

A GENERAL SOLUTION FOR COMPLEX VACUUM SYSTEM CONTROLS

G. Bischof, A. Barbour, A. Walter, B. Sobhani
 Brookhaven National Laboratory, Upton, New York, USA

Abstract

At the National Synchrotron Light Source II (NSLS-II) there are many different ultra-high vacuum system configurations on the unique beamline end-stations. The proposed controls solution attempts to capture the requirements of all of these configurations with a single standard logic and graphical user interface. Additional design considerations include: resource management for multiple users, providing a high level of abstraction to simplify operation for users, providing a high level of flexibility to do nonstandard operations, minimizing shock from pressure differentials when opening valves, supporting a variety of pumps, and maximizing pump lifetime. At NSLS-II it was determined that all vacuum configurations can be captured by the composition of three standard objects: a "rough vacuum group", "high vacuum group", and a "smart vacuum manifold" which implements a blocking queue. These objects can be flexibly linked together to meet the needs of the beamline experiments. This solution is platform independent, but implemented and tested here using Pfeiffer vacuum Pumps, Allen Bradley PLC, EPICS, and Control System Studio (CSS).

mistakes. This project attempts to address these challenges, and simply the operation of end station vacuum systems.

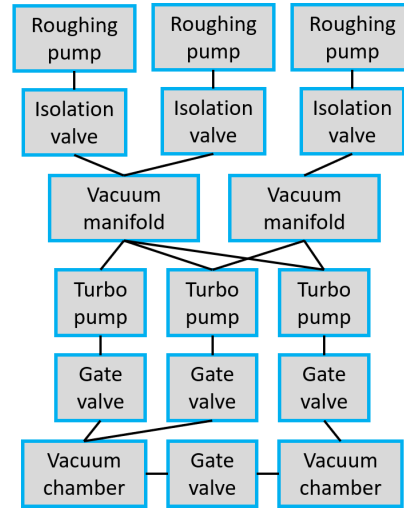


Figure 1: Example vacuum system configuration.

INTRODUCTION

At the National Synchrotron Light Source II (NSLS-II) there are many different ultra-high vacuum system configurations on the unique beamline end-stations. The end-station vacuum systems often differ from the vacuum systems of the rest of the accelerator and beamline in that they frequently need to have sections vented. They also often have unique vacuum requirements for their experiments. End-station vacuum systems often incorporate roughing pumps and turbo pumps to meet their pumping speed and throughput requirements. End-station vacuum systems also incorporate ion pumps and NeG pumps to achieve the best possible vacuum. Some configurations are quite simple, for example: a single vacuum chamber with a vacuum system consisting of a roughing pump, isolation valve, turbo pump, and a gate valve. Other vacuum system configurations are more complex. Figure 1 shows a vacuum system configuration that is typical for a more complex end-station. Vacuum system configurations can vary greatly between end-stations.

There are challenges that come with working with these vacuum systems. Multiple users can be executing independent operations on the vacuum system that require access to the same resource. Currently this is handled by staff coordination. Due to the complexity of these vacuum systems, users are often uncomfortable with executing vacuum system operations. Even expert staff can make

REQUIREMENTS

The goal of this project is to create a modular standard logic and Graphical User Interface (GUI) that supports all of the different end-station vacuum configurations. The design should simplify user operation, by automating common operations, and by providing an abstraction to hide the complexity of the system. The automation of the common operations should reduce the chance for human error and improve response to faults. Users of the system may also need to do a non-standard operation not supported by automated logic, so the solution must also allow for manual control of all output signals. The GUI should clearly and simply communicate vacuum status to its users. The modular standard logic and GUI should make it more efficient to automate new vacuum systems by allowing a copy and paste of standardized modules and linking them together to meet the vacuum requirements.

