Distributed I/O for Dynamic Equipment

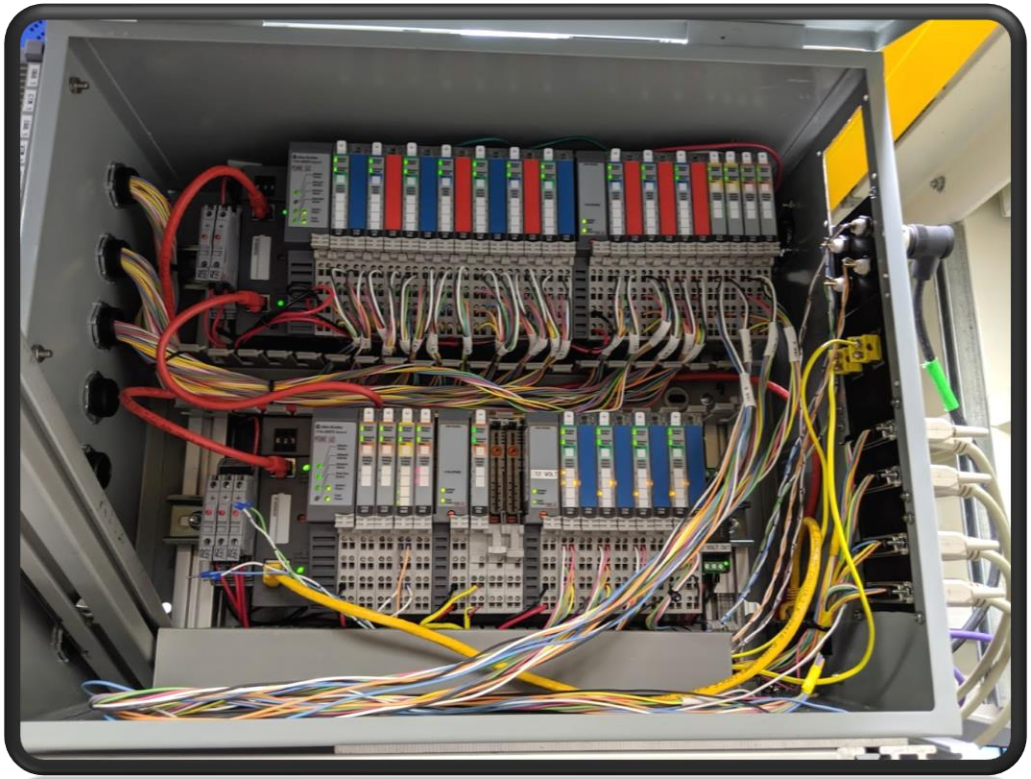
Nathanael Maytan
Anton Derbenev

10/05/19

U.S. DEPARTMENT OF ENERGY | Office of Science
BROOKHAVEN NATIONAL LABORATORY

Contents

- Introduction
- Background
- Scope
- Design
- Hardware
- Software
- Project Status
- Next Steps
- End/Questions



Introduction

- Beamlines need to integrate lots of different equipment
- Some equipment requires protection or protects something itself (i.e. other devices or staff)
- For these devices, dedicated protection systems exist at each beamline
- Other equipment does not have protection implications, but may still need to be integrated
- We call this category of devices “dynamic equipment”
- DIODE is a PLC-based solution aimed at the need for dynamic equipment integration
- DIODE makes it possible to integrate equipment in a flexible, safe, and maintainable way while providing a system that is versatile, extensible, and reliable

