# DESIGN AND CONSTRUCTION OF CONTROLS FOR A KV/MVA CLASS POWER ELECTRONICS TESTING FACILITY

Clinton L. Perdue

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

APPROVED:

Dr. Fei (Fred) Wang, Chairman

Dr. Rolando Burgos

Dr. R. Krishnan

September 8, 2006

Blacksburg, Virginia

Keywords: Power Electronics Test Facility, Recirculating Power Ring, Supervisory Control and Data Acquisition (SCADA)

# DESIGN AND CONSTRUCTION OF CONTROLS FOR A kV/MVA CLASS POWER ELECTRONICS TESTING FACILITY

By

Clinton L. Perdue

Chairman, Dr. Fred Wang

Electrical Engineering

## Abstract

In order to facilitate research and testing of kV/MVA class power electronics systems, Virginia Tech has constructed the High-Power lab facility. This lab supports testing of equipment operating at up to 1.3 MW, with maximum supply ratings of 4,160 V or 480 A, depending on how the system is configured. When operated as a recirculating power ring, the system will make minimal demands on utilities. An industrial supervisory, control, and data acquisition (SCADA) [1] system will be used to control the facility. In this paper we will detail the lab design and give insight to the decisions behind it, with an aim toward helping the reader in their own similar effort.

[1] The terms, “Human-Machine-Interface, HMI” and “Man-Machine-Interface, MMI” have been used to mean the same thing.

# Contents

## Figures

## Tables

# Dedications

Thank you to Susan, who got me through this. Thank you also to all the GE field service employees who added a practical dimension to my education. Finally, thank you to the students, faculty, and staff who listened and put in resources to get this done.

# Acknowledgements

Many thanks to TMEIC-GE (the successor of GE Industrial Systems) for the donation of equipment and to the staff there for their time and encouragement. Thanks also to the staff of GE Supply in Salem, Va. and of GE-Fanuc Automation in Charlottesville, Va. for their assistance both in gathering the materials for this effort and troubleshooting during commissioning.

# Chapter 1 – Introduction

## 1.1 Our Motivations

Power electronics have reshaped our world in the past fifty years by making electrical energy conversion equipment vastly smaller, cheaper, more capable, and more efficient. The ubiquitous cell phone is a good example of this on a low-power scale. The casual observer will have noted that both handsets and chargers are continuously getting smaller and lighter and batteries last longer. Part of the vision of The Center for Power Electronics Systems at Virginia Tech is to realize similar improvements at the high-power scale in large electrical power utilities.

Much of the research and design work that goes into developing a new electronic device or control system depends on computer modeling.[2] Today there exist many simulator platforms (MATLAB, SABER, and others) with numerical models for the well understood aspects of conventional power systems.

Why, then, is CPES interested in building a facility to physically test power electronics? In a word, research. Investigating new concepts in device physics, manufacturing, control schemes, and the like will require not just using computer models, but creating new ones.[3] Those models, in turn, need to be validated by comparing their predictions to real-world test results. Some of this validation can begin on low-power or physical scale models, but truly new work will ultimately require a full-size, full-power prototype.

The goal of the HIGH-POWER LAB project is to build the attendant test facilities required for novel work in utility-scale power electronics. To this end, the Office of Naval Research, ONR, has granted Virginia Tech funding under the DURIP program to build the high-power testing lab.

[2] Wolfram Research, the publishers of MATLAB, have placed a full page advertisement in IEEE Spectrum (usually on the back cover) touting, "model-based design," nearly continuously for *years*.
IEEE Spectrum Magazine, IEEE
On rear cover, Volume: 43, Issue: 8, August 2006

[3] The Power Systems Toolbox in MATLAB is evidence of this. It was written by Hydro-Quebec in the course of their work.

The HIGH-POWER LAB will also serve the more general, educational purpose of allowing students to gain practical experience with a “real-world” system.[4] They will encounter practicalities they may not have otherwise considered. For example, safety precautions and oversight become “serious” when operating in a realm where the safety of people and facilities can not be taken for granted. Finally, it is hoped that the *presence* of large power equipment will make an impression on students who might otherwise have no concept of the scale of these systems inhabit.

## 1.2 Requirements

To be useful, this facility must provide a power distribution system comparable to the environment where the other technologies developed at CPES are intended to operate. This calls for distribution level (2400\4160) AC voltages, DC links at 3.3 kV or greater, and at least 100 kVA continuous power.

The facility must also be able to host multiple projects simultaneously. These projects do not necessarily need to be powered at once, but a variety of connections have to be available and changes between them need to be fairly simple. The operating environment, including operator controls, equipment interlocks, and data recording also needs to facilitate this by being flexible and easy to use.

There is a common sense and ethical, if not legal, requirement to avoid endangering people, which is not casually accomplished with equipment of this scale. This requirement is doubly important because this is an academic institution, not an industrial setting with its attendant safety codes and inspections. Operating procedures need to be developed to keep people out of harm's way while working in the lab. The automatic portion of the lab control system must be as fail-safe as possible in the face of both equipment faults and operator errors.

[4] HIGH-POWER LAB students report prospective employers react quite favorably to their experience with an industrial SCADA system, I/O hardware, networks, and PLC programming.

## 1.3 The prior art – other testing facilities

Testing facilities can generally be broken into four categories, depending on whether or not they have moving parts (static/dynamic), and if they primarily dissipate or recirculate energy (dissipative/regenerative).

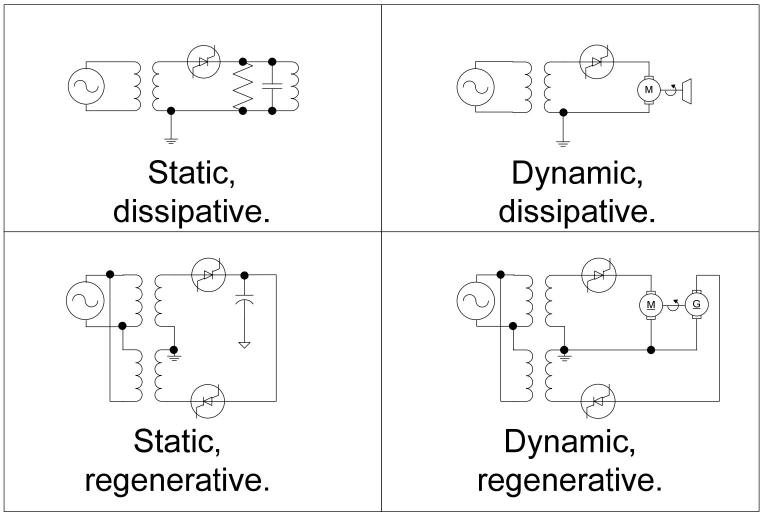


**Figure 1-1; Types of power electronics testing facilities.**

### 1.3.1 Static, dissipative loads

In the most straightforward class of systems, power from the grid is processed in some way by the test equipment and then dissipated. Reactive load elements may change the power factor as they temporarily store energy, but eventually it all is dissipated.

This can be a very realistic setup for simulating heating, lighting, or some other dissipative load that has no mechanical dynamics. The early stages of design for a dynamic load may also benefit from using a simple, well behaved load, because it will not complicate the analysis. CPES has several large air and water cooled resistor banks for this purpose.

### 1.3.2 Dynamic systems

A system with an actual mechanical load is superior in some ways to one with electrical dummy loads for testing electronic power conversion equipment. Motor drives, for example, must deal with *mechanical* realities such as rotor radial displacement, vibration, and acoustic noise. There are integrated design tools which can model all these phenomena simultaneously, but the problem tends to be exceedingly complicated. Running a real machine to validate the model is still quite important[5].

This class of system also provides an opportunity for collaboration with mechanically-minded colleagues, because mechanical dynamics can be investigated just as easily as electrical ones. A "mixed" facility is useful for design and evaluation of mechanical equipment (motors, generators, gearboxes, couplings) and the electrical controls invariably called on to compensate for their shortcomings.

A useful sub-class of mechanical load systems would be motor-generator, MG, sets. In these arrangements, a DC motor turns an AC generator to create a variable voltage and frequency supply[6]. The advantage in a testing facility would be that a MG set could provide a degree of isolation from the world at large.

The logical extension of having a generator in the system is recirculating power back into the utility supply. Besides saving on the power bill, this allows investigations of both loads and generators, along with their interactions with the power grid.

The disadvantage of a mechanical system is that the mechanical element is always there, even when it is not of interest. Beyond simple cost and space requirements, these components will always cause more losses and noise. Bearings, brushes, and the like must be maintained. Large machines often require forced lubrication and cooling, leading to another level of

[5] In further reference to THE MATHWORKS advertising copy, despite having a very good model they *still* built a real machine to test. Ibid., [1].

[6] In the days before electronic AC drives, MG sets were used to power AC motors which had too much inertia to start directly from the utility supply. In truly ancient times, the DC might itself be generated by one or more smaller AC/DC MG sets.

complexity. These support systems significantly complicate the system controls as their status become critical states that must be monitored when the system is operating.

As a point of reference, see Figure 1-2 below for a view of the sort of large machines we are considering. This machine has three lubrication systems, a liquid heat exchanger, a forced air system, and twelve thermocouples mounted in the windings and bearings. The gearbox (not visible) has an additional two lubrication systems, liquid cooling, and more thermocouples. Any load would be similarly complex. Obviously the space and mechanical support requirements are not trivial either.

Far from being unusual, this is one of four 11 MW and five 5 MW units all in a row at this site, a fairly typical industrial facility (the *really* big machines are found in mines and on ships). Once again, we see the need for a large scale testing facility for power electronics and controls[7].

[7] This particular machine has suffered a burned stator winding (one of 144), probably caused by a void in the insulation. The drive system was *probably* capable of compensating for the bad winding, but such an algorithm had never been tested on this scale, and the customer was disinclined to host a "science project." Consequently, the motor manufacturer is replacing this unit at a cost of approximately $US 2,000,000.00.

**Figure 1-2; The author in a former life, working on an 11 MW, 6.6 kV AC induction motor. The cooling and lubrication systems are not installed, but the cooling water lines and heat exchanger can be seen atop the unit to the right. Total volume is about 60 m$^3$, the rotor alone masses about 23 MT and the stator another 36 MT.**

### 1.3.3 Static, regenerative systems

Given modern power electronics, the mechanical elements of a recirculating system are purely optional. Line power can be rectified, inverted, and fed back into the supply. Alternately, the DC bus can be fed from a line rectifier and a test rectifier (one controlling current, the other voltage), with a load inverter returning bus power to the test rectifier[8]. Both options are depicted in the following, Figure 1-3.

[8] This latter topology has been used at the General Electric Company's works in Salem, Va. to test motor drives in the multi-megawatt range.