SYSTEM DESCRIPTION

The end-station vacuum systems at the NSLS-II are composed of a small set of hardware devices: Roughing pumps, vacuum gauges, flow meters, angle valves, gate valves, turbo pumps, ion pumps, and NeG pumps. The signals of the vacuum devices have been wired to distributed PLC I/O modules. The distributed I/O communicates to the PLC via ethernet. At NSLS-II we use EPICS, which is a distributed real-time control system and SCADA, which allows us to integrate many independent control

systems and devices over ethernet. A distributed control system like EPICS is not required for this solution.

As shown in Figure 2, a unique part of this solution is the logical grouping of the vacuum devices into three standard objects. These objects can be linked together in different arrangements to meet the various vacuum requirements. It was determined that the composition of these standard objects can meet the requirement for all end-station vacuum systems at the NSLS-II. The three standard objects are the "rough vacuum group", "high vacuum group", and "smart vacuum manifold". These objects each have their own function in the PLC logic and their own templated GUI. The functions in the PLC logic, as well as the GUI for each object, can be linked together to represent the entire vacuum system configuration. To provide a complete end-station vacuum solution the vacuum groups can be integrated with additional vacuum components such as valves, ion pump, and NeG pumps.

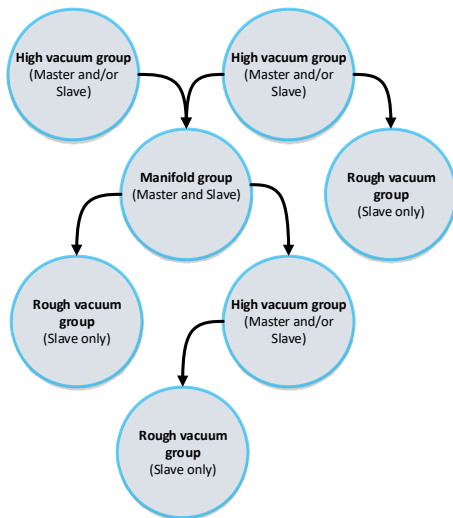


Figure 2: Example of object hierarchy.

VACUUM GROUP OBJECTS

Each vacuum group object consists of PLC logic, an EPICS template file (used when implemented with EPICS), a detailed GUI, and a minimalistic abstracted GUI. This section will describe these features that are common to all vacuum group objects.

PLC Logic

In order to simplify the logic, it does not use a finite state machine. Instead, each output signal is given an independent logic. The functionality of the vacuum group is best explained by its output logic. All vacuum group objects have "Force mode". If the force mode option is selected, the user can select the value for each output signal manually. This gives the user the ability to do any operation that they want. After disabling force mode all outputs will return to their automated outputs determined the requested vacuum state. Each vacuum group has a set of parameters which can be used to specialize each instance of the object. Critical parts of the PLC logic are presented in this paper.

Detailed GUI

The detailed GUI displays the value of all signals, allows the user to enable and use force mode, and set the value of the vacuum group instance's parameters (see Figure 3).

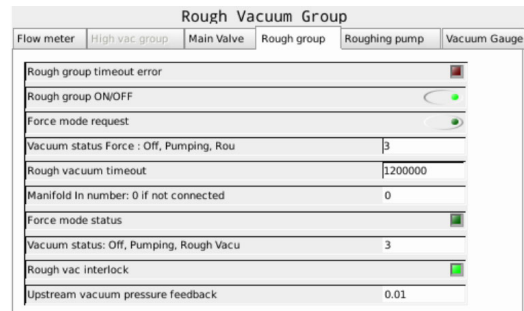


Figure 3: Rough group detailed GUI.

Abstract GUI

The abstract GUI is designed to be minimalistic, intuitive, and hide a reasonable amount of detail about the system. Boolean statuses are displayed as red/green indicators. For example, if the vacuum is better than the vacuum threshold parameter, the vacuum pipe will turn green. The abstract GUIs also display alarm messages if there are any alarms present. The abstract GUI is made with all SVG graphics. The abstract GUIs can be linked together to depict the entire system.