Background

- There are currently two major controller-based systems in use at each NSLS-II beamline: the Equipment Protection System (EPS) and the Personnel Protection System (PPS)
- Historically, dynamic equipment would be lumped onto the EPS
  - The EPS is much more mutable than the PPS, which is more strictly controlled
  - The EPS is typically more physically-distributed throughout the beamline
- While using the EPS has been a simple solution to the growing need for dynamic equipment integration, there are real consequences to that approach
  - Every time the EPS is touched, there is a risk of introducing mistakes or errors
  - Unnecessary complexity is added within the EPS.
  - I/O boxes become overloaded
- DIODE was proposed as a third controller-based system to enhance the reliability and maintainability of the beamline subsystems
- With DIODE, the existing problems surrounding dynamic equipment integration can be avoided
  - Using DIODE does not require introducing changes to the EPS
  - New devices can be added or modified more quickly with less overhead

Scope

- Each beamline will have its own DIODE installation
- DIODE will support remote I/O boxes for distributed control capability
- Existing dynamic equipment at a beamline can be migrated to DIODE
- EPS/PPS functions should remain in their corresponding systems
- DIODE software, hardware-base, and delivery will be uniform across every beamline

Design

There were many requirements to consider in forming a solution which can be applied uniformly across all beamlines

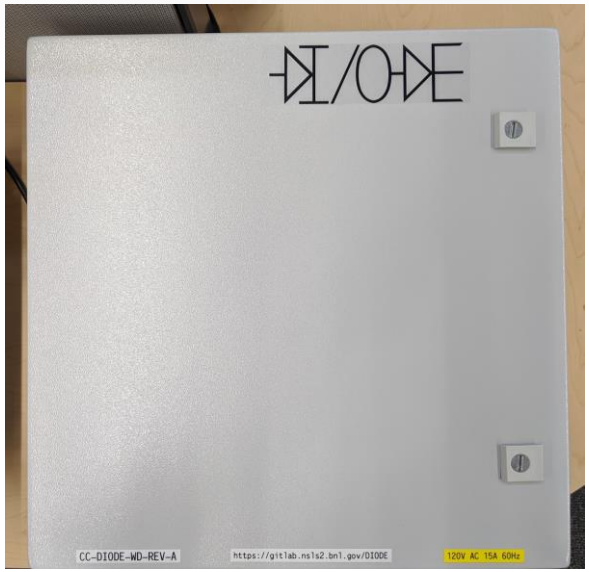
Amount of remote nodes	Possibilities for complex controls and automation	Flexible installation requirements	Room for expansion	Program storage space
Electrical Safety	Compatibility with existing hardware/spares	Component, fabrication, and installation costs	On-line code modifications	Sufficient passive heat dissipation
Availability of replacement or alternative parts	Maintenance and support overhead	Support for device hot-plugging	Facility-wide upgradability	Self-help options
	Separate I/O and controls network interfaces	Completeness of solution	Cable ampacity and hardware derating	

Hardware – Short Description

- Allen-Bradley 5069-L310ER PLC
  - Supports up to 24 remote nodes and has 1MB of storage
  - Dual ethernet ports can be configured with independent IPs
  - Compatible with existing facility components
  - Expertise from EPS team can be carried over
- Default box layout includes one OB16 module and one IB16 module
  - 16-digital outputs and 16-digital inputs
- 4x fused 24V supply outputs + 4x I/O network ethernet ports
  - Plus one controls network port
- 10A Power supply to power both the controller and remote nodes
- High-density terminal blocks provide plentiful source/return connections for equipment
- Housed in a typical NEMA-rated electrical enclosure (20x20x8 in.)
- Plugs into a standard 120V outlet (NEMA 5-15P)
- Placement is up to the beamline, typically inside of a hutch

