Implementation of a new Triton control system for ROC

Daniel Kabuß

Abstract
The ROC experiment at the COLLAPS beamline in the ISOLDE facility utilized a the Triton control system in an earlier setup. As it is planned to use this setup again, the control system is required again. As Triton has not been used in the meantime, the installation has not been maintained. This resulted in multiple problems that make it nearly impossible to use without prior experience with the Triton control system. To avoid the problem in the future, an installation routine based on files that are available at CERN and a new control interface have been created in this work.

1 Introduction

My summer student project was performed in the context of the COLLAPS beamline, which is used for collinear laser spectroscopy on ions delivered by the ISOLDE facility. The excited ions can be detected by the **R**adioactive detection of **O**ptically pumped ions after state selective **C**harge exchange (ROC) setup.

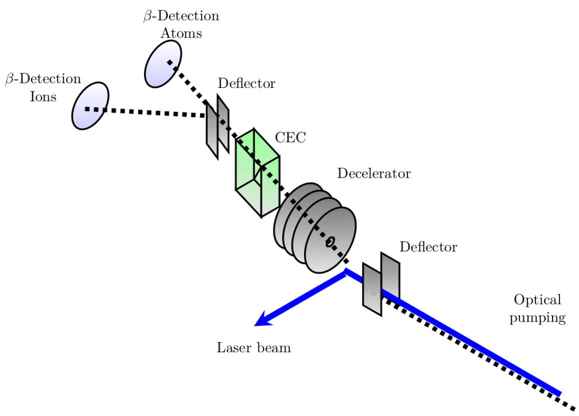

Fig. 1: The main components of the ROC setup.

In the ROC setup (scheme in figure 1), a laser pumps the ions into a state that has the possibility to decay into a metastable state. This metastable state has a different probability of being neutralized in the charge exchange cell compared to the ground state. This means the relation between the number of ions and atoms is changed if the ions are pumped successfully.

The difference in the neutralisation probability only occurs at low beam energies. Therefore, the ions have to be decelerated before entering the charge exchange cell. This means that the charge exchange cell will be on a high voltage and the control can not utilize a direct electric connection from the used computer.

The voltages in the experiment are controlled by the Triton control system, which has last been used a few years ago and has not been updated. Now it is planned to use the system again, so an update of the slow control system is mandatory.

2 Triton

Triton (**TRI**ga py**T**hon c**ON**trol system) is an experiment control system created for the TRIGA experiment at the Johannes Gutenberg-Universität Mainz, Germany. At the moment, the main development is done by the LaserSpHERe group at the TU Darmstadt, Germany.

Triton is currently used to control collinear laser spectroscopy experiments at the TU Darmstadt, Argonne National Laboratory and the COLLAPS beamline at the ISOLDE facility. It is designed to allow control of and communication between multiple devices used in an experimental setup that are possibly connected to different computers. This is achieved by the use of the Pyro4 package for Triton.

At the same time, the use of this package is a major complication, as its usage has changed with its versions and Triton has not been updated accordingly, requiring the outdated version 4.20 (the current version is 4.76). This version is also incompatible with Python versions greater than 3.6. To ensure compatibility, Miniconda is used for version control. This combination makes the installation somewhat complex and strictly limits possible updates. Miniconda has more specific requirements for Pyro4 version 4.20 version that lock Triton to running on Python 3.4 that can not be circumvented easily.

Miniconda also has a more limited package repository compared to the PyPi.org default Python repository that does not include all packages used by Triton components. This makes it necessary to install additional packages by different means, creating additional complications in the installation process.

The information about the devices is stored in a database. This database is shared across all instances of Triton in a setup and allow communication between the different instances. The database also defines which of the components that are present in the Triton code are actually available and their configuration. This allows avoiding cluttering of the interface by components that are not supported or useless at the setup.

An important feature of Triton for this work is the use of signals to communicate between components. When a component sends a signal, it is broadcasted to all components that have subscribed the sending component. This allows arbitrary components to receive the signal, but limits the possibility to send information to specific components.

3 Previous installation at COLLAPS

There has been an old installation of Triton from the last time it was used. This installation was not usable any more due to a small error in the configuration, making it unable to find the device to control.

The board number assigned to the Measurement Computing USB-3105 device by the InstaCal program has to be entered into the Triton database. If InstaCal is launched without the connected device, the new setup can change this number. Effectively recognizing and fixing this error is only possible with knowledge of the driver used, as the configuration was not documented sufficiently.

A further problem of the installation is the fact that no installation files are available for COLLAPS. The usual source for the files is the ownCloud of the institute of nuclear physics at TU Darmstadt and the internal Git server of the institute. The instructions can be found in the Lablog of the LaserSpHERe group.

This means that neither of those is easily available for users at COLLAPS and is is not possible to install Triton on a new computer or update the existing installation.