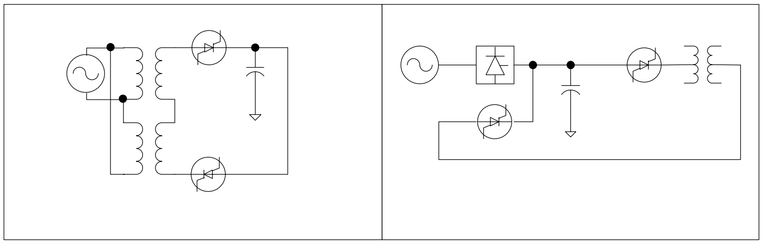


**Figure 1-3; Two options for a static, regenerative system.**

The result of both of these topologies is a realistic static load, with the option to test either rectifiers or inverters, some benefit of recovered energy, and less complications than a mechanical system.

**Table 1-1; Testing topologies' relative strengths.**

| Load type | | Overall complexity | Construction cost | Operation cost | Cooling load | Space requirement | Realism | Flexibility |
|---|---|---|---|---|---|---|---|---|
| Dissipative | Static | Simple | Least | High | High | Least | Limited | Limited |
| | Dynamic | Medium | Medium | Most | High | Medium | High | Better |
| Regenerative | Static | Medium | Medium | Least | Least | Medium | High | High |
| | Dynamic | Very | Most | Medium | Medium | Most | Most | Most |

The relative strengths of the four main approaches are compared in Table 1, above. In some situations there may be other factors to consider, but given these a static, recirculating system is the best value.

## *1.4 The VT HIGH-POWER LAB - available facilities*

With the above analysis in mind, it is not surprising that CPES has opted for a static recirculating system when laying out the high-power lab. The line-side power distribution topology is shown in Figure 1-4.

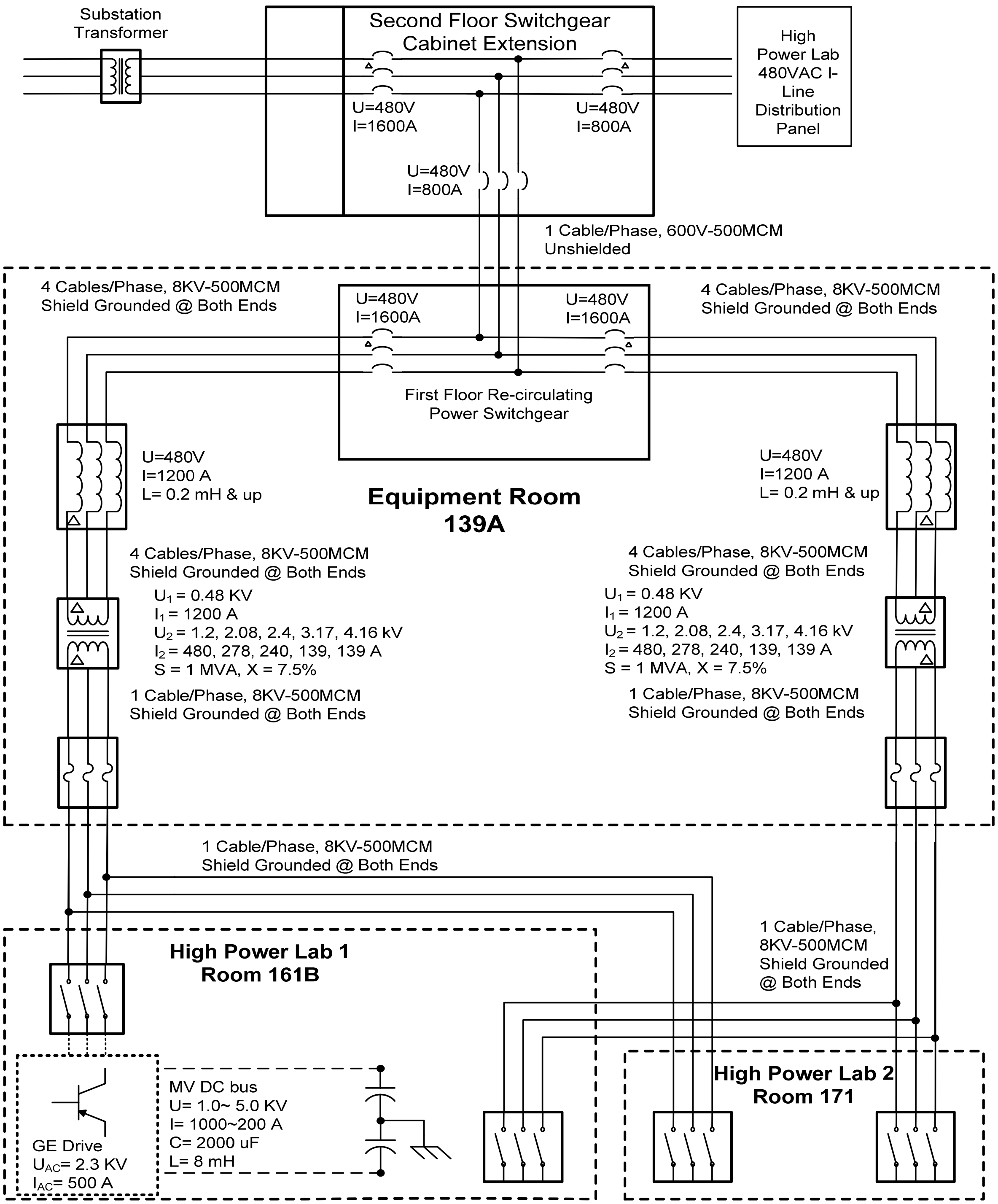

Substation
Transformer
Second Floor Switchgear
Cabinet Extension
U=480V
I=1600A
U=480V
I=800A
High
Power Lab
480VAC I-
Line
Distribution
Panel
U=480V
I=800A
1 Cable/Phase, 600V-500MCM
Unshielded
4 Cables/Phase, 8KV-500MCM
Shield Grounded @ Both Ends
U=480V
I=1600A
U=480V
I=1600A
4 Cables/Phase, 8KV-500MCM
Shield Grounded @ Both Ends
First Floor Re-circulating
Power Switchgear
U=480V
I=1200 A
L= 0.2 mH & up
Equipment Room
139A
U=480V
I=1200 A
L= 0.2 mH & up
4 Cables/Phase, 8KV-500MCM
Shield Grounded @ Both Ends
$U_1$ = 0.48 KV
$I_1$ = 1200 A
$U_2$ = 1.2, 2.08, 2.4, 3.17, 4.16 kV
$I_2$ = 480, 278, 240, 139, 139 A
S = 1 MVA, X = 7.5%
1 Cable/Phase, 8KV-500MCM
Shield Grounded @ Both Ends
4 Cables/Phase, 8KV-500MCM
Shield Grounded @ Both Ends
$U_1$ = 0.48 KV
$I_1$ = 1200 A
$U_2$ = 1.2, 2.08, 2.4, 3.17, 4.16 kV
$I_2$ = 480, 278, 240, 139, 139 A
S = 1 MVA, X = 7.5%
1 Cable/Phase, 8KV-500MCM
Shield Grounded @ Both Ends
1 Cable/Phase, 8KV-500MCM
Shield Grounded @ Both Ends
1 Cable/Phase,
8KV-500MCM
Shield Grounded
@ Both Ends
High Power Lab 1
Room 161B
GE Drive
$U_{AC}$= 2.3 KV
$I_{AC}$= 500 A
MV DC bus
U= 1.0~ 5.0 KV
I= 1000~200 A
C= 2000 uF
L= 8 mH
High Power Lab 2
Room 171

**Figure 1-4; power distribution system elementary diagram.**

Variable transformers allow distribution of between 480 and 4,160 VAC to any of four utility connections. A GE Innovation Series ™ drive, located in the east bay, can provide four-quadrant AC/DC conversion using 3,300 VDC / 1,200 ADC IGBTs.

The lab is divided into two experimental bays with a substation in an additional equipment room. Figure 1-5 illustrates this floor plan and anticipated load-side lab configurations. All operations are conducted from an outer room, on the safe side of armored glass windows.

The circulating power loop, which allows the lab to operate at 1 MVA, is available in either bay. If the loop is not used, the facility is limited to 250 kVA from the utility feed. There is space to introduce moderate sized electromechanical elements should the need arise.

The drive, transformers, and switchgear are all air cooled. Additional cooling is available from a 3" diameter water line.

