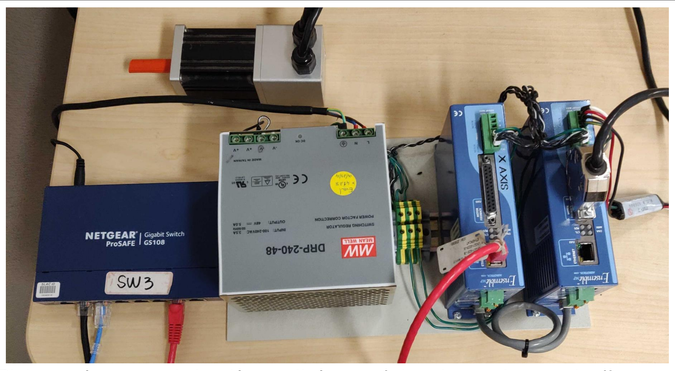
Python and PyDM Interface to Communicate with Aerotech Ensemble Motor Controllers

Anthony Le

Linac Coherent Light Source, SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA.

Contact: le123456@slac.stanford.edu

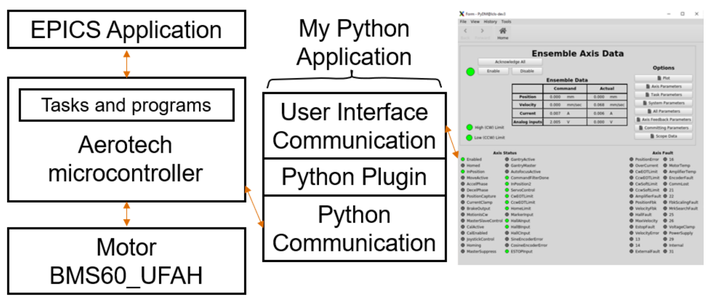
Introduction



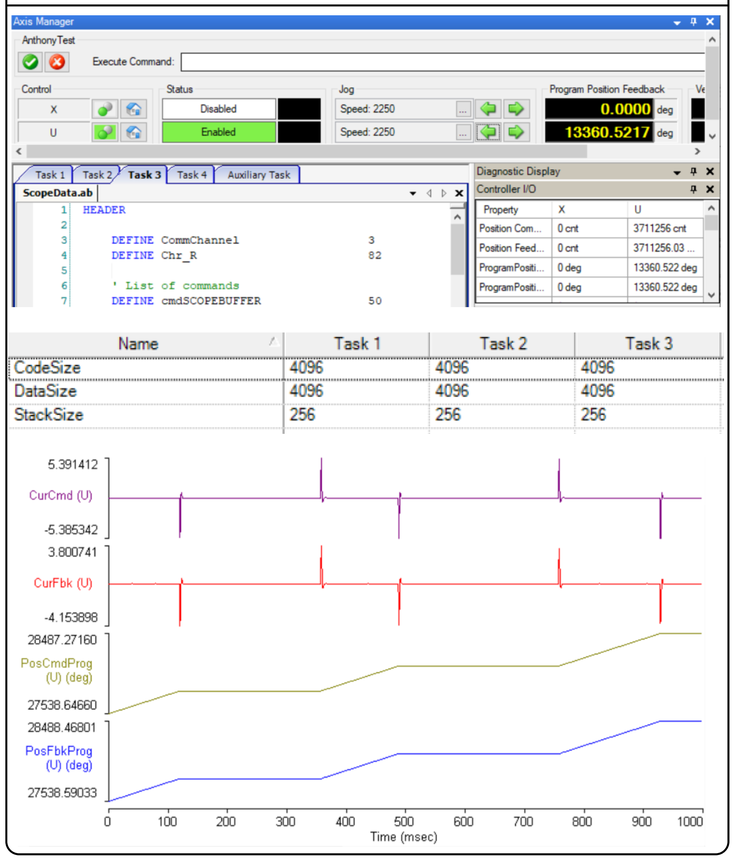
Picture above contains the switch regulator, two microcontrollers for two different axes, the motor, and router.