4 New installation routine

The easiest way to allow fixing the potential configuration problem for future uses and to make the reinstallation of Triton possible was the creation of an installation routine with files that are stored in a way that is accessible for anyone at COLLAPS.

To avoid excessive divergences from the the main development at Darmstadt, the supporting packages have been copied from there. As there are unique components and configurations for COLLAPS, the Python files and configuration database for Triton had to be taken from the old installation. Together with the programs required for the drivers and the database setup, they have been stored in the COLLAPS2 workspace on DFS. To be usable by everyone, instructions for the installation procedure have been created and added to the ROC section of the COLLAPS Labbook, found under COLLAPS archive/ROC/Triton.

The instructions have been tested by installing Triton on a new PC by a user that has not installed Triton previously to ensure they can be understood without further knowledge about Triton. The instructions don't include the necessary drivers for all possible components of Triton, because most of them are not used at COLLAPS and they have not been documented in the previous installation, making it hard to determine which drivers and other supplementary components are necessary.

5 New control interface

The old control interface for the voltage sources for the ROC experiment was implemented in a way that was specific to the application and allowed configuration only by directly editing the configuration database. Also, the way this configuration works was not documented. This makes the interface very specific to the setup used at the time. This is mostly unnecessary, as the operations performed are rather basic.

To improve this situation, a new control interface was created. Instead of adjusting it specifically to the task, it was designed to be easily configurable by the user. To make it compatible with more than one possible device, it uses Triton system of signals. These are broadcast by components to make their outputs available. They can also be used to trigger actions if the component is set up the right way. While this kind of interaction is not spread widely at the moment, it is easy to add to existing devices.

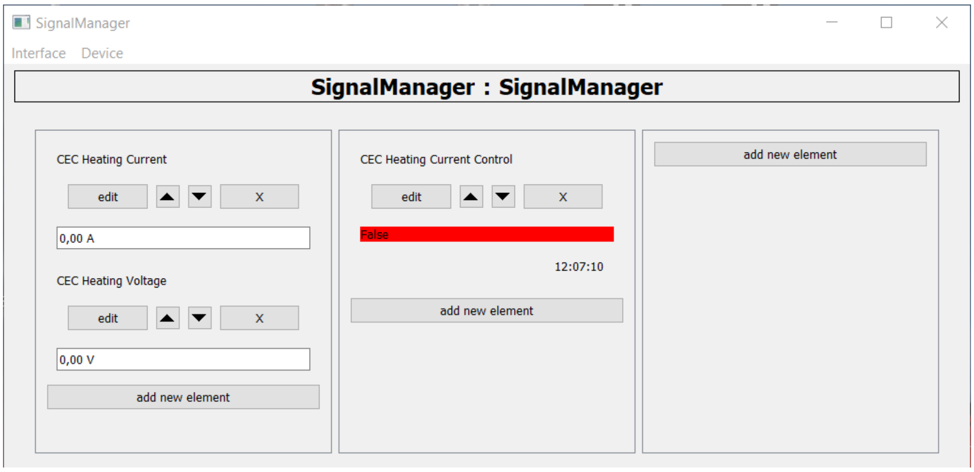

Fig. 2: Interface of the new Triton interface for the ROC experiment

The new interface consists of a number of columns. The amount of columns can be changed by editing the numCols value in the config column of the SIgnalManager entry in the devices table of the Triton database. This setting should only be adjusted when setting up a new computer, as other settings can only be loaded when the number of columns is equal to the number of columns at the time of their creation. After Triton component has been started, the interface will be empty. Elements can be added to the columns using the "add new element" button in the respective column. This will open the configuration menu for the new element.

