was making device support applications which contain the files required to create PVs. The device communication was accomplished by installing the synApps software package including areaDetector, ASYN, and StreamDevice.<sup>2</sup>

<sup>2</sup>For more information on ASYN and StreamDevice, see reference [9]

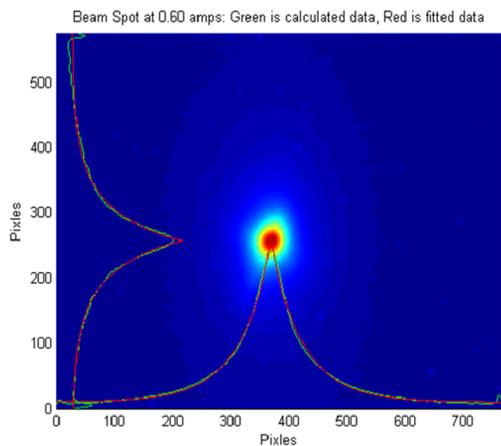


Figure 2: MATLAB processed beam image.

### Control System for CCD Cameras

To develop the control system for the Prosilica GC1290 GigE CCD camera, the module areaDetector was used. This module provides pre-made device support with PVs and OPI panels for various detectors and cameras. First of all, to control the camera with EPICS, one should determine the right camera settings to obtain an image using the program SampleViewer. After that, the IP address of the camera should be determined and placed within the *st.cmd* file [8]. This CCD camera control system along with the EPICS extension MATLAB Channel Access (MCA), was used to develop an automatic emittance measurement tool for the 44 MeV linac [10]. A beam picture produced by this tool is shown in Fig. 2. The other GigE CCD camera was used to monitor the beam position and shape on a YAG screen in real time with the program SampleViewer [8, 10]. The creation of PVs for the power supplies had to be done manually.

### Application Structure for Power Supplies

Before creating PVs for the magnet power supplies, first of all, an application was created for the PVs to live in. The structure of the application looks like Fig. 3. Since the 44 MeV has different types of power supplies, two separate applications were made to separate them by the MOXA terminal servers, which they are connected to. One application was made for the TDK-Lambda ZUP magnet power supplies, and the second for the Lambda EMS and TCR magnet power supplies. The EMS\_Sup, TCR\_RSTL\_Sup, and TDK\_Sup directories contain the files to create PVs. The directory under iocBoot contains the *st.cmd* file<sup>3</sup>. In order to create PVs, two types of files are required, protocol files and database files.

### Protocol Files

The module StreamDevice uses protocol files to translate commands that follow the SCPI (Standard Commands

<sup>3</sup>For more information on the commands to create an application, see reference [7]

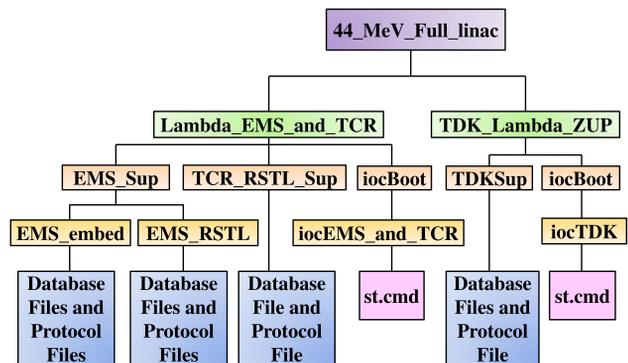


Figure 3: Magnet power supply application hierarchy.

for Programmable Instruments) format into protocols for use in records [8]. For example, the SCPI query command MEASURE:CURRENT?, from the Lambda EMS with embedded control, looks like this [8]:

- getMCURR{out “MEASURE:CURRENT?”; in “%6f”; in “OK”;}

In this case, getMCURR is a protocol representing the SCPI query command from the users manual. This protocol name is then used later in a record.

### Database Files

A database file is where the records are located. The PVs are made by putting the protocol name (i.e. getMCURR) into the input or output fields in the record [8]. Here is an example of a record to read the current:

```
record( ai, “$(P):HE_SOL_18_19:curr_r”) {
  field( DESC, “Measure Current”)
  field( DTYP, “stream”)
  field( INP, “@EMS.proto getMCURR $(PORT9) $(A)”)
  field( PREC, “3”)
  field( EGU, “Amps”)
  field( HOPR, “18”)
  field( LOPR, “0”)}
```

In the record example, a field of particular interest is the INP field. This field sends the command defined as the protocol “getMCURR” located in the protocol file “EMS.proto” to the port and address of the device. The \$(PORT9), \$(A), and \$(P) are all defined in the *st.cmd* file. When a record performs it’s assigned function, in this case reading the current, it is known as “processing”. In the example record above, the only way processing happens is

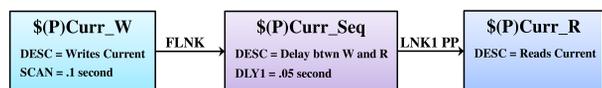


Figure 4: Representation of the common database used for the control system of the 44 MeV linac.