ROUGH VACUUM GROUP

A rough vacuum group (Figure 4) is responsible for supplying rough vacuum to a chamber, the high vacuum group or manifold. The rough group has only two states, "On" and "Off". It can be turned "On" by pressing the power symbol on the GUI, or by a request from a master (manifold or high vacuum group). Clicking the menu icon will open the detailed GUI.

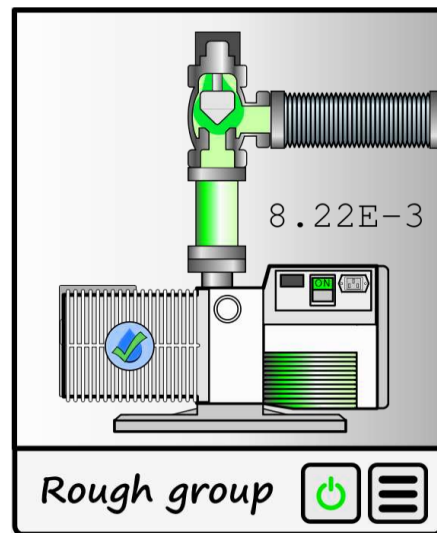


Figure 4: Rough vacuum group abstract GUI.

Components: Roughing pump, water flow meter/switch, vacuum gauge, isolation valve (main valve).

Input signals: Roughing pump water flow, roughing pump on/off status, roughing pump warning, roughing pump alarm, rough vacuum value, upstream vacuum gauge value, main valve opened feedback, main valve closed feedback.

Output signals: Roughing pump on/off command, main valve command.

Command inputs: Rough group on/off, force mode request.

Status outputs: Force mode status, vacuum status.

Parameters: Flow meter low threshold, flow meter lolo threshold, rough vacuum timeout, manifold input number, rough flow meter option, vacuum gauge high threshold, vacuum gauge hihi threshold.

Hard-stop alarms: Roughing pump water flow fault, main valve feedback error, alarm signal from roughing pump, roughing pump status feedback error.

```

If RoughVac < (0.95 * UpstreamVac) then
    InputLessThanUS := 1;
end_if;
If RoughVac > (1.3 * UpstreamVac) or not Output_RPOnOff then
    InputLessThanUS := 0;
end_if;
    
```

Code block 1: Rough group intermediate logic

```

If CmdRoughOn and Interlock_RoughOK and not HardStopAlarm then
    Output_RPOnOff := 1;
end_if;

if (Input_MainVlvClosed or Alarm_MainVlvFB)
and (not CmdRoughOn or not Interlock_RoughOK or HardStopAlarm) then
    Output_RPOnOff := 0;
end_if;

if CmdRoughOn and Interlock_RoughOK and not HardStopAlarm
and Input_RPOnSts and (InputLessThanUS or RoughVacGood) then
    Output_MainVlvCmd := 1;
end_if;

if not CmdRoughOn or not Interlock_RoughOK or HardStopAlarm
or not Input_RPOnSts then
    Output_MainVlvCmd := 0;
end_if;
    
```

Code block 2: Rough group output logic

HIGH VACUUM GROUP

The high vacuum group consists of a turbo pump, roughing line vacuum gauge, high vacuum gauge, and an optional gate valve (GV) on the inlet side of the turbo (Figure 5). The high vacuum group can be used to generate high vacuum in a chamber, or in a manifold. The logic for the high vacuum was developed to not require a vacuum gauge between the GV and turbo inlet. The high vacuum group can function as a master, by making requests from a smart manifold, or a rough group. It can also function as a slave and respond to requests from a manifold. It can also be setup to do both at the same time.

The high vacuum group has several states that can be requested, “Play”, “Pause”, “Stop”, and “Vent”. These automated states should cover the most of the operations required by the users. In the “Play” state, the turbo is running, and GV is open. In the “Pause” state, the turbo is running and the GV is closed. In the “Stop” state, the GV

is closed, the turbo is stopped, and the turbo section is vented if the vent option is enabled. In the “Vent” state, the GV is open, the turbo is stopped, the chamber and turbo section are vented. The “Vent” state can be requested through the detailed GUI.