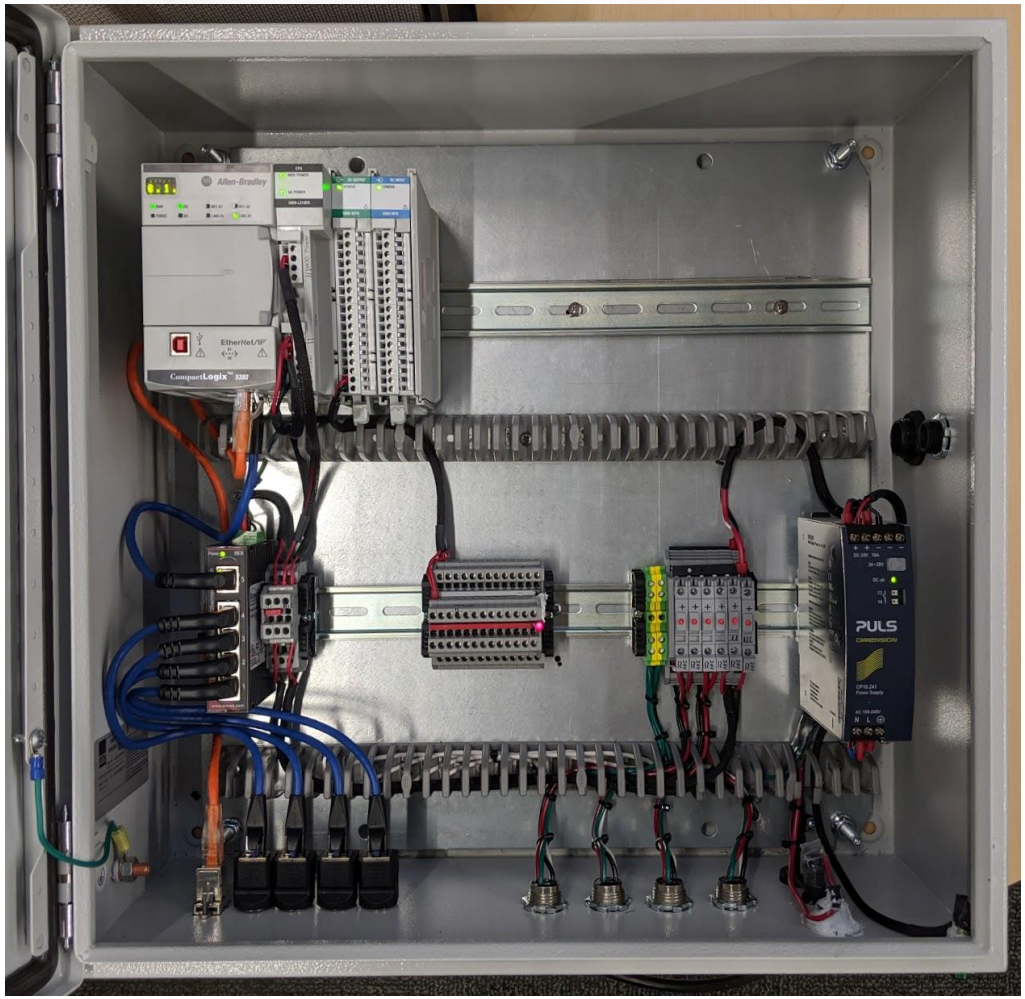
Hardware – Box Pictures

Front



Bottom

Inside



Software – IOC Overview

- EPICS IOC uses EtherIP driver for CompactLogix controllers
- IOC record substitutions are generated via bash script for every PLC module in a DIODE installation
- The IOC will only attempt to build databases for which a substitutions file is found
- PLC module I/O channels are immediately usable through “generic PVs”, thus enabling device hot-plugging without software changes
- New PLC modules are supported by adding a corresponding database template in the IOC source code
- Custom databases can also be included with the DIODE IOC by an available bash script without manually modifying the IOC
- The goal is to make things as self-help friendly as possible. No in-depth expertise is needed to swap devices or add record aliases for a channel