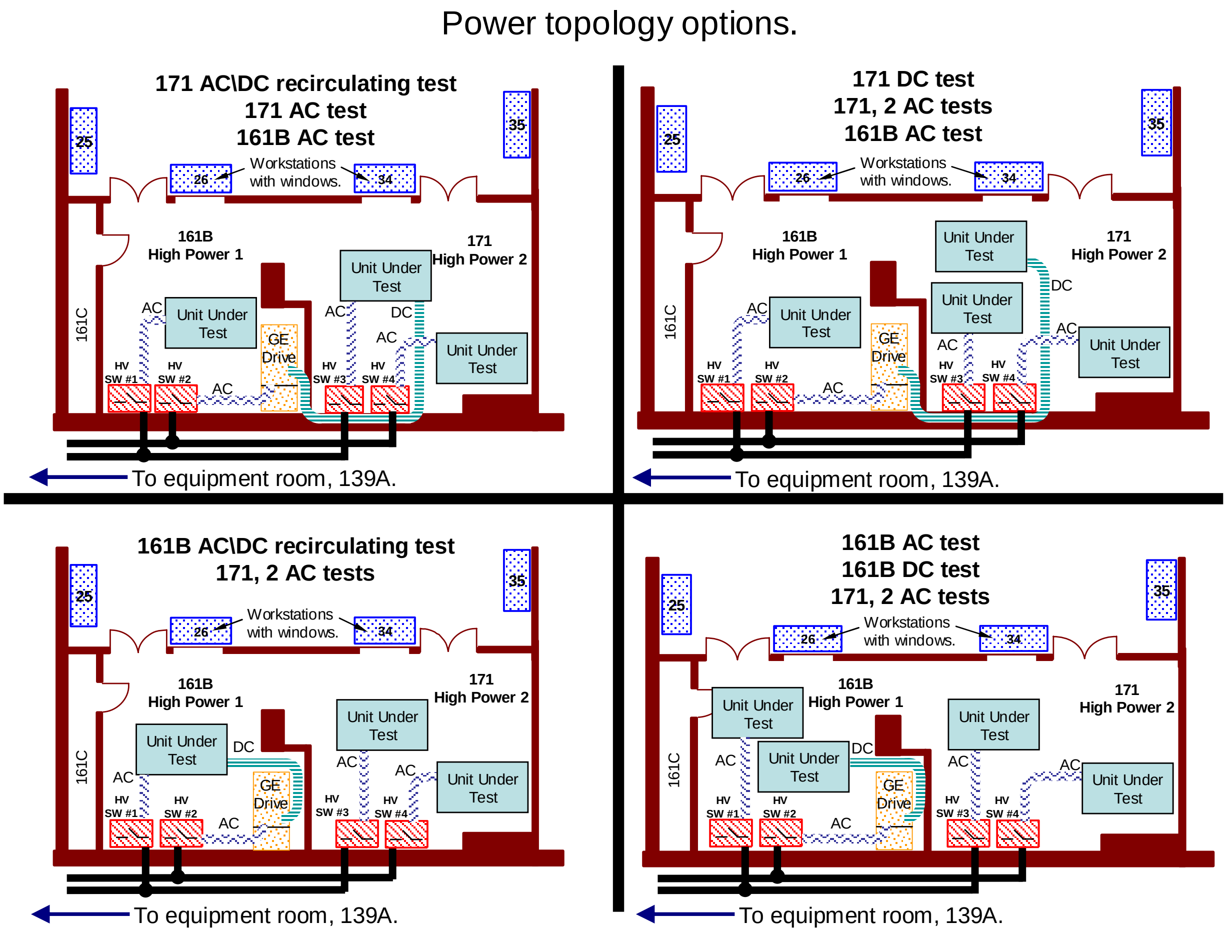


**Figure 1-5; Lab floor plan for various load–side power distribution topologies.**

## 1.5 Objectives of this thesis

A convenient way to plan for a major project such as the high-power lab is to break the specifications and requirements into three groups; capital equipment, physical layout, and control, Figure 1-6. These elements are all interdependent, and it may take several iterations to achieve an integrated design.

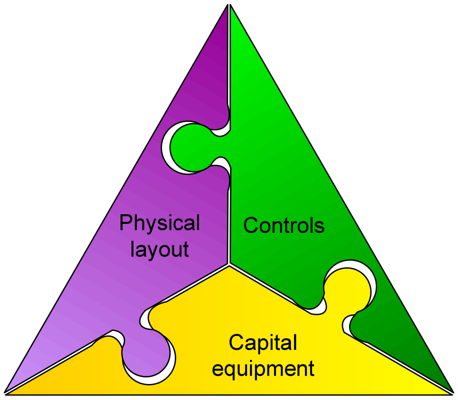


**Figure 1-6; Project elements.**

The selection and placement of  the switchgear, transformers, fuses, reactors, high-voltage cabling and disconnects, and the AC/DC converter is beyond the scope of this thesis (some interesting points of the design discussion are included as Appendix-E).  All that work had been done and the equipment installed prior to this part of the project.  This thesis will document the design and implementation of a control system for the High-Power Lab.

Our objective is first to identify the needs of the users and characterize the capabilities of the existing equipment.  Following this, we must identify technologies and methods which may be brought to bear on the problem and plan a solution.  Detailed designs for networked and discrete hardwired communications, selection and placement of sensors, I/O connection boxes and cabling, graphical user interfaces, and a system security plan are produced.  Finally the system is implemented and all major functions of the control are demonstrated, concluding with an operational plan and user's guide.

The structure of this thesis is as follows: Chapter one has oriented us to the motivations and methods of our lab, and familiarized us with its major hardware and physical layout. Chapter two will delve more into the operational requirements of the lab.  It will cover what we want the control system to do both in terms of enforcing safety protocols and of conducting experiments.   This is also where the philosophy of the hardware selection and the control design used in the lab is explained.

Chapter three will detail the choices made to meet these requirements.  It includes both hardware and software designs.  Chapter four will document verification and testing of the control system elements that have been commissioned to date.  Finally, chapter five summarizes what has worked and what hasn’t and gives guidance toward future work as the facility is adapted to meet emerging needs.  A user's guide is included as Appendix-B.

# Chapter 2 – Control System Requirements and Background

This chapter is in two sections. The first speaks to the specifications for the control system, both in terms of enforcing safety protocols and of operating the facility, and proposes solutions to satisfy them.

The second section is largely background information on the methods, materials, and technologies with which the control system is built. It is included in hope of furthering the broader educational purpose of the high-power lab.

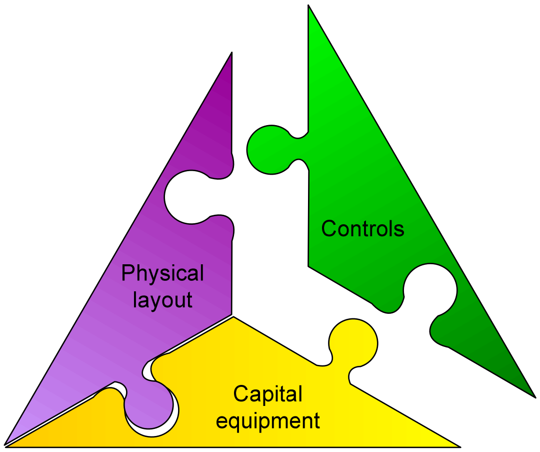


**Figure 2-1, Design scope.**

## *2.1 Lab Operational Requirements*

The High Power Lab equipment is impressively large and potent, beyond the scale of what most people deal with regularly. How then does one construct the lab so it is appropriate for an academic environment? So that it is safe, yet not so restricted as to be unusable? All the equipment needs to be interlocked for safety purposes and integrated in terms of control and data collection, so useful work can be done. As is evident from the physical layout and electrical distribution topology presented in the previous chapter, doing so will require simultaneous operation and monitoring of widely distributed equipment.

From these general considerations, we derive the following list of specifications for the control system. It must:

- Communicate data and commands from remote, dangerous locations to a safe, centralized one.
- Guide the operators in taking correct actions and react automatically to dangerous situations.
- Facilitate data collection.

Additionally, by virtue of being a university facility, the control system will have to be reconfigured periodically to host new experimental equipment. The system must be flexible in that:

- It must be scalable and modular, able to accommodate new needs.
- It must depend as little as possible on components that will become obsolete and hard to maintain.
- Interfaces must be "open" and intuitive to minimize the user learning curve.

System security is also an issue:

- Supervisory constraints and system interlocks must be reasonably robust and readily verifiable.
- Communications must be private and secure.

## 2.2 Proposed solutions.

To address the above specifications, the author proposes the following:

- Use a system of distributed I/O devices, connected by a digital data network for the bulk of control and data communications. Safety-critical signals will be carried by discrete, hardwired circuits.
- Logical elements in the system will continuously evaluate the state of the system and permit or prohibit the power distribution equipment from feeding power to the load. Animated, graphical status displays will indicate the state of the facility, and illustrate the necessary conditions to operate the power system.

- A system monitor will periodically poll the system internal states and log them. It will be possible to trigger external data collection devices in a coordinated way.
- Use a commercially available system of small, inexpensive, network to I/O transducer modules designed to be mounted near to their loads. The individual transducers can be added or replaced as required. Design enclosures with spare capacity for future needs, and place them to minimize wire runs to I/O devices.
- Wherever possible, "name brand" equipment will be used so that the OEMs are more likely to offer future product support. Preference will be given to products that are not tied to a particular generation of hardware or operating system and that have a large installed base.
- Selection will also favor tools and protocols that are well documented, have technical support available, and are industry standards (if not open-source).
- "Safety" and "user" functions will be segregated so that the users can not compromise the safety functions. This may require physical separation of the I/O and control platforms, lockable circuit enclosures, and reference code templates. Potentially dangerous actions will require a "man in the loop."
- The system network(s) will either be isolated or reside behind a router/firewall.

Additionally, the lab is expected to be a high-EMI environment so all circuit enclosures and wiring will have to be shielded. This also leads us to favor I/O devices which do not rely on field effects.

## 2.3 SCADA, what it means, and why it satisfies our needs.

The implementation of many of these ideas is commonly referred to as a “Supervisory Control And Data Acquisition," SCADA, system. This is an umbrella term for integrated instrumentation, operator interface, and automation systems. This is a core element of our proposed solution, so at this point we give a brief explanation of how such a system works. In the following sections we examine in detail the elements of a SCADA system and the design choices that were made in this case.

The concepts certainly are not new; since the advent of the industrial revolution, people have placed recording devices on instruments and connected signaling and control devices from remote equipment to central control stations.[9] This term then describes a mature set of philosophies for doing the things we need in a facility like the high power lab.

The “Supervisory” aspect of such a system means the system designers set operational limits on the equipment which may not be exceeded by the operators. The users/operators may also set their own limits on the equipment for the system to automatically enforce.

A good example of "supervision" would be a pair of limits on a time-power heating curve[10] model used to govern the operation of a machine. The higher, less restrictive limit could be set by lab administration to protect the distribution equipment. The lower, tighter limit could be set by a user to protect some experimental device, perhaps by reducing a reference (Figure 2-2).

This layer of supervision should not to be confused with the system fail-safe protections. It is only a regulation, a software enforced limitation on operation, that allows the system to continue running. The fail-safe protections, like the E-stop buttons, always force the system toward a safe shutdown condition. These concepts will be explored in more detail in section 2.5, Redundancy and fail-safe.

[9] One of the very early applications of a computer to a SCADA system was the automation of the Norfolk and Southern railway switching yard at Roanoke, Virginia, by the General Electric Company. A tiny computer with only a few kilobytes of memory maintained records of the position of cars and switches throughout the yard. *Recollections of instructor David Ryan, General Electric Corporate Training Center, Salem, Virginia during a lecture on the DC2000 motor drive control system, August, 1993.*

[10] A time-power heating curve is a simple model of the thermal mass in a machine, integrating power lost in the machine, minus cooling, over time to yield current temperature. It tends to take a saw-tooth form, with a fast rise and slow decay.