Electrical Engineering Department (EED) uses different controllers to remotely monitor and control moving devices along the accelerator. One of the motion controllers used extensively in Linear Accelerator Coherent Light Source-II (LCLS-II) is Aerotech Ensemble Drive. These Aerotech drives control servo and stepper motors through an EPICS IOC (Input output Controller).
The objective of this project was to review and update a Python interface to communicate with Aerotech Ensemble controller in Linux without an EPICS IOC. Data collected from the motors connected to this controller is saved into several files. These files along with controller parameters are exposed to the Linux server using this Python interface.

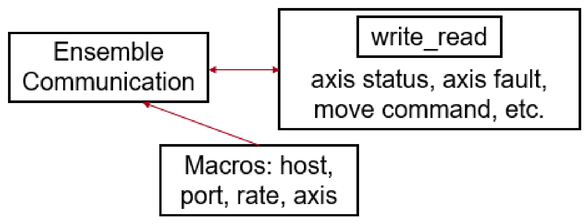
Overview



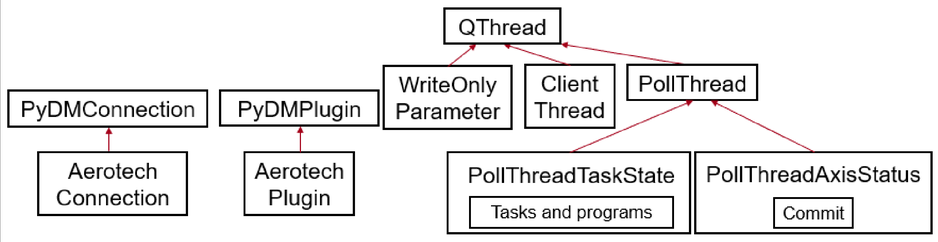
Windows Ensemble Interface



Python Communication



Aerotech PyDM Plugin



Inheritance diagram of PyDM Plugin

Channel for a display parameter

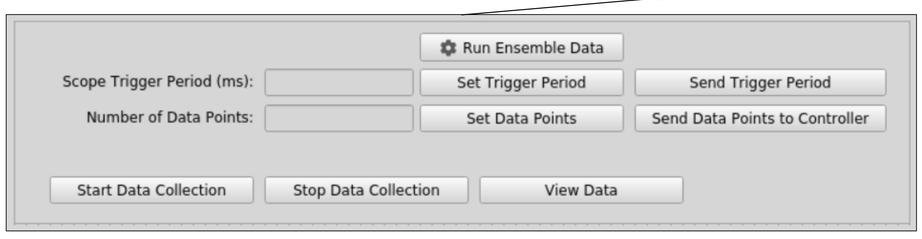
User Interface

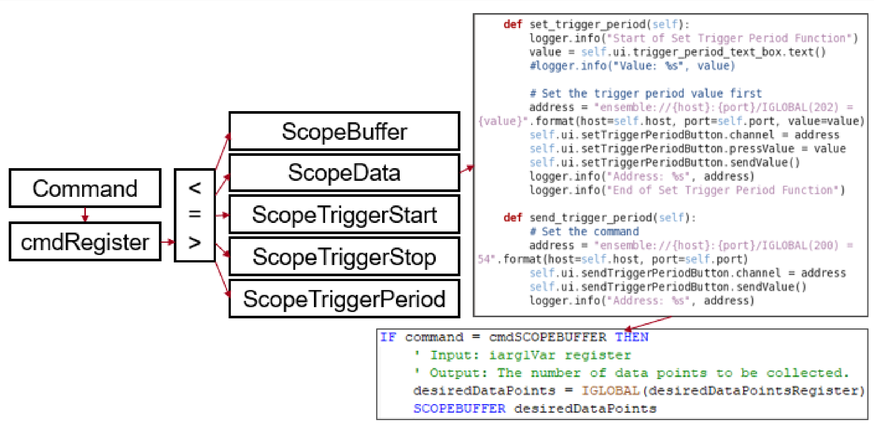
UnitsName	deg
AxisType	0
CommutationOffset	0
ReverseMotionDirection	0
CountsPerUnit	278
ServoRate	5
ServoSetup	0x5
GainKpos	11


```
from pydm import Display
import logging
import aerotech
from os import path

logger = logging.getLogger(__name__)

class ensemble_committing_parameters(Display):

    def __init__(self, parent=None, args=None, macros=None):
        #logger.warning("Done with Initialization")
        super(ensemble_committing_parameters, self).__init__
(parent=parent, args=args, macros=macros)