5.1 Configuring a new element

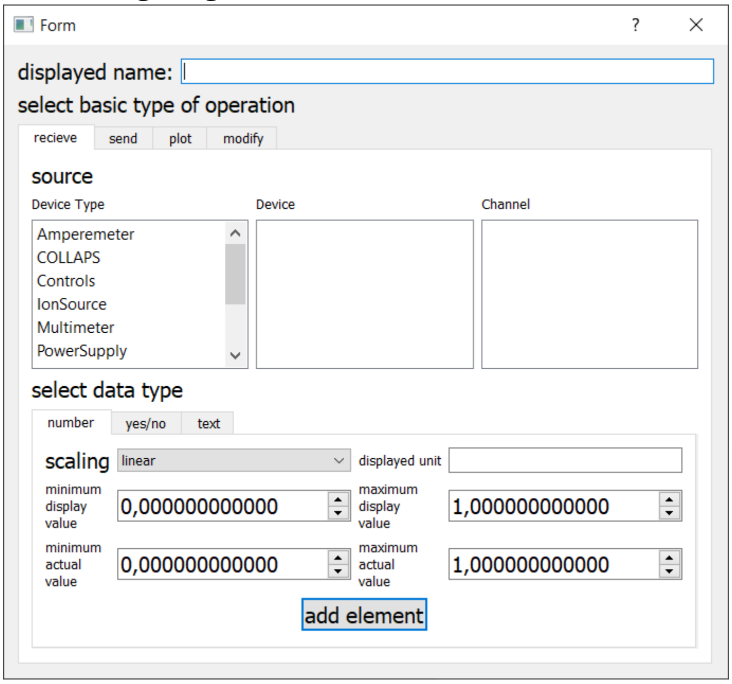

Fig. 3: Empty configuration interface for elements of the new interface

The interface for the configuration of an element can be seen in figure 3. It is required to give the element a name. This serves both the purpose of allowing the user to see what a certain element does and to identify the elements internally.
Every element needs to receive or send data. The source or target for the signal can be specified by selecting the appropriate device group, which brings up the list of devices in the group, and selecting the device in this list. If the element is to receive data, this will show a list of channels. These are the channels a device will send data in. Select the appropriate channel to complete the selection.

For a element that sends data, the channel selection is more complicated (compare figure 4). As the Triton database only stores information about the channels a device is expected to send on, the possible options can not be filled in automatically. It has to be found in the documentation of the component. For the component used to control the voltages in the ROC experiment, see section 5.2.
Depending on the data type used, further options might be available. For numerical data, it is possible to append a unit to the displayed value. The other option is to define the scaling and allowed range of values. This is done by entering the range of the values shown by the interface and the range of the actual values send as signals. The scaling entry defines how the ranges are mapped onto each other.

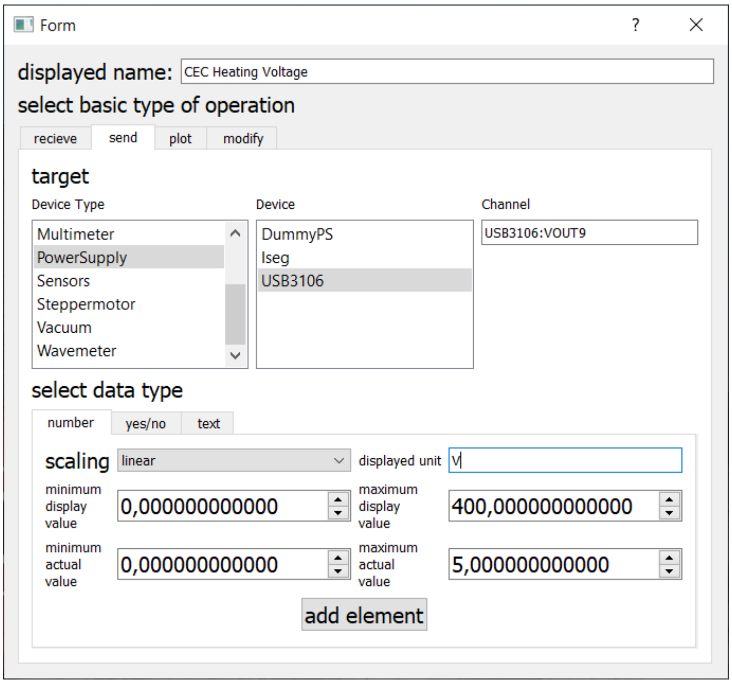

Fig. 4: Configuration used for the heating voltage element in figure 2

When sending signals, the range should be set to the rang of values the user is allowed to set. When receiving, the range should be set to the range of values during normal operation. The interface will allow values outside the range, but will colour the element red to indicate a unexpected value. For receiving boolean data, there is a option to invert the value. If the normal value is false, it is recommended to choose this option, as the element will be coloured green if the value is true and red if it is false. For text signals, no options are available.

5.2 USB-3105 component

The main goal for the use of Triton in the ROC experiment can be achieved by using the USB-3105 DAC device produced by Measurement Computing. The old use of this device included unnecessary complications and was not flexible enough to allow adaption to changes in the setup, for example a change in the number of controlled voltages.

To adapt the component (which is for some reason called USB3106) to the new interface, it was programmed to react to signals. The names of the signals are started with the name of the component and a colon, to make clear that the signals are intended for the specific component. This is followed by the name of the output channel as printed on the device, which is VOUT followed by a number from 0 to 15. The resulting signal name is ”USB3106:VOUT0” to ”USB3106:VOUT15”.

While there are 16 analog output channels on a USB3105, only nine of them work on the one at ROC. This limits the use of the device until a replacement has been bought. The woring channels are the following: 4, 5, 6, 7, 8, 11, 12, 13, 14. All eight digital input channels have been confirmed to be useable.

6 Conclusion

The new installation routine and interface for the use of Triton at the ROC experiment been created. The installation routine has been tested to be understandable without prior knowledge of Triton. The interface is available and can be configured by the configuration interface. It is possible to save and load the interfaces created this way to avoid having to recreate the interface each time.

The complications in the Triton installation have been discussed with the LaserSpHERe group at TU Darmstadt and it has been decided to look into possible alternatives to make both installation and updating of existing installations easier.