The most critical section of logic for all of the vacuum groups is the logic of the GV command. Opening this valve at the wrong time can damage the turbo, or components in the chamber. The parameters, “MaxDifferential” and “TurboMaxPressure” should be selected carefully. “MaxDifferential” is the maximum pressure differential across the GV where it is still safe to open the GV. A smaller value for this parameter will reduce shock on components in the chamber. “TurboMaxPressure” is the maximum chamber pressure where it is safe to open the GV on a full speed turbo. It is recommended to contact your turbo manufacturer to get this information.

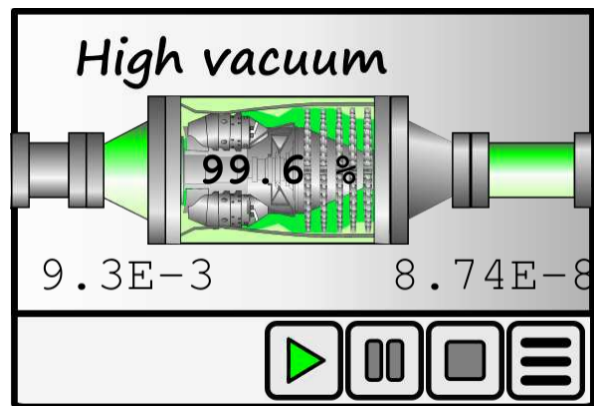


Figure 5: High vacuum group abstract GUI.

Components: Turbo pump, roughing line vacuum gauge, high vacuum gauge, gate valve (optional).

Input signals: Turbo error, turbo speed, gate valve position feedback, fore-line vacuum pressure, chamber vacuum pressure.

Output signals: Turbo auto vent, turbo on/off, gate valve command.

Command inputs: “Play”, “Pause”, “Stop”, “Vent”, force mode request.

Status outputs: Force mode status, rough vacuum status, high vacuum status, current vacuum state.

Parameters: Gate valve option, turbo vent option, roughing-line vacuum pressure status thresholds, chamber vacuum pressure status thresholds, high vacuum timeout, rough vacuum timeout, smart manifold output port number, MaxDifferential, TurboMaxPressure.

Hard-stop alarms: Turbo pump alarm, GV feedback error.

```

If RoughVacuum < (0.9 * HighVacuum) then
    InputLessThanUS := 1;
end_if;
If RoughVacuum > (1.1 * HighVacuum) then
    InputLessThanUS := 0;
end_if;
    
```

Code block 3: High vacuum group intermediate logic

```

// Cmd_VacStateRequest possible states
// 1 = Stop
// 5 = Play
// 6 = Pause
// 9 = Vent
if (Cmd_VacStateRequest = 5) or (Cmd_VacStateRequest = 9) and Interlock_HighVacOK
and (InputLessThanUS or (ChamberVacLessThanThreshold and TurboSpeedGreaterThan80)
and not GreaterThanMaxDifferential and not HardStopAlarm) then
    Output_MainVlvCmd := 1;
end_if;

if (((Cmd_VacStateRequest = 1) or (Cmd_VacStateRequest = 6))
or not Interlock_RoughOK or not Interlock_HighVacOK or HardStopAlarm)
and Interlock_MainVlvClose then
    Output_MainVlvCmd := 0;
end_if;

Output_TurboOnOff := Interlock_RoughOK and Interlock_HighVacOK and Sts_RoughTCG
and ((Cmd_VacStateRequest = 5) or (Cmd_VacStateRequest = 6));

Output_TurboVentEnable := (Cmd_VacStateRequest = 9) or (not Output_MainVlvCmd
and Input_MainVlvClosed and (Cmd_VacStateRequest = 1) and Param_VentOption);
    
```

Code block 4: High vacuum group output logic

SMART VACUUM MANIFOLD

The smart vacuum manifold is used to share rough vacuum with multiple high vacuum groups or manifolds. The manifold can also be configured to also be backed by a high vacuum group or another smart manifold group. The output side of the manifold is defined as the side that goes to the lowest pressure vacuum. The other side is called the input side, this side in most cases is connected to the rough groups. The smart manifold consists of an array of input valves, and an array of output valves as shown in Figure 6.