Software – SNACK Delivery

- The DIODE IOC is deployed and version controlled using SNACK, an Ansible-based configuration management tool
- DIODE IOC source code lives in its own repository, whereas configuration entities are kept in our central configuration repository
- Each IOC instance requires only a beamline-specific SNACK template file (and any optional custom IOC databases)
- All other application customization comes through configuration templates
- PV-generating scripts are made into templates and handled by SNACK, though they can be used on their own

```
TASK [new_deployapp : Perform local builds - CUSTOM] ********************************
ok: [xf11bm-ioc1] => (item=diode)

TASK [new_deployapp : Set ownership for the deployed application] ********************
changed: [xf11bm-ioc1]

TASK [new_deployapp : Perform actions at 'after-deploy' hook - CUSTOM] ***************
changed: [xf11bm-ioc1 -> xf11bm-ioc1] => (item=action command "./diode-acf.sh")
changed: [xf11bm-ioc1 -> xf11bm-ioc1] => (item=action command "./ioc-autosave.sh")
changed: [xf11bm-ioc1 -> xf11bm-ioc1] => (item=action command "./ioc-start.sh")

TASK [new_deployapp : Create timestamp for logging] *********************************
ok: [xf11bm-ioc1]

TASK [new_deployapp : Create application log path if it doesn't exist] ***************
changed: [xf11bm-ioc1]

TASK [new_deployapp : Set text to write to app log] *********************************
ok: [xf11bm-ioc1]

TASK [new_deployapp : Write text to application log file] ****************************
changed: [xf11bm-ioc1]

TASK [new_deployapp : Create central log path if it doesn't exist] *******************
ok: [xf11bm-ioc1 -> localhost]

TASK [new_deployapp : Set text to write to central log] ******************************
ok: [xf11bm-ioc1 -> localhost]

TASK [new_deployapp : Write text to log file] ****************************************
changed: [xf11bm-ioc1 -> localhost]

TASK [new_deployapp : Remove local copy of the application image] ********************
changed: [xf11bm-ioc1 -> localhost]

TASK [new_deployapp : Remove local copy of app-specific config] **********************
changed: [xf11bm-ioc1 -> localhost]

TASK [new_deployapp : Remove local copy of app-specific templates] *******************
changed: [xf11bm-ioc1 -> localhost]

TASK [new_deployapp : Remove local copy of app-specific sources] *********************
changed: [xf11bm-ioc1 -> localhost]

TASK [new_deployapp : Remove lockfile] ***********************************************
changed: [xf11bm-ioc1 -> localhost]

PLAY RECAP ***************************************************************************
localhost                  : ok=0    changed=0    unreachable=0    failed=0
xf11bm-ioc1                : ok=85   changed=39   unreachable=0    failed=0

nmaytan@ansible02:~/deploytools/snack-kit$
```

Software – SNACK Template File

This list determines what SNACK templates are included with the app during deployment