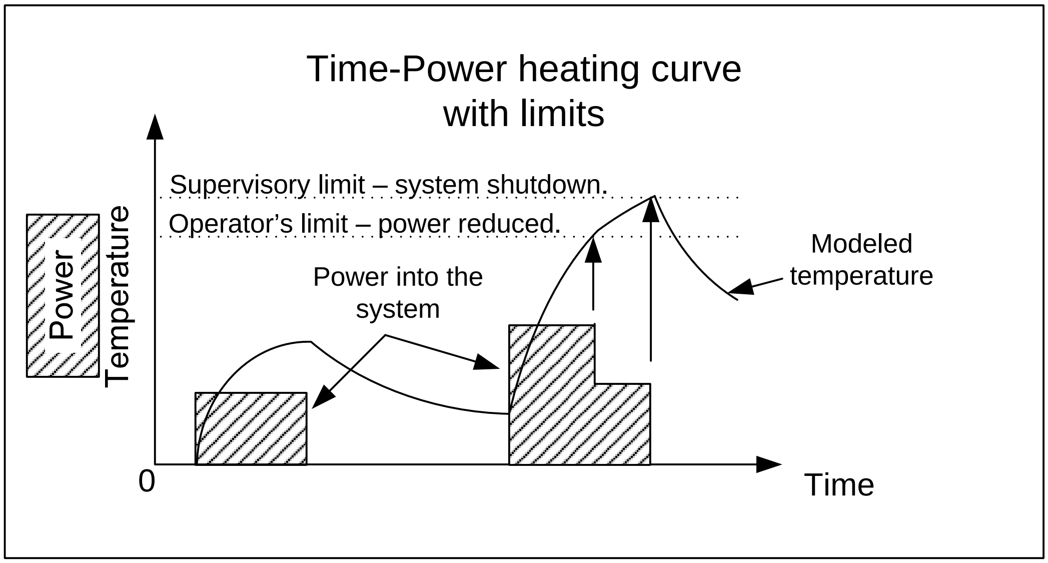


**Figure 2-2; Example of a time-power heating curve used for control.**

In terms of "Control," the SCADA system is the operators' primary means of interaction with the equipment. The system provides a display, typically on a PC monitor, of plant status feedbacks and measurements, and allows control of operational set-points for all the various pieces of equipment. The information on display at an operator station is collected by sensors scattered throughout the installation, and likewise operator control inputs are distributed to the remote equipment. These remote devices can be *miles* from the control station. Operator interface design principles will be discussed in depth in section 2.4.2.

The domain of system automation spans both supervisory and control functions. By analogy the supervisory tasks could be termed "autonomic," and the other functions, "conscious." The prior example of the system heating limit is autonomic. This is a built-in protection, set up by the system administrator and "standing guard." True, the operator may *tune* the function by setting a limit, but in *extremis* the function acts without operator intervention.

Alternately, "conscious" controls would include user-initiated actions. These could be pushing a control button, setting a numerical reference for a regulator, or running a user-authored program to take a sequence of such actions. Common examples would be a schedule of references in support of performance and transient testing, simulation of faults, injection or suppression noise, etc.

There will, of course, be a need to measure the results of all this. At present, “Data Acquisition” in the labs is handled on a small, local scale by individual instruments (scopes, meters, etc.). A coordinated, integrated "Data Acquisition" system will allow data to be collected across the facility both by permanent instruments and by ones added on for a particular project.

Some data will be collected continuously, other only when triggered. Ideally, everything will be assimilated into a coherent collection to allow for a holistic analysis. A major use for this sort of data collection is forensic fault analysis. Often finding the root cause of a fault involves noting some *supposedly* unrelated event that occurred (such as a reference change to another piece of equipment) just before the device in question changed state in some way. For this to work, the collection needs to be long term, continuous, broad in scope, and reasonably frequent.

## 2.4 Elements of a SCADA system

Physically implementing a SCADA system requires sensors and actuators, an operator interface, communications, logic, and data processing functions.

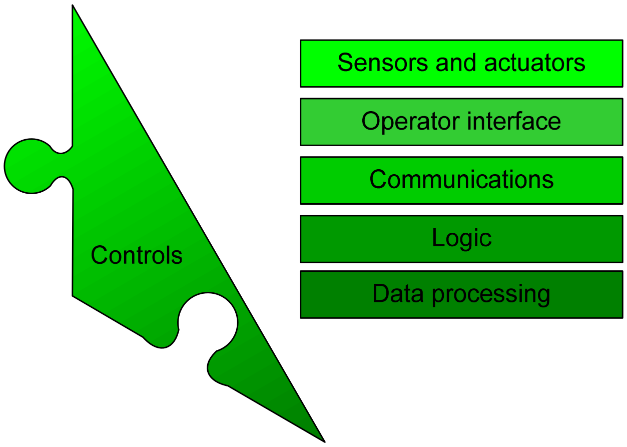


**Figure 2-3; Elements of a SCADA system.**

### 2.4.1 Sensors and actuators

A control system of any description is of little use without good feedback. Achieving this requires proper selection and placement of devices with due concern for their environment. One must consider the range of temperature, shock and vibration, voltage, radiated EMI, contamination, and other abuses a sensor will be exposed to.

Often a sensor can give data about its health *in addition to* what it is supposed to be measuring. For example, an analog transducer may return a 4 to 20 mA signal to encode for a range of detected conditions (i.e., levels, temperatures, positions, etc.). If such a signal ever reads outside the 4-20 mA range then the wiring may be broken, shorted, or the sensor itself damaged.

Discrete valued, solid-state sensors can also be interrogated for health. They usually have a minimum leakage current even in the "off" state and a maximum current in the "on" state. These characteristics again form bounds to certify that the sensor is *probably* functional.

Making these health checks can be quite a problem, requiring sensors to check the sensors that watch the sensors *ad infinitum*. Commercial I/O systems typically have an interface module that monitors whatever field device is attached to it and reports data and status to the next higher level of control. That level then checks whether the interface module is acting as *it* should. This secondary status is added to the package, and the whole lot is reported further upstream, a process that repeats with each device that handles the data until it is acted on.

Often the *quality* of the measurement is just as important as the data returned. If a signal related to some important state is "unhealthy," then it is necessary to assume the worst from a control perspective. The control system must compensate for this "blind spot" by making decisions based on other data, operating in a modified mode so that the missing data is not crucial, or shutting down entirely until things are fixed (never a popular option!). Obviously, some things are not worth the trouble to check – it depends on what is at stake. For safety-critical signals this entire structure is eschewed in favor of hardwired mechanical interlocks because their state transitions are more simple and predictable.

Note that just because a particular sensor is healthy, that is no guarantee it is still physically in position to take a meaningful measurement. Whenever possible, crucial states should be monitored by three independent sensors. From the multiple inputs, a robust state estimation can be achieved. Often it is better if the sensors are measuring different kinds of quantities, so they are not all fooled by a single phenomenon. For example, a valve position reading may corroborate a temperature to out-vote an alarm from an erroneous pressure reading, which may be due to vibration.

There is always a temptation to choose a "fancy" sensor, but often a simple (and cheap) one will do the job. For example, when measuring ambient temperature or coolant flow is an analog value really necessary, or will a discrete threshold detection suffice? Sometimes not – motor windings and bearings are typically fitted with analog thermocouples so that rate-of-rise can be monitored. Often multiple, discrete sensors can be used to provide a distributed, robust sampling which may be a better design.

Environmental factors are just as important in actuator selection as they are in choosing sensors. The control system must also monitor actuator health. Certain types (solenoids, motors) can be monitored for short/open circuit conditions. Others, such as servo-actuated valves (which regulate a position), necessarily incorporate sensors in their control mechanism. This sensor can be used to verify the output is probably working. Another example is typical of motor control centers; the large power relay that connects a motor to a source through a main or "M" contact often has a mechanically integrated, but electrically isolated auxiliary, "AX" (or just "X"), contact which signals the state of the main contact back to the control system.

This geometric increase in I/O complexity is one of the reasons mechanical linkages were not used in the high-power lab. Presently the lab's only moving (output) parts are cooling fans and a few signal-level relays[11]. The cooling fans, incidentally, have auxiliary motor contacts to indicate they are operating[12].

There are other considerations for I/O devices in the High-power lab:

- The interface components must be modular so that failed or damaged components can be replaced easily.
- A variety of I/O loads must be supported, including analog, PWM/high-speed, high voltage isolation.
- The system must be expandable to accommodate future needs.
- Wiring connections should be kept short, especially in light of our expected high-EMI conditions, so the I/O system should be distributed.
- Devices themselves should be EMI tolerant. In other words, simple and metal bodied. Contact-free proximity sensors probably won't work.
- Cost.
- Some sensors will have to be shock/vibration tolerant.

Generally, High-power lab devices are not required to be explosion-proof[13], nor concerned with dust, moisture, temperature extremes, caustics, or radiation.

Some care is required in designing mounting enclosures and cabling:

- Everything should be shielded.
- Cable runs need to be easily identified.
- Anything related to safety interlocks needs to be physically secure. Enclosure boxes need to be lockable.

[11] Dr. Porche, when designing the Volkswagen Beetle, was quoted as saying, "A part that does not exist can not fail," to explain his draconian simplification of the design. We hope he would approve of the high-power lab.

[12] This is actually a calculated risk on the part of the drive manufacturer. What they really know is that the motor has been selected on, not that it is actually running!

[13] "Explosion-proof" indicates that a device will not be a source of ignition in volatile environments.

### 2.4.2 Operator interface

Operator stations are where one finds all the buttons, switches, dials, gauges and indicator lights required to run the equipment. Modern ones include CCTV and computer terminals, usually with a graphical control interface. Things are always evolving in this area, sometimes due to changes in common skills and preferences among users. Other changes are driven by research in human factors engineering and the ergonomics of operator interface design. At this point we will present some practical, useable principles for setting up a good operator interface without delving into the physiology and psychology more than necessary.

There will always be call for some discrete devices – the big, red E-stop button for one thing, fire alarms for another. Discrete devices have some advantages such as fixed location and function. Sometimes they are easier to use (as when wearing gloves), offer better performance (video gamers use joysticks for a reason), or have security features such as key locks.

Discrete devices do have drawbacks, though. Operator station I/O has the same health-checking requirements as field I/O. Also, compared to graphical controls, discrete devices are more expensive[14], hard to reconfigure, require more power, and get covered up[15] or pushed accidentally[16].

Graphical screens have the advantage of being infinitely configurable. The same indication can be presented in many different ways, regrouped with others to suit the current situation, or even the operator's taste. Changes can be made very quickly[17]. In modern systems

[14] An industrial quality illuminated pushbutton can cost $30 or more. A five-position joystick can be hundreds of dollars. Interface hardware, mountings, and wiring add many hundreds more.

[15] The famous accident at the Three Mile Island power station was exacerbated by the fact that the operator failed to observe some of the critical warning indicators. Reportedly, he was so fat that he couldn't see them over his belly. *Recollections of accident report sent to other nuclear stations worldwide by General Electric engineer Christian Duplaa, who was working at a French nuclear power facility at the time.*

[16] The Apollo 16 lunar landing was almost scrubbed because of a mechanical fault in the abort switch, which caused it to falsely indicate that it had been pushed.

[17] In the minutes after the attack on The Pentagon on 9/11/2001, engineers working on the new environmental control system were able to reconfigure an undamaged control station to shut down air handlers in the part of the building that was burning, thereby preventing the fire from spreading. The primary controls had been destroyed. Smoonian, D., "Smart buildings, Saving the Pentagon," IEEE Spectrum Magazine, IEEE, On page(s): 18-23, Volume: 40, Issue: 8, Aug. 2003

then, every control without some special requirement to be discrete tends to become a screen graphic.