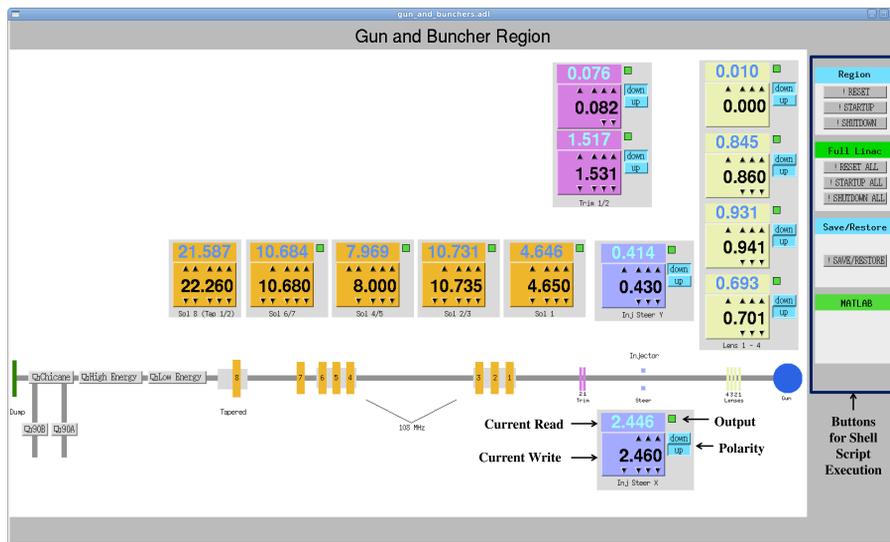


Figure 5: One of the 6 MEDM OPIs to control and read the current of the magnet power supplies.

when some outside record forces it to process. This processing nuance is shown in Fig. 4. In this example, the record to write the current processes every 100 ms. Then this record forces another outside “sequence” record to process at the same time. The sequence record’s purpose is to wait for 50 ms and then process the next record in the chain. In this case, the next record reads the current from the power supply, and the current is read from a power supply approximately every 150 ms. This process was followed because most power supplies for the 44 MeV linac, except for the Lambda EMS power supplies with embedded controllers, require some delay between write and read operations. The power supplies that require a delay, have a physical mechanism that requires at least 50 ms to trigger the readback. The TDK-Lambda ZUPs took 50 ms for this trigger to occur, but the external RSTL controlled power supplies took about 200 ms [5]. To complete the application setup for the magnet power supplies, it’s necessary to put some information in the *st.cmd* file.

### ST.CMD FILE

This is an executable file to establish a connection with the device by using an IP address, a portName, and an address. This file also loads all of the records that are in the database files. The IP addresses of the power supplies were specified in this file by using lines similar to the following:

```
drvAsynIPPortConfigure(“P9”,“134.50.x.y:4009”,0,0,0)
```

Here P9 is a portName, which is an arbitrary identifier used to define \$(PORT9) from records as an IP address (134.50.x.y) and a port (4009) of the MOXA terminal server [8]. In this case, P9 was used because this particular power supply was connected to the ninth port of the MOXA terminal server. The rest (0,0,0) specifies *priority*,

*noAutoConnect*, and *noProcessEos* [8]. To specify a database in the *st.cmd* file, a line similar to the following was used:

```
dbLoadRecords(“db/S.db”,“P=$(P),PORT9=P9,A=9”)
```

Here the S.db is the database file that contains all of the needed records. The macro P was used as a device identifier for the particular linac, in this case P is 44MeV. The PORT9 is the same as the portName. The address (A=9) for this set-up ran from 1 - 32, because there were 32 ports on each MOXA terminal server [8].

### MEDM OPI PANELS

Motif Editor and Display Manager (MEDM) is an EPICS extension which provides a way to create simple OPI’s [9]. One of the OPIs used for the 44 MeV control system is shown in Fig. 5. The 44 MeV linac is split into six regions: the “Gun and Bunchers”, “Low Energy”, “High Energy”, “Chicane”, “90A”, and “90B” regions. The “Gun and Bunchers” as shown in Fig. 5 is the injector portion of the linac. After that, the low and high energy regions are the accelerating structures of the linac. The “Chicane” region contains a chicane used during the IAC-RadiaBeam THz project in November 2012 [8]. The 90A and 90B degree lines are used to divert the beam for laser Compton scattering experiments [1, 11]. All displays have a schematic of the linac showing the magnets being controlled and related display buttons to navigate to other region displays. The various control and readback widgets have the same color and label as the components being controlled. There is also additional information such as the status of the output of relevant power supplies (on/off) and the polarity of relevant power supplies (up/down) [7]. Other common features that all displays have are the shell command widgets on the right hand side of each display. When an operator

clicks these shell command buttons, various shell scripts created for the 44 MeV are executed.