Defines the macros in the template startup script

If used, this defines the macro that sets the logserver host

Describes what modules are in a given I/O box so that PVs can be generated

Sets what folder in the app config contains the custom databases to include

Defines which hosts have write permissions

```
# Template file for DIODE IOC deployment

# The list of templates to include during deployment
templates:
  - DIODE/diode                         # The main DIODE IOC template
#  - DIODE/diode-logserv                 # Enables ioclogserver in st.cmd
  - DIODE/diode-modules/diode-box1      # Generates generic PVs for a DIODE box (1-25 available)
  - DIODE/diode-customdb                # Adds custom databases to IOC compilation
  - DIODE/diode-acf                     # Adds access security definitions for the DIODE IOC
  - ioc-stop                            # Automatically stops the current diode IOC (if it exists)
  - ioc-autosave                        # Add autosave directories to the IOC
  - ioc-start                           # Automatically start the newly deployed IOC

# There are several macros for the diode ioc:
# 1. DIODE_ENGINEER  - the engineer deploying the IOC
# 2. DIODE_LOCATION  - where the controller is physically located
# 3. DIODE_IP_ADDR   - the IP address of the controller
# 4. DIODE_BROADCAST - the broadcast IP, typically <PLC IP subnet>.255
# 5. DIODE_SYSTEM    - typically the beamline prefix
# 6. DIODE_NAME      - the desired name for the DIODE PLC
# 7. DIODE_HOST      - the host where the IOC is to be deployed
DIODE/diode:
  - $(DIODE_ENGINEER): Nate Maytan x6237
  - $(DIODE_LOCATION): C Hutch
  - $(DIODE_IP_ADDR): 10.19.2.50
  - $(DIODE_BROADCAST): 10.19.2.255
  - $(DIODE_SYSTEM): XF:19ID
  - $(DIODE_PLC_NAME): XF19ID_DIODE
  - $(DIODE_HOST): xf19id1-ioc1

# There is only one macro for this template:
# 1. DIODE_HOST - the host which runs the logserver
#DIODE/diode-logserv:
#  - $(DIODE_HOST): xf19id1-ioc1

# There are several macros for the diode-box templates:
# 1. DIODE_SYSTEM    - typically the beamline prefix
# 2. DIODE_PLC_NAME - the desired name for the DIODE PLC
# 3. DIODE_BOX_NAME - the name of the box as it is in the PLC program
# 4. DIODE_SLOTS     - a mapping of the modules in the specified box
# NOTE: if duplicate slots are defined, the last definition will be used
DIODE/diode-modules/diode-box1:
  - $(DIODE_SYSTEM): XF:19IDC-CT
  - $(DIODE_PLC_NAME): XF19ID_DIODE
  - $(DIODE_BOX_NAME): Local
  - $(DIODE_SLOTS): >-
      ["Slot1"]="5069-OB16"
      ["Slot2"]="5069-IB16"

# There are two macros for this template:
# 1. DIODE_PLC_NAME  - the desired name for the DIODE PLC
# 2. DIODE_CUSTOM_DB - the relative path to custom databases in the config
# NOTE: custom databases will be loaded in alphabetical order
DIODE/diode-customdb:
  - $(DIODE_PLC_NAME): XF19ID_DIODE
  - $(DIODE_CUSTOM_DB): diode-custom

# There is only one macro for this template:
# 1. DIODE_ACCESS - the list of hosts which should have access rights
# NOTE: the DIODE_ACCESS macro expects a space-separated list of hosts.
DIODE/diode-acf:
  - $(DIODE_ACCESS): xf19id1-ioc1 xf19id1-ws2 xf19id1-ws3
```

These will do several things:
- Add the template startup script to the app
- Generate PVs for the DIODE “Local” box
- Add custom databases to the IOC
- Set the specified hosts in the ACF
- Stop the existing IOC
- Create autosave folders
- Start the new IOC when deployment is finished

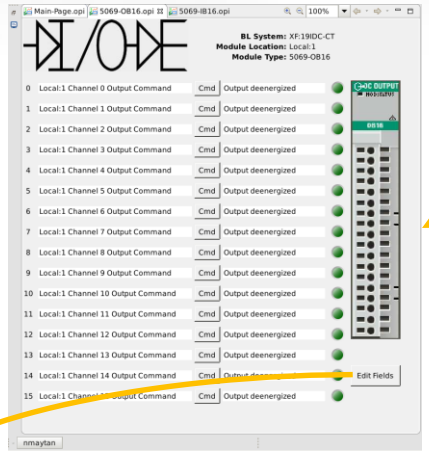
A “diode-box” template should be included for every I/O box. Each template can have different modules defined in any order. The first box is named “Local” according to the Studio 5000 defaults