        self.host = macros['host']
        self.port = macros['port']
        self.port = int(self.port)
        self.rate = macros['rate']
        self.axis = macros['axis']
        self.ui.pushButton_set_parameter.clicked.disconnect()
        self.ui.pushButton_set_parameter.clicked.connect
(self.send_command_parameters)
        #logger.info(args)

    def send_command_parameters(self):
        parameter = self.ui.parameter_text_box.text()
        value = self.ui.value_text_box.text()
        address = "ensemble://{host}:{port}/@{rate}/SETPARM @
{axis},{parameter},{value}".format(host=self.host,
port=self.port, rate=self.rate, axis=self.axis,
parameter=parameter, value=value)
        logger.info(address)
        self.ui.pushButton_set_parameter.channel = address
        self.ui.pushButton_set_parameter.pressValue = value
        self.ui.pushButton_set_parameter.sendValue()
        #logger.info("Done sending command")
```

```
    def ui_filename(self):
        return 'ensemble_committing_parameters.ui'

    def ui_filepath(self):
        # Return the full path to the UI file
        return path.join(path.dirname(path.realpath
(__file__)), self.ui_filename())
```

```
' List of commands
DEFINE cmdSCOPEBUFFER          50
DEFINE cmdSCOPEDATA            51
DEFINE cmdSCOPETRIGSTART       52
DEFINE cmdSCOPETRIGSTOP        53
DEFINE cmdSCOPETRIGPERIOD      54

' List of registers
DEFINE cmdRegister             200
DEFINE collectionStatusRegister 201


DEFINE triggerPeriodRegister   202
DEFINE desiredDataPointsRegister 203
```