The smart manifold acts as both a master and slave. High vacuum groups can make request to manifold for rough vacuum. The manifold can make requests to rough vacuum groups or other high vacuum groups.

Requests to the manifold can be “blocking” requests, or “non-blocking requests”. A blocking request means that all other output valves other than the output that is receiving the request must be closed. If multiple blocking requests are issued at the same time the requests will be added to a FIFO queue in the order that they were received. If there are no blocking requests, then all non-blocking requests can be fulfilled at the same time. This is useful for allowing multiple users to make requests at the same time.

Output valves of the manifold can be configured as “soft-pump” valves. Soft pump valves have two different size openings. The smaller opening size can be adjusted. In situations where is desirable to reduce stresses on chamber components, a soft-pump valve is useful. First the small opening of the valve will open, once the output pressure the below the set threshold the valve will fully open.

Due to the array nature of the smart manifold object it will be simpler to describe it in words and not code.

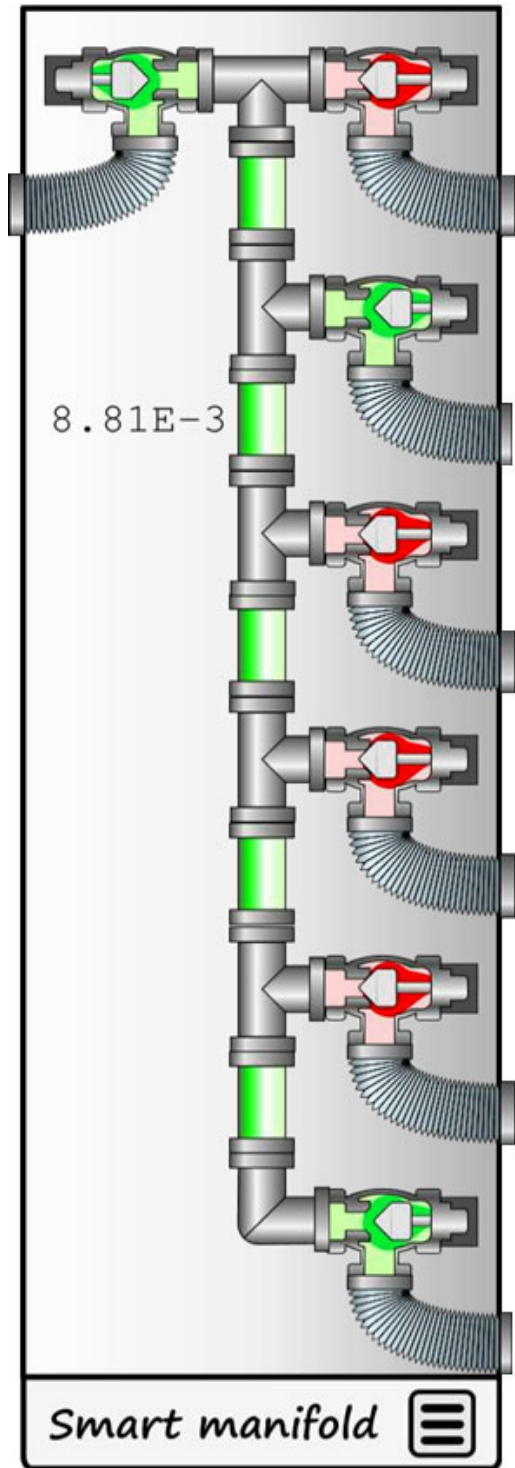


Figure 6: High vacuum group abstract GUI.