There are three major things to consider when designing a graphical control station:

- Graphics must be designed for appropriate density and clarity of information. Don't add distractions or fluff. Emphasize the important and limit opportunities for misinterpretation.
- Organization of controls and navigation between groupings. Make the *right* things available.
- Alarms and other messaging. Make these punctual, meaningful, and avoid overloading the user.

### 2.4.2.1 Graphic design

Good graphics design will always put the most important information forward in a consistent, predictable manner. The graphics require a certain degree of abstraction, but the display should also mimic the appearance and layout of the real equipment. The relationship between them should be quite clear and intuitive.

On a given screen many of the objects will be active. Some will depict control "buttons" which will take operational actions directly (start/stop, select a mode, etc.). Other objects will set a numerical value. These objects can look like anything - pointers to be dragged along a slider bar, knobs, switches. Often the choice of "input device" depends on required precision and speed to make a change, which in turn depends on the equipment and process involved. Sometimes it is better to click and hold a vernier adjust button while looking at something else, other times this is too slow and typing in a number is better. Having both methods available, and others besides, is both possible and reasonable.

Status display objects will use text, numbers, color, and position or animation to distinguish various states and bring especially important things to the operator's attention. Changes in appearance should mean something definitive, not be interpolations adrift between known states. A word of caution, neophyte screen designers tend to clutter screens with distractions. This is *not* a video game.

Colors and behavior of screen objects also need to follow a consistent schema. In a classic case of dueling standards, sometimes "Stop" buttons are red and "Start," green (like traffic signals), and other times they are reversed to green for Stop (safe) and red for Start (unsafe)[18]. Adding to the confusion, in some schemes a control is illuminated when it is available, and in others illuminated to signify when the machine is already in that state. To make matters worse, colorblindness is a serious consideration[19]. Clearly then, since people's associations with colors, lights, etc., can not be taken for granted. There need to be other cues. For example, controls need to be clearly marked with verbiage like, "Push to Stop," and "Now ON."

### 2.4.2.2 Organization and navigation

In the discrete world, you have to physically go to where a control or indicator is located. Oftentimes, this requires jumping back and forth between control stations or coordinating actions between operators who may not know what the other is doing[20].

Things are easier in the graphical control world. Graphical screens potentially have access to, and control of, any state available to the control network. This makes it practical to group controls in support of a given situation or procedure, and regroup them differently on another screen for another purpose. For example, an I/O check-out screen may have a simple listing of sensor states corresponding to items on a check list, while a logic/diagnostic screen would have a subset of the I/O, but add some internal logic states.

Given a system with dozens or hundreds of special purpose screens, finding the one you need can be a challenge. We are all familiar with menu and icon navigation – good operator

[18] Controls on Unites States Navy vessels use one scheme, controls at nuclear power stations have been standardized on the other. The latter is often staffed by people who have been trained on the former, and the potential for error is sobering.

[19] Colorblindness was *discovered* as a result of people misreading traffic signals. This is why signals are now always red on top, green on bottom.

[20] The author once observed as two operators, using discrete, local controls on widely separated pieces of equipment, managed to push a twenty ton coil of steel over a ledge and drop it three stories. It bounced.

screens generally build on this philosophy. Dedicated buttons serve as shortcuts to overviews of major equipment ("DC source" for example), or functional groupings ("cooling").

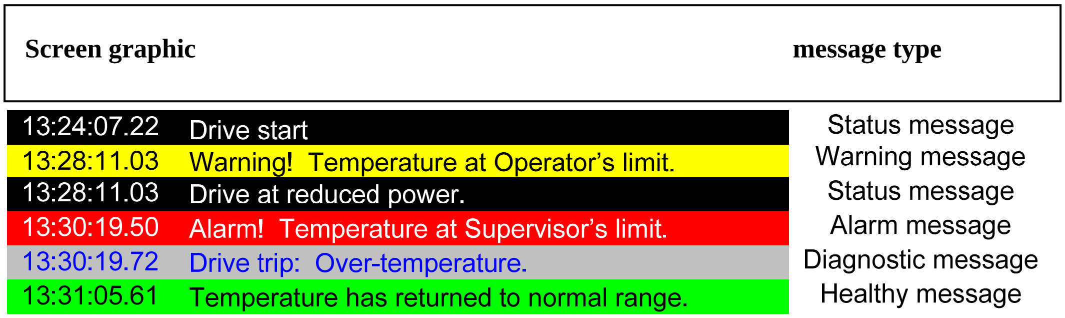
The depictions of devices themselves may also serve as icons, which lead to successively more detailed sub-screens dealing with a particular system (you want to know more about the status of something? Then click on its picture). Screen objects may also open special purpose sub-windows (help, diagnostics), or navigate to another set of controls entirely. This makes the navigation more intuitive and allows for ample alternate paths to retrieve desired information.

It is important to consider the balance between using pop-up windows, which by necessity obscure part of the parent screen and jumping to an entirely different screen you then have to navigate back from. Again, it bears considering the task at hand.

### 2.4.2.3 Messages and Alarms

The system logic can be configured to recognize important events and log them for later analysis. Some of these events are critical, enough so that the operator needs to be alerted to them even if they are not related to things represented in the current screen view. To accommodate this, graphic displays commonly reserve an area of the screen for displaying messages.

Status messages may be classified and color-coded according to the type and severity of information presented. A message window will accommodate a small number of such messages – usually the most recent couple in each category, with older ones scrolling away. The previous time/power heating example, Figure 2-2, may generate a sequence of messages like the following in Figure 2-4.



| Screen graphic | | message type |
|---|---|---|
| 13:24:07.22 | Drive start | Status message |
| 13:28:11.03 | Warning!  Temperature at Operator's limit. | Warning message |
| 13:28:11.03 | Drive at reduced power. | Status message |
| 13:30:19.50 | Alarm!  Temperature at Supervisor's limit. | Alarm message |
| 13:30:19.72 | Drive trip:  Over-temperature. | Diagnostic message |
| 13:31:05.61 | Temperature has returned to normal range. | Healthy message |

**Figure 2-4; Example of a sequence of status and alarm messages.**

These messages typically have variable fields which are populated at the time the message is created, something like, "Device X reports Y."  Sometimes these messages are also linked to diagnostics or help documents that suggest corrective actions.

### 2.4.3 Communications

The core concept of SCADA systems is the coordination of equipment.  Clearly, there can be no coordination without communications between devices.  The control system depends on good flow of data.  The definition of "good" depends largely on what is being communicated.  For some signals, a dedicated, hardwired connection makes sense.  Others signals are well suited to network communications in one form or another.  The most critical require redundant channels and error correction logic.  In the following, we examine the various types of connections typically found in a system.

#### 2.4.3.1 Direct wiring.

Signals with exceptional bandwidth requirements are directly wired between source and user.  An oscilloscope probe is a fine example of this, as is a pulse-tachometer feeding motor position data back to a drive.  Burdening these signals with a network overhead would severely degrade them and limit their utility.

In other cases, critical signals (such as E-Stop) are hardwired in order to increase their reliability. The philosophy is "simple is better." Electronics, especially computers, should be avoided in the chain of communications because they can fail in unpredictable ways[21, 22].

In "the olden days," direct wiring was the most common way to make logical connections between equipment. To signal between devices A and B, A would energize a relay coil to close an isolated set of contacts, signaling an output. This would close a circuit (probably at 110 VAC) wired from device B. Often this would energize another relay in B which would in turn close a low voltage circuit to drive an internal logical indication.

This sort of connection is the lowest common denominator, default way to signal between devices. Sometimes it makes sense for very simple connections, but is counter productive from a centralized control perspective – these signals are not generally available for operator display or logging. There are better ways to make logical connections which require less space, power, and are generally cheaper.

#### 2.4.3.2 Networking – the modern marvel.

Networked communications is one of the greatest advantages modern control system designers have over their forebears. In a modern system, small clusters of I/O devices are connected to network interface points. At this point, some form of interface module adjusts voltage levels, provides isolation, encodes/decodes I/O to logical messages, and acts as a network transceiver to share the data with controller(s) located remotely.

All this is no small task, and it always incurs a time penalty. Even so, it is worthwhile because a network connection can replace *hundreds* of wires with just a single pair. If they have to cover any great distance then the savings in the cost of the wire (and the cableways, and the

[21] There are no computers involved in the operation of nuclear power plants. Period. *All* signals are considered to be safety critical.
Lecture and tour of the North Anna Nuclear Power Station by engineer Douglas Struckmeyer, 1990.

[22] The switching system for the New York City subway still relies on electromechanical relays because they open by gravity under all fault conditions.
Postman, Monroe. "Robust Relic." Letter. IEEE Spectrum Jul. 2005: Vol. 42, Number 7.

installation labor) quickly pays for the networking equipment. Furthermore, if the connections must pass through a confined area, a network connection may be the only practical method.

By virtue of being digital signals, network communications can benefit from error correction techniques to achieve better fidelity than analog signals in the same channel. Indeed, sometimes a digital signal over a fiber-optic link is the only practical way to establish communications at all. Lastly, networked systems can be much more energy efficient both in the communications themselves and in power distribution to the I/O loads.

Choosing what type of network to use can be challenging, especially when multiple platforms and vendors are involved. Generally speaking, the first step in forging components of various provenance into a coherent system is to get as many of them as possible to communicate via some form of network.

#### *2.4.3.2.1 Serial networks*

Dedicated, private, serial networks are very common for applications with a small, fixed number of devices. Most test equipment, for example, typically has a PC-style serial connection for data transfer. There are many more varieties of serial networks: ModBus, ProfiBus, Genius, hp-GPIB, CAN bus, USB and others, which accommodate more devices. Some are short-haul only, others can span hundreds of feet.

In general these connections work well. Some are proprietary and very expensive. Others were devised to operate on now-obsolete hardware, making them slow or otherwise limited. The biggest problem tends to be shortage of connection ports. Configuration can also be troublesome and require a bit of knowledgeable tweaking.

#### *2.4.3.2.2 Ethernet*

Ethernet has become a sort of *lingua franca* - most modern instrumentation and control equipment either connects directly by Ethernet or can be adapted to do so by a bridge device. Ethernet supports huge bandwidth and long distance communications with cheap, off-the-shelf hardware from a multitude of vendors. It is also very easy to make connections.

As discussed previously, most physical I/O devices need some sort of interface module. Small, collocated groupings of driver modules, all within a single I/O cabinet for example, will be networked serially to an Ethernet interface (bridge) module. This in turn provides an Ethernet connection to the operator interface, data logging, and logical elements.

If there is a drawback to using Ethernet for control, it is that the format has become too common. Everyone wants to plug in a PC to the control network, check their email, and stream in music. This is a drain on network resources and a security risk, and these activities should not be allowed.