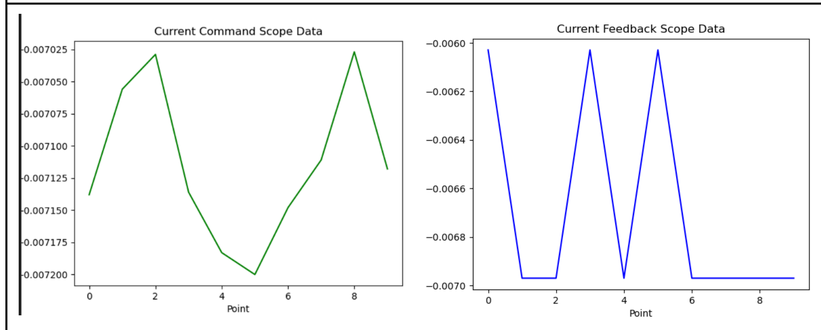




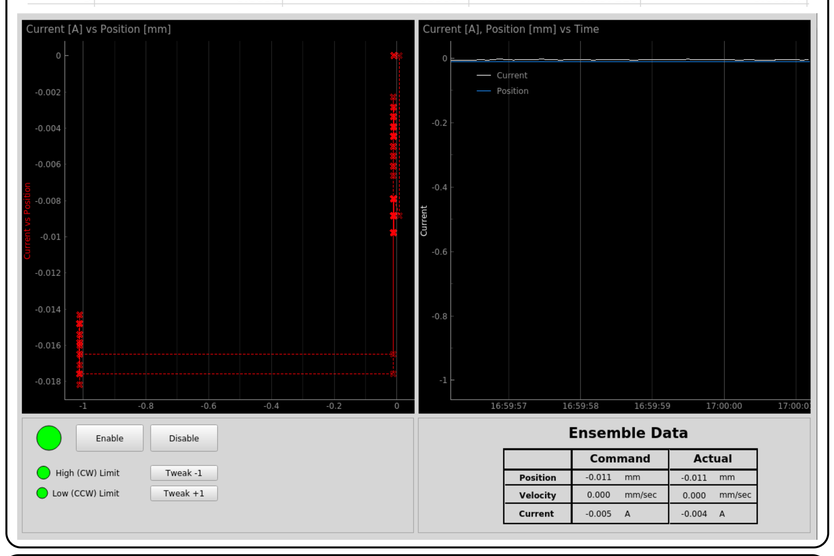

```
    def set_trigger_period(self):
        logger.info("Start of Set Trigger Period Function")
        value = self.ui.trigger_period_text_box.text()
        #logger.info("Value: %s", value)

        # Set the trigger period value first
        address = "ensemble://{host}:{port}/IGLOBAL(202) =
{value}".format(host=self.host, port=self.port, value=value)
        self.ui.setTriggerPeriodButton.channel = address
        self.ui.setTriggerPeriodButton.pressValue = value
        self.ui.setTriggerPeriodButton.sendValue()
        logger.info("Address: %s", address)
        logger.info("End of Set Trigger Period Function")

    def send_trigger_period(self):
        # Set the command
        address = "ensemble://{host}:{port}/IGLOBAL(200) =
54".format(host=self.host, port=self.port)
        self.ui.sendTriggerPeriodButton.channel = address
        self.ui.sendTriggerPeriodButton.sendValue()
        logger.info("Address: %s", address)
```

```
IF command = cmdSCOPEBUFFER THEN
    ' Input: iarglVar register
    ' Output: The number of data points to be collected.
    desiredDataPoints = IGLOBAL(desiredDataPointsRegister)
    SCOPEBUFFER desiredDataPoints
```

User Interface (cont.)



Point	Position Command (U) (deg)	Position Feedback (U) (deg)	Current Command (U) (A)	Current Feedback (U) (A)
0	-0.0036	-0.003604	-0.007138	-0.006029
1	-0.0036	-0.003475	-0.007056	-0.00697
2	-0.0036	-0.003604	-0.007029	-0.00697
3	-0.0036	-0.003665	-0.007136	-0.006029
4	-0.0036	-0.003604	-0.007183	-0.00697
5	-0.0036	-0.003536	-0.0072	-0.006029
6	-0.0036	-0.003604	-0.007148	-0.00697
7	-0.0036	-0.003665	-0.007111	-0.00697
8	-0.0036	-0.003787	-0.007027	-0.00697
9	-0.0036	-0.003604	-0.007118	-0.00697



Future Steps

This project only tests one object: the servo motor. The three software programs mentioned in the introduction section are used to control the object. To ensure that the motor works, correct configuration settings need to be in place in the Configuration Manager by connecting to the microcontroller, having the right values for the parameters, and ensuring that the computer and the microcontroller are connected. Different motors can be tested in the future like the stepper motor to view the parameters that are not applicable to the current motor.

Some Ensemble controllers have the ability to control multiple motor axes. The current interface only support one axis. In the future, this can be expanded to multiple axes.

Acknowledgments

Acknowledgements also go to my mentor Namrata Balakrishnan, and my coworkers who helped me with this project: Yekta Yazar, Ryan McClanahan, Matt Gibbs, Zach Domke, and Gibreel El-Halabi.

Special thanks to Kristi Luchini, An Le (my brother), and SLAC staff for this internship and showing me around campus

Date: 8/11/2022