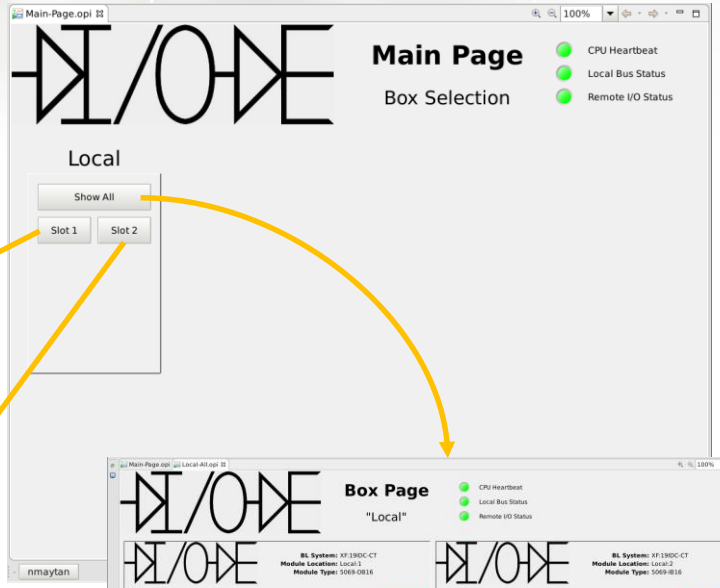
Software – User Interface

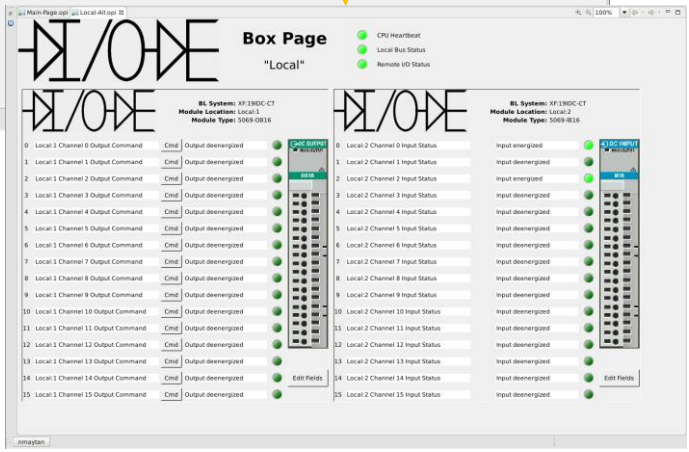
- DIODE comes with a set of CS-Studio OPI screens
- Each supported module has its own OPI, while each beamline has its own DIODE page
  - Module OPIs give control and readback for every channel and allow for configuration of related I/O settings
  - Beamline-specific DIODE pages will provide an overview of available boxes and their installed modules, allowing any box or slot to be accessed