## SHELL SCRIPTS

There are four different categories of shell scripts for the 44 MeV linac, each of which are shown on the right hand side in Fig. 5. The first category, represented by the box entitled “Region”, execute shell scripts that act on the particular region of the linac. The buttons included in this category are: “Reset”, “Startup”, and “Shutdown”. When the operator clicks Startup, a script turns on the output of each magnet power supply in the region. When an operator clicks the Reset button, a shell script is run which sets all region power supply current values to 0.000 A. The Shutdown script sets the region current values to zero, and then shuts the output off. The second category of shell scripts act on the whole linac. The “Startup All” script turns on all of the output for all power supplies. The “Reset All” script sets all power supply current values to zero in the entire linac. The last “Shutdown All” script saves all of the power supply current values and polarity orientations to a file with the date and time, sets the current values to 0.000 A, and shuts the output off for every power supply. The third category runs the save/restore program, and the last category is reserved for MATLAB programs which haven’t been added to the displays yet.

## BURT SAVE/RESTORE

The save/restore program chosen for the 44 MeV linac is the EPICS extension Back Up and Restore Tool (BURT) [12]. This program is used when the “Shutdown All” script is executed. A BURT command and shell scripting allow one to create an automatic save function when “Shutdown All” is executed. In order to save PV values, the programmer must first create a “request” file. A request file is a simple C program file that contains a list of relevant PVs to save. The 44 MeV linac has several request files listing PVs for specific regions or one for the entire linac. When “Shutdown All” is executed, BURT reads the full linac request file and automatically saves all values of the PVs to a “snapshot” file. The snapshot file is used to restore the previous PV values when the BURT restore operation is executed. The operators also have the option to manually save/restore current values by clicking the “Save/Restore” button on a display as shown in Fig. 5. When this button is pressed, the BURT GUI shown in Fig. 6 is displayed. From this GUI, operators have the option to “Backup” (save) the state of the control system, “Restore” any previous saved states from snapshot files, algebraically manipulate PVs using “Add/Sub” and “Mult” options, combine multiple snapshot files into one large one using “Set”, and create/alter request files using “Def/Inc”.

## CONCLUSION

The goal of the 44 MeV linac control system upgrade was to create an EPICS control system for a set of TDK-



Figure 6: GUI supplied with the program BURT.

Lambda ZUP magnet power supplies, Lambda EMS magnet power supplies of mixed control type, a TCR magnet power supply, and Prosilica GigE CCD cameras. This was accomplished by using ASYN and StreamDevice to communicate with the power supplies and convert SCPI commands into PVs. Then several MEDM OPI displays were created to control and monitor PVs for the magnet power supplies in each region of the 44 MeV linac. To control the camera with EPICS, areaDetector was installed which came with the needed PVs and MEDM OPI displays. This upgrade also included a save/restore function with BURT. Using this upgraded EPICS accelerator control system, the operators have seen improvements in the efficiency of the linac including: stability in magnet power supply current values that don’t fluctuate with temperature or humidity, and the ability to save/restore the values of the magnet power supplies reduces the time to optimize the 44 MeV linac.

## ACKNOWLEDGEMENTS

Anthony Andrews would like to thank Dr. Kukhee Kim, Dr. Ernest Williams, and Dr. Debbie Rogind for their valuable training of various EPICS related things during his summer internships at SLAC from 2011 to 2013.

## REFERENCES

- [1] <http://www.iac.isu.edu>
- [2] <http://www.alliedvisiontec.com/us/products/cameras/gigabit-ethernet/prosilica-gc/gc1290.html>
- [3] [http://www.us.tdk-lambda.com/hp/product\\_html/emspower1u.htm](http://www.us.tdk-lambda.com/hp/product_html/emspower1u.htm)
- [4] [http://www.us.tdk-lambda.com/hp/product\\_html/tdcrpower.htm](http://www.us.tdk-lambda.com/hp/product_html/tdcrpower.htm)
- [5] [http://www.tdk-lambda.com/products/sps/ps\\_adj/zup/indexe.html#](http://www.tdk-lambda.com/products/sps/ps_adj/zup/indexe.html#)
- [6] [http://www.moxa.com/product/nport\\_6650.htm](http://www.moxa.com/product/nport_6650.htm)
- [7] A. Andrews *et al.*, in *Proc. NA PAC 2013*, Pasadena CA, USA.
- [8] A. Andrews *et al.*, in *Proc. IPAC2013*, Shanghai, China.
- [9] A. Andrews *et al.*, in *Proc. IPAC2012*, New Orleans LA, USA.
- [10] C. Eckman *et al.*, in *Proc. IPAC2013*, Shanghai, China.
- [11] M. Titberidze *et al.*, in *Proc. IPAC2012*, New Orleans LA, USA.
- [12] N. Karonis, “BURT: Back Up and Restore Tool”. Argonne National Laboratory, <http://www.aps.anl.gov/epics/EpicsDocumentation/ExtensionsManuals/Burt/BurtManual.book.html>