### 2.4.4 Logic

The SCADA system must have logical elements – be they clockwork, relays, or computers, which oversee the plant and take supervisory control actions. Some of this function is delegated to "off the shelf" devices such as circuit breakers and self-regulating valves, with limited programmability suited to their very narrow scope. A circuit breaker, for example, may have adjustable trip settings, but will not care about the state of other equipment. "Programming" it may only require a screwdriver.

Of more interest here are the general purpose logical elements, those which assimilate data from diverse inputs and may take a wide range of actions. When choosing equipment and software, issues of migration and obsolescence, coding languages, and expandability\adaptability for future needs all require consideration. It is important to avoid implementing something for which there is no technical support, no spare parts, a large learning curve, and no way to update it.

The standard answer for implementing a control system is the programmable logic controller, or PLC. These machines are optimized as controllers in many ways; the "bookkeeping" tasks of managing memory, I/O handling, interfacing to (multiple) networks, timing, etc. are already done. PLCs are also typically closely integrated to the I/O they need, so there is a minimum of delay in the system.

The PLC environment leaves the programmer free to code the function they care about without worrying they are going to "break" something else with a coding error (equipment maybe, but the controller will not crash). Often coding can be done at a very high level, such as with a graphical control-block language (… drag your mouse to connect the "regulator" block between the "sensor" and "actuator" blocks …). This makes them easy to use and fast to set up[23].

PLC vendors also supply tool suites for trending, data logging, and other necessities. These tools have become so powerful that sometimes they can replace the PLC itself - the tools become a "soft-PLC."

A soft-PLC has decided advantages over a hardware one. If the computer the PLC software runs on breaks, or simply becomes obsolete, it is cheap and easy to replace. This is an important consideration because industrial customers often find themselves paying huge premiums for a "last of its kind" piece of control hardware[24]. Also, any tools to configure and troubleshoot the control will not be marooned on an obsolete computer\OS kept solely to support the controller.

#### 2.4.4.1 The Role of the Control System vs. the Role of the Operator.

There is a temptation with any supervisory system to make it the responsible party in any possible fault situation. "It shouldn't let the operator do that," is a common refrain. This sounds fair enough, but remember, you can make the control bullet-proof, but not operator-proof. The control system can observe, advise, and impose some degree of order, but if people behave recklessly, *especially* by defeating safety interlocks, then nothing is going to stop them from causing a disaster.

[23] It can be, quite literally, child's play. LEGO produces a simple PLC for automating toy robots.

[24] There is actually a conventional hardware PLC add-on board integrated into the lab's GE drive. We are not using it because if it breaks we may never find a replacement.

It is also important to remember that computers are fallible. Cosmic rays cause random changes to memory[25] and\or destroy components[26]. This can cause things to "turn on" all by themselves. A computer must never be trusted to protect life and limb. This must be emphasized in lock-out/tag-out safety training for anyone working with *any* equipment. *Always* put physical protections (i.e., grounding) in place before working on equipment connected to the utility or that could store energy. Remember, at some point a rodent will A) chew the insulation off critical wiring or B) decide to urinate on the high-voltage bus bar. In either case the control logic will not save you.

#### 2.4.4.2 The Administrative Role.

The role of the lab administrator is to make the control operator-proof. This means maintaining physical security and supervision of anything that is, or could be, energized. The administrator will have to enforce lock out\tag out procedures and supervise pre-energization checks.

From a software perspective, the administrative code will be that which sets the outer limits of the lab's use and behavior. Users may write software as they require, but it will never be able to supersede the administrative limits. User code will issue "requests" for state changes, the administrative code will vet these and issue "commands" to the actual hardware.

### 2.4.5 Data processing

Data processing in this context is the collecting, archiving, and presentation of the status of the equipment over time. The time span can vary from a few milliseconds during a transient test, to days or weeks for statistical trend analysis. One of the bigger challenges is to collect

[25] During one modest solar storm soon after the VT supercomputer came on line, approximately 11 such events happened in its main memory.

[26] The type of IGBTs used in the GE drive have been seen to fail spectacularly due to the impact of an energetic neutron.
Borovina, D. L. et al. "Neutron-Induced Failure Tests of 3300-V IGBTs for the Spallation Neutron Source Accelerator." Proceedings of the IEEE Particle Accelerator Conference, 2003.

enough of the right kind of data so that unexpected relationships can be sussed out after the fact. This is, again, an advantage of having all possible equipment communicating on a common network; place a data recorder on the net to eavesdrop on and store whatever comes by.

Part of making use of this data is knowing when things *happened* as opposed to when they were *recorded*. All the elements in the system will typically be referenced to a stable, common, local clock. This in turn will be synced to a high-quality time master, such as an atomic clock, which is publicly available on the web. This will allow meaningful time stamping of events. A further understanding of latency can be gained by recording reference events through multiple channels and comparing when they appear to occur.

An example of this "temporal interferometry" would be using a marker pulse on a motor's tachometer to trigger both an oscilloscope recording external data and an internal recording of the motor drive's states, which then get uploaded to the common data recorder. Seeing the same event from two perspectives allows one to put all the other data collected from the two sources into context.

However, the two recordings in this example are likely to be fairly dense and of different formats. Getting them into the common archive probably will not happen in real-time. Doing so at all may require manual processing. A more reasonable goal would be for an outside recorder to simply grab the data and store it in a way that reliably correlates the two files with low-bandwidth, real-time data from other sources. This could involve yet a third recording of the marker pulse via a discrete input module.

The system data recorder will collect a broad, but modest resolution view of the plant to add context to specially triggered, high resolution recordings of interesting events. With good planning, the conditions surrounding an experiment or a fault condition can be reconstructed accurately.

To analyze these events, data recordings must be searchable at will ("show me variables X, Y, and Z between then and now"). Once the interesting events are isolated, it must be possible to do basic frequency domain and statistical analysis and to export the raw data for processing in other packages (MATLAB does not do well on bitmap images, it needs numbers).

Operators, of course, deal in real-time data. They may want a short-term trend display, but the real concern is what is happening right now. With this in mind, most data acquisition and analysis functions are not going to be prominent on the operator interface screens. They may not even be accessible directly – why add the distraction?

## *2.5 Redundancy and fail-safe*

This topic warrants its own section even though we have touched on it in other places. The design of a control system has to account for the things that will inevitably go wrong. When this happens, the control system must do whatever is possible to keep people safe. This means detecting a fault and then de-energizing the system in a safe, orderly way.

### 2.5.1 Fault detection – the *fail-safe* principle

In some cases, there are conditions that are *so* critical to safe operations, they merit inclusion in a hard-wired "fault-string." This is a serial connection of indications that must be maintained "true" for the equipment in question to operate. Primary among these are the Emergency-Stop switches.

The operative attitude in fault detection is, "Guilty until proven innocent. Sometimes twice." Practically speaking, if there is a state that must be maintained to ensure safety, then that state must be monitored in such a way that a complete circuit has to be maintained in order to signal a "safe" condition.

This means that protective sensors will be configured to open to signify a condition that is not intrinsically safe (temperature too high, lubricant or coolant level too low, door open, etc.). Thus, a broken wire will not go undetected, but will be interpreted as a shut-down command.

In the very most critical cases, redundant elements may be used (think of having multiple smoke detectors in your house). To be really, really careful redundant sensors should share no common elements – not even power supply or cableways.

Commonly, sensors will have complimentary outputs. The "normally-closed" contact will be used for the fault detection, and the "normally-open" contact can be wired to a networked input point. This second path is not itself fail-safe, but provides supporting evidence for the primary signal. Control logic can now watch for discrepancies. This protects against failure in the sensor's supporting wiring and gives the control system the opportunity to take further action if the initial response has failed.

### 2.5.2 Protective actions

Lacking proof that a safe condition exists, protective equipment ideally needs to act *without a further supply of energy.* If this is not possible then steps need to be taken to ensure that energy is available – installing a battery backup system for example.

Sometimes simply turning something off when there is a problem is not the best approach. Coordinated, controlled shut-downs are usually safer, especially when there is stored energy to be dissipated. Careful design will separate the indications and controls for these situations, which may still be fail-safe and redundant, from the truly last-ditch effort of E-stop controls.

## *2.6 A case study.*

The following is a hypothetical case study of a motor drive system undergoing a series of faults. This will illustrate how the control system reacts and where redundant or fail-safe elements help.

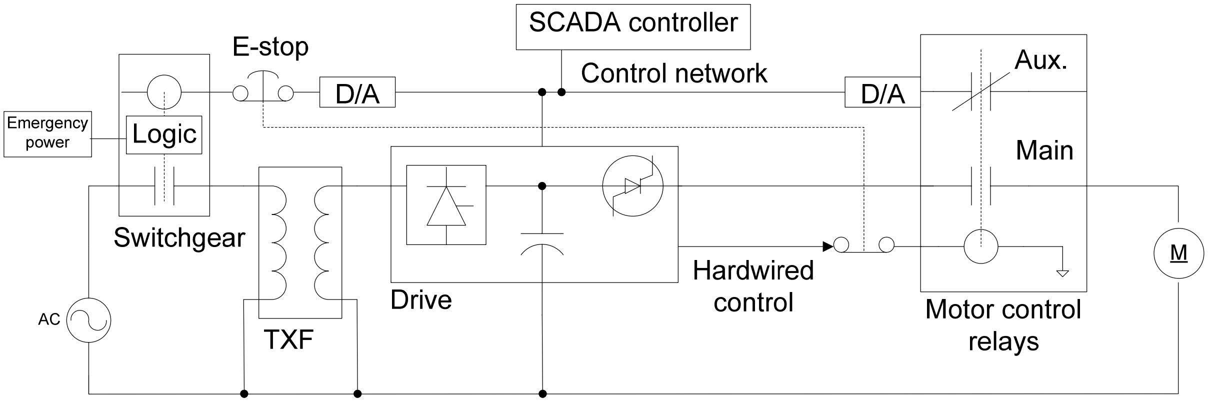


**Figure 2-5; Typical SCADA  for motor control.**

A drive feeding a motor through a motor control relay suffers a shorted IGBT in the output stage.  The motor current increases beyond limits, so the drive's logic removes the hardwired "close" command to the motor relay.  The motor relay "close" command energizes a solenoid that pushes the contacts together.  With power removed, an opposing spring opens the contacts.  Hence, this relay is a fail-safe device.

The main (motor) contact does open after some hundreds of milliseconds, and its auxiliary (instrumentation) contact closes simultaneously.   This auxiliary contact is wired to a control I/O point, informing the control system of the motor relay's state.  This is a redundant, fail-safe status signal.

The motor relay is undersized and unable to interrupt the full short-circuit current.  When it opens, an arc is formed, maintaining a closed circuit.  This happens to some degree even in properly sized equipment, eroding contact surfaces before the arc is snuffed.  Consequently, regular inspection and maintenance of power contactors is crucial.

At this point, the SCADA logic could note that the drive is still reporting a non-zero output current over the control network, yet the motor auxiliary contact is closed.  This is not normal.   The control could then decide to signal the switchgear to open the utility feed breaker by removing the "energize" command.  Without the feedback from the motor auxiliary contact, the SCADA would have to wait for some other fault indication from the drive. Such an indication may come first anyway – another good case of redundancy.

The switchgear logic waits for the AC current to be near zero (to minimize arcing) and then opens the breaker. This is not a fail-safe operation because external energy is required to open the breaker and power the logic. The situation is improved by use of a backup emergency power supply, but the equipment can still fail to open. Should the switchgear fail, eventually a line fuse or other protective device will have to open.

Approximately a second after the initial problem develops, the startled operator manages to hit the E-stop button. This mechanically opens circuits the control has already deenergized. This action accomplishes the same thing as the automation (albeit later and with potentially more damage to equipment). This aspect of the system is also fail-safe. A broken wire, or failure of the control network, would also lead to an E-stop action in the switchgear and motor control relay.

## 2.7 Summary of design goals

The controls for the high-power lab will be akin to those found in modern industrial settings. The I/O system will be distributed about the lab space for ease of access with minimal wire runs and support sufficient instrumentation to observe the critical system states. Wiring and other construction practices will accommodate the expected high EMI environment and be fail-safe. The operator interface will conform to known best practice ergonomics and provide diagnostic and troubleshooting aids.

# Chapter 3 - Design and Implementation

This chapter deals with the implementation of the control system for the High-Power lab. We will detail the design choices made for this application and illustrate how they satisfy both the operational requirements and the design philosophies outlined in chapter 2.

In some cases, the design effort was an exercise in "hacking," in taking existing things and making them work together by any means necessary. This is a practical reality and a warning - revamp installations are always difficult. One manifestation of this in the VT lab is that all the disconnect knife switches are *inside* the lab bays, instead of in a safe area. This means the potential will always exist to close an energized switch while in proximity to the experimental equipment. Secondly, the switchgear and circuit breakers are *not fail-safe* (and cannot be modified to be so) in that they A) do not require an external signal to remain closed and, B) require external control power to open. Lastly, there is no way to lock-out the 480VAC breakers. We have addressed these issues as best we can in the planning of the software, instrumentation, and procedures.

Since the design requirements are formulated as high level statements, a top-down approach seems appropriate for explaining our efforts. We begin then with the design of automation. Later sections will deal with planning of instrumentation and communications/network topology to get necessary information to the automation system. Lastly, we deal with design of operating procedures.

## *3.1 Automation design*

The very highest edict of the specification is that the lab is to be safe to work in. The first step is to decide what this means. Our solution is to, with a high degree of abstraction, describe the lab's overall condition in terms of a finite state machine with three basic states:

- <u>Operating</u>, when the power is on and the system is operating normally.
- <u>Fault</u>, any time there is a condition that should inhibit energizing the system.
- <u>Safe</u>, when the power is reported by the control system to be physically disconnected, and the controls are operating normally.

This state machine determines whether the lab utility connections may be (or remain) closed, or if the system automation must open (trip) them.

The inputs which drive the transitions between system Safe, Operating, and Fault states are; Power (On/Off), Room (Vacant/Occupied), and Control (Healthy/Fault). Since these are themselves abstractions, the resultants of many low-level inputs and other factors, we term them "mid-level" states. These are described as follows, with the system state results illustrated in figure 3-1 and associated ladder logic code in Appendix-A.

For "Lab Occupied":

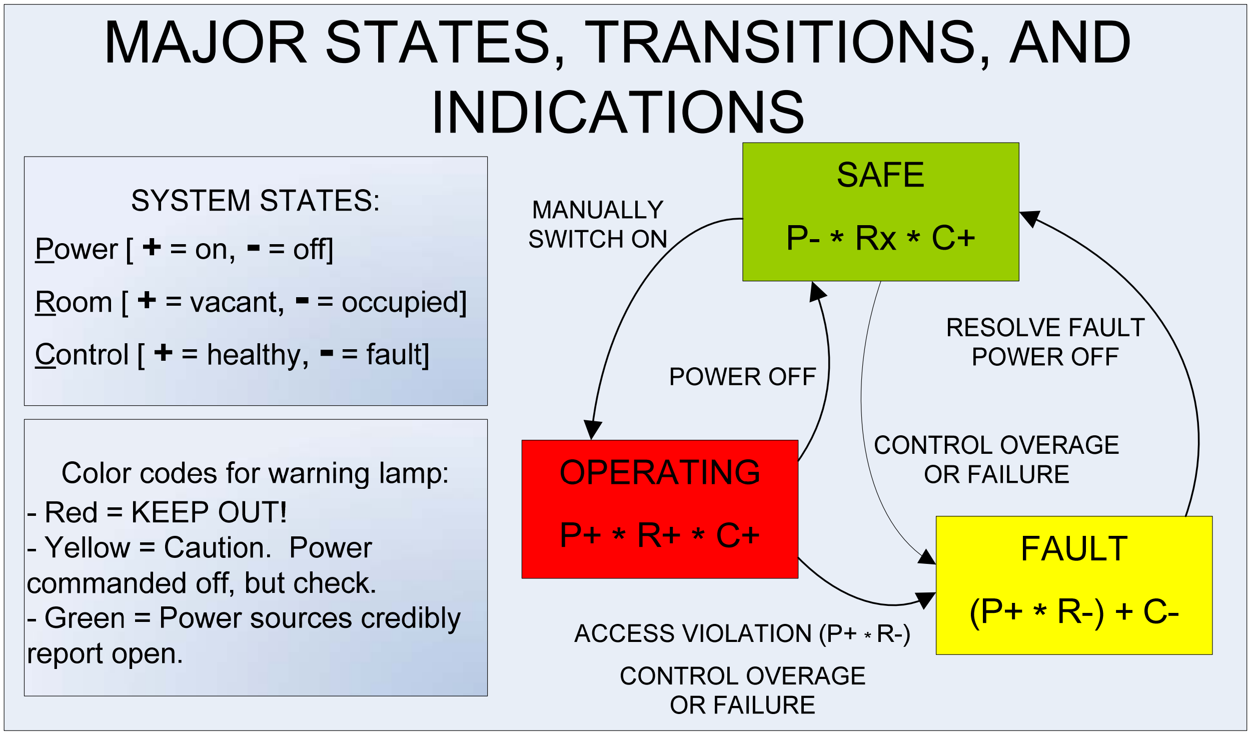
If the door is open, the lab space is considered to be occupied. Additionally, any transition of a sensor inside the lab space which could reasonably be the result of a manual operation latches the occupied state "true," until a rising edge of the door closed indication occurs.

For "Power-On":

The power is assumed to be "On" unless there is a positive indication from a disconnect device for each circuit feeding the lab area that every circuit is open. Any equipment in the lab capable of storing a dangerous amount of energy *must* be included in this decision.

For "Control-Healthy":

A positive "Control-Healthy" output must be continuously asserted in order for equipment to be energized. The control is assumed to be unhealthy unless there are good communications with all control elements and those elements all report that satisfactory conditions exist. Thermal faults will include a lag appropriate to the thermal time constant of the equipment and be conditioned by which equipment is in operation.



MAJOR STATES, TRANSITIONS, AND INDICATIONS

SYSTEM STATES:

Power [ + = on, - = off]

Room [ + = vacant, - = occupied]

Control [ + = healthy, - = fault]

Color codes for warning lamp:

- Red = KEEP OUT!
- Yellow = Caution. Power commanded off, but check.
- Green = Power sources credibly report open.

SAFE

P- * Rx * C+

OPERATING

P+ * R+ * C+

FAULT

(P+ * R-) + C-

MANUALLY SWITCH ON

POWER OFF

RESOLVE FAULT POWER OFF

CONTROL OVERAGE OR FAILURE

ACCESS VIOLATION (P+ * R-)

CONTROL OVERAGE OR FAILURE

**Figure 3-1; System-level states and their transitions.**

Again please note that there is no automatic way to energize the system. This is a manual-only action. The 480 VAC is controlled by a manual-close circuit breaker with a remote (shunt) trip input, which students may be allowed to operate. The 4160 VAC switchgear has provisions to be closed remotely, but these have been disabled to obviate the risk of an erroneous or unauthorized "close" command doing harm. Energizing the 4160 VAC is the sole responsibility of the lab administrator, who must physically unlock the utility room and manually close the breakers. This will be explained further in section 3.4, Procedures.

The choice of a particular SCADA software package was constrained by the capital equipment already in place. After brief reflection, we realized only a GE product would interface with the drive effectively, making the built-in "higher functions" (the automatic referencing, data capture, diagnostics, etc.) available. GE-Fanuc's "Proficy Machine-Edition" (formerly Cimplicity) is capable of providing the graphical user interface, data-logging, alarm/messaging, and logic functions we require. Proficy is also capable of communicating by ModBus-IP, which is necessary to make use of the advanced features of the switchgear.

## *3.2 I/O system*

This section deals with the particulars of selecting I/O devices, their placement, and connections. A bill of materials is included in appendix 2.

Choice of the SCADA software package further constrained us to use GE I/O products. Others devices are available, but using them would forfeit the benefits of using an integrated family of products. Configuring communications with "brand-X" (as we discovered with the ModBus) quickly becomes an exercise in low-level programming and data formatting.

GE offers several lines of I/O products. We have chosen the VersaMax system because it is modular, expandable, and *relatively* inexpensive. The Ethernet bridge unit and power supply clamp to a standard 35mm DIN rail. "Carrier modules" also clamp to the rail and plug serially into the side of the bridge unit. These provide serial network and power connections to the bridge, as well as screw terminals for field wiring to I/O devices. Finally, the actual I/O interface module plugs into the carrier. Interface modules support between four and thirty-two channels with input or output, analog or discrete, source or sinking, current or voltage signals operating from 12 VDC up to 240 VAC. There are also special functions such as PWM output, high-speed counter input, and direct interface to RTDs and thermocouples[27].

### 3.2.1 What I/O?

From the discussion of the automation design, we can begin to decide what sensors are required, and where they must be placed. Refering back to figures 1.4, Power distribution system elementary diagram, and 1.5, Lab floor plan, may be helpful at this point. We need to know:

- If the lab bay doors are open or closed.
- If the 480\4160 VAC distribution knife-switches are open or closed.
- If the high-power distribution breakers are open or closed.
- If the lab bay ambient temperature is acceptable.
- If the substation ambient temperature is acceptable.

[27] Please see VersaMax® Modules, Power Supplies, and Carriers, User's Manual, GFK-1504K (or later revision), GE Fanuc Automation, for a complete discussion of all the elements of a VersaMax system and available modules.

- If the transformers' temperatures are acceptable.
- If there is an E-stop command from the lab.

These allow us to determine as certainly as possible that the lab is "vacant" or "occupied," and that the power is "on" or "off."

Additionally, we would like to know:

- Voltage and current at the switchgear.
- Current from the second floor substation.

These will be useful in determining time/heating and other fault states.

The system drives the following outputs:

- Warning lights (3) and siren at lab bay doors.
- Permission to energize high-power switchgear.
- Permission to energize 480 VAC distribution.

Figure 3-2; I/O locations. places these devices on the lab floor plan. Please note that I/O cabinets (with interface modules) have been placed near clusters of I/O devices. This keeps wire runs to devices as short as possible.

To serve the I/O needs detailed above, we require discrete, 24 VDC inputs and isolated relay contact outputs. These have been provided, along with a judicious number of spare points in consideration of future applications. Additionally, we have selected modules which may serve as PWM drivers or high-speed counters. Please see appendix 5 for wiring diagrams.

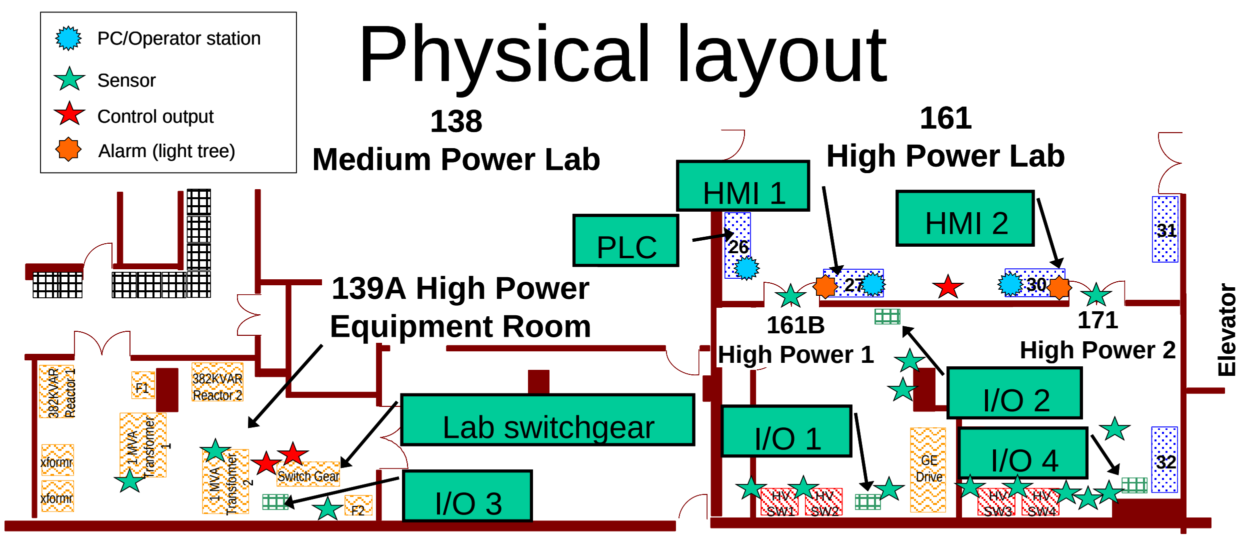


**Figure 3-2; I/O locations.**

### 3.2.2 Sensor hardware.

The sensors we have selected are all metal bodied, discrete, mechanical types. These were chosen over other types in an effort to improve immunity to EMI and because they tend to be physically robust. They are also simple to use and explain to people – they are either on or off, no confusion about the meaning of an analog value is possible. Analog signals are certainly supported by this system, but, generally speaking, discrete measurements are sufficient to indicate if a situation is within safe limits.

At first, there was a push to measure voltage at each supply outlet and equate a zero voltage measurement with a safe condition. This violates fail-safe philosophy, because zero volts measured can simply mean that the sensor is broken or disconnected. You can not test for zero volts measured as a sensor fault state either, because zero is a legitimate value. In the end, we decided to positively measure the mechanical position of the disconnect switches.

A position sensor with a "wobble stick" (a flexible spring coil) actuator is mounted inside each of the manual power disconnect switches. These are placed so that when the disconnects' internal cam snaps to the open position, it deflects the actuator, and the sensor switch closes.

The disconnects' cam moves quite violently (in order to break any arc that may be formed if it is operated under load), but a wobble stick can withstand this.

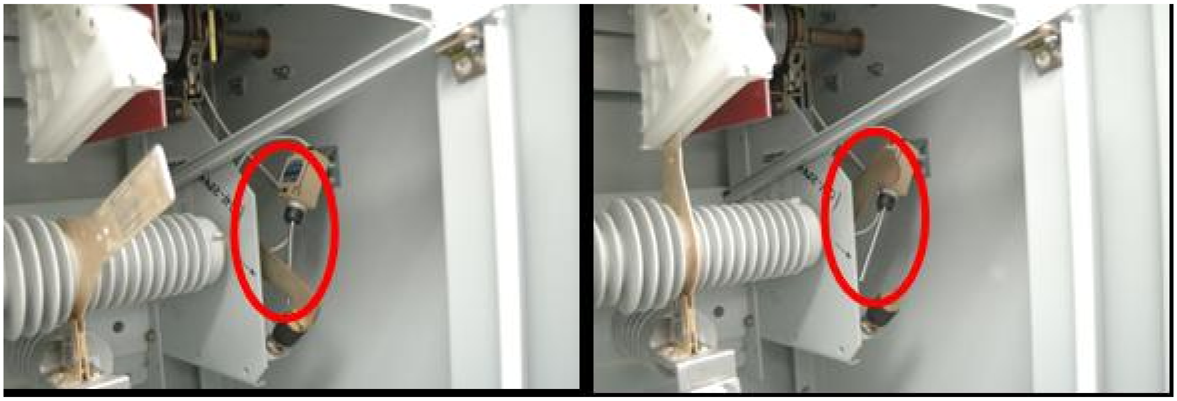


**Figure 3-3; High voltage switch status sensor "open" and "closed."**

Knowing that the lab is vacant is a more difficult proposition. We have mounted position sensors with roller-tipped, rocker arm actuators on the inside of the labs' door lintels. A closed circuit indicates the labs' doors are closed.

We have not found a satisfactory way to detect people in the room directly. Acoustic motion sensors mounted on the ceiling *might* work, but cooling fans will be quite loud and could interfere. Infrared sensors would be hopelessly blinded by heat sources in the room, and vision systems seem overly complex. "People-counters" at the door, such as photo-gates, would be very fallible unless there was a turnstile or other single-person passageway. This would be unacceptable because it would get in the way of moving equipment and slow emergency egress. In the end, it comes down to trusting in good operating procedure, rather than infallible sensor schemes.

The ambient temperature sensors are mounted near the ceiling, where heat is expected to collect. These are a simple, bimetallic strip type, with an adjustable trip point set at ~55 deg. C. This should give a generous margin for operation yet trip before wiring insulation really begins to suffer. Finally, there are other similar temperature sensors imbedded in the transformers by the manufacturer. Their setpoint is unknown, but presumably appropriate to the machine.

The lab at large has an E-stop system based on hardware relays, CRLV, CRMV, CRHV, which stand for, "Control Relay Low\Medium\High Voltage." This is a misnomer as the lab is divided into areas by *power*. Our lab is part of CRHV's bailiwick.

CRHV had a spare contact which has been used to signal its status to the high-power lab trip circuits by energizing a secondary, HVOK1, relay. HVOK1 also depends on a "permission to energize" output from the control system to close (or remain so). If either of these conditions fails, HVOK1 opens, thereby opening HVOK2 and HVOK3, which in turn send trip commands to the high-power breakers. Loss of HVOK1 will also send a trip command to the 480 VAC main breaker. See Appendix-C for elementary diagrams "EStop1AA" and "LVSG1AA," as well as the switchgear datasheets for additional information.

As stated previously, the switchgear itself has a sophisticated power analyzer built into the trip unit. This provides breaker status, as well as voltage, current, and other data via a ModBus network.

### 3.2.3 Outputs

The system outputs include the "permission to energize" signal detailed above, as well as commands to light trees mounted beside the two lab bay doors. The system states "Safe," "Operating," and "Fault" are represented by the lights. As noted before though, colors mean different things to different people, so a clear explanatory sign is required for this context.

## *3.3 Communications systems.*

We have already touched on the communications requirements of some key equipment in the lab. The following section details the complete communications topology for the lab. As each element of the system was considered, we had to identify what its possible modes of communication were and choose the most practical connections to make. The results are tabulated below. "Installed" indicates that the equipment on hand, or in general, has such a port. "Available" means that such connections are possible, usually requiring a special piece of interface equipment (PC card, bridge, etc.). Connections that are unavailable or extremely impractical are marked "N/A".

**Table 3-1; Possible communications links.**

| | Available communications modes | | | | | | |
|---|---|---|---|---|---|---|---|
| Equipment | Ethernet | Serial | USB | ModBus | Fanuc- Genius | GPIB | Hardwired |
| Soft-PLC | Installed | Installed | Installed | Available | Available | Available | N/A |
| HMI PC | Installed | Installed | Installed | Available | Available | Available | N/A |
| I/O modules | Available | Available | Available | Available | Available | N/A | N/A |
| Switchgear | Available | N/A | N/A | Installed | N/A | N/A | Installed |
| Data-logger | Available | Available | Available | Available | N/A | Available | N/A |
| Oscilloscope | Installed | Installed | Installed | N/A | N/A | Available | N/A |
| Remote PCs | Installed | N/A | N/A | N/A | N/A | N/A | N/A |
| E-stop system | N/A | N/A | N/A | N/A | N/A | N/A | Installed |
| Equipment under test | To be determined. | | | | | | |
| GE drive | Installed | Installed | N/A | N/A | N/A | N/A | Installed |

To the greatest degree practical, all wiring connections were made in accordance with installation guidance instructions published by GE Industrial Systems[28]. This document is a rather exhaustive treatise on wiring methods and materials and is highly recommended.

[28] Instillation Guidance for Innovation Series™ Drive Systems, publication GEH-6380, issued 30 June, 1999, © 1999 General Electric Company, USA.

All sensors are connected with AWG #18, solid Cu wire with shield. Some use a cable with a single pair of conductors, and others are bundled into larger cables. In a given instance the choice was based mostly on cost (in some cases, multiple small strands are cheaper than a single larger cable).

University regulations prohibit students from making connections to permanently