Software – CS-Studio Pages

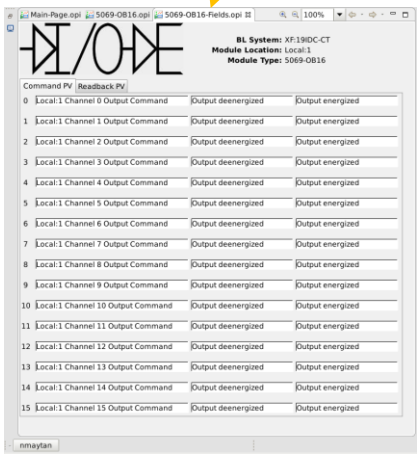
Module OPI for Slot 1



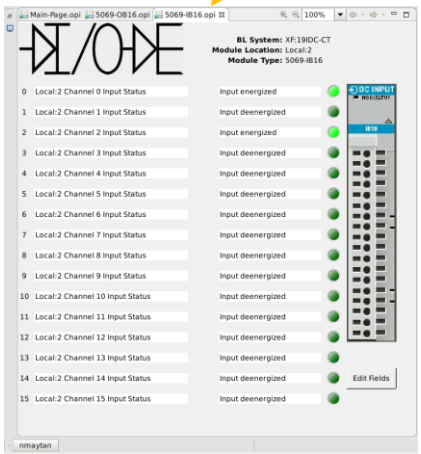




Box Page – Shows all slots on one screen



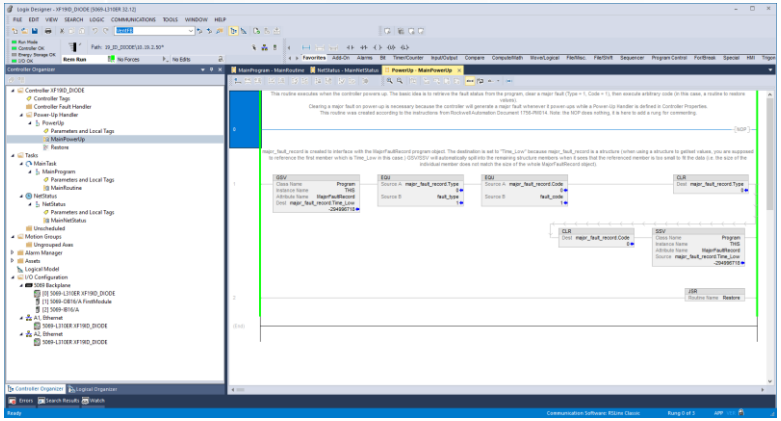
Module config for Slot 1

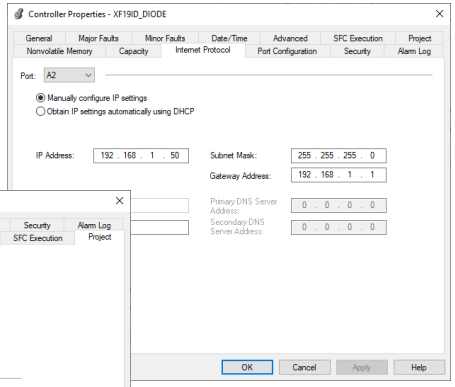


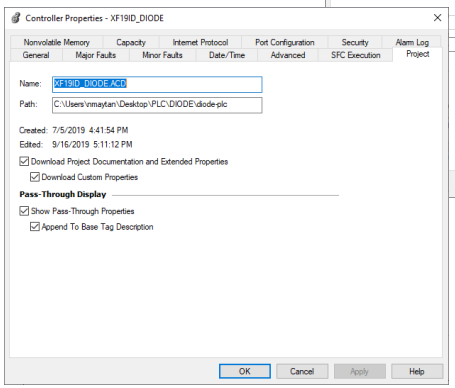
Module OPI for Slot 2

Software – PLC Program

- A starter PLC-program is downloaded to each controller
  - Developed in Studio 5000 v32 (latest)
  - Enables power-up restore, comm status checks, and controller heartbeat
- Controllers run AB firmware major revision v32
  - Unlike older controllers, newer software base supports important new features such as:
    - Preserve comments when uploading to the controller
    - Independent IP assignment per ethernet port







Project Status

- DIODE has been successfully installed at two beamlines: 11-BM CMS and 19-ID NYX
- At NYX, some equipment has already been integrated and is controlled by generic PVs (pneumatic screens)
- Boxes have been assembled for 8-ID ISS and 22-IR FIS/MET – boxes are awaiting integration
- Assembly of the next box has started and will be installed at 17-BM XFP
- The DIODE IOC is available for on-demand deployment and configuration through SNACK, complete with CS-Studio screens
- DIODE is being supported and can be shipped as a complete solution to dynamic equipment integration requests

Next Steps

- Continued roll-outs to the beamlines in queue:
  7-BM QAS, 8-ID ISS, 17-BM XFP, 22-IR FIS/MET
- Support for more modules through generic PVs (implies database templates and module OPIs)
- DIODE setup to be installed in the Controls Lab for equipment and integration testing
- More documentation updates and additions
- Gather feedback from first adopters
- Identify use-cases for DIODE at other beamlines
- If there is demand, introduce some self-configurable control tools (i.e. PID loops, pulse generators, timers, etc.)
  - The uniformity that DIODE provides means that new features can be rolled out facility-wide
- DIODE Trainings?

End

Thank you for your time

Questions?