Components: Array of output angle valves, array of input angle valves, manifold vacuum gauge, array of output vacuum gauges.

Input signals: Array of output angle valves position feedbacks, array of input angle valves position feedbacks, manifold vacuum pressure, array of output vacuum pressures.

Output signals: Array of input angle valve commands, array of output valve commands.

Command inputs: Force mode request, vacuum state request (array of integers), master block request (32 bit integer).

Status outputs: Force mode status, manifold vacuum status, input array vacuum status, output array vacuum status, blocking status array.

Parameters: Input valve count, output valve count, output valve soft-pump array mask, input array vacuum pressure status thresholds, output array vacuum pressure status thresholds, blocking request timeout.

Hard-stop alarms: None.

Blocking queue: The smart manifold object has an input called “master block request”, which is a Boolean array. Each object that will be making requests for vacuum from the smart manifold can set the bit of the index which matches the output port that it is connected to. This will initiate a blocking request by pushing the index to a FIFO queue. If the item at the front of the queue stops blocking, by clearing its block request, it will be popped. If an item blocks for a time greater than the parameter “blocking request timeout” it will be popped and pushed to the end of the queue. If there are any blocking requests present, only one output valve can be open at a time.

Output logic if there is an item in the queue: Only the output valve for the master at the front of the queue can be opened. To open this valve the closed feedback of all output valves must be observed, and the interlock satisfied. The slow pump output valve will open if the master requesting the “Play” or “Pause” state and the slow pump option is enabled for the port. The corresponding angle valve will fully open when the pressure on the master side becomes better than the output array vacuum pressure status threshold parameter. The output valve will close if the master is no longer requesting the “Play” or “Pause” state, or if the queue has moved to the next item.

Output logic if queue is empty: If the interlock for the output valve is satisfied and the slow pump option is enabled for the port, then the slow pump output valve will open, if the master on the same port is requesting the “Play” or “Pause” state. The corresponding angle valve

will fully open when the pressure on the master side becomes better than the output array vacuum pressure status threshold. The output valve will close if the master is not requesting the “Play” or “Pause” state.

CONCLUSION

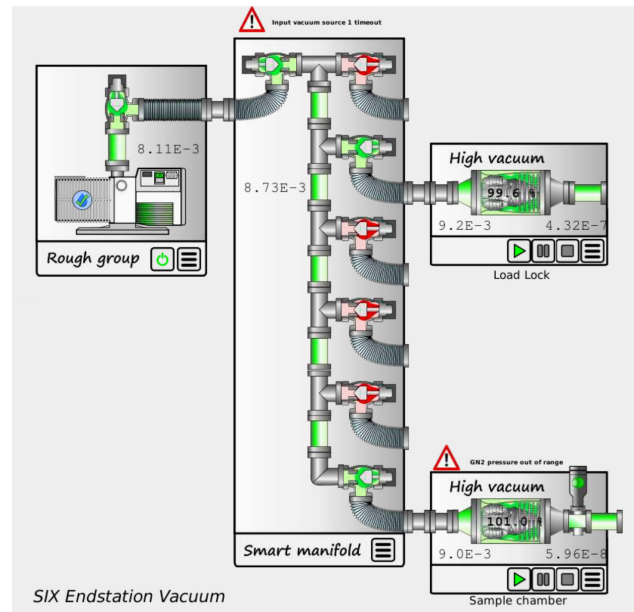


Figure 7: SIX end-station vacuum GUI.

This solution was deployed and tested at the SIX beamline and is currently being used for their end-station. It currently meets the goals defined in the requirements section. Figure 7 shows their current GUI, depicting their current vacuum system configuration. Some polishing is still required, and further development and testing to improve the robustness is planned. The CSX, IOS, and ESM beamlines were also considered and have the existing hardware to roll out the new vacuum control system. This new system will provide unprecedented protection and automation of these beamline vacuum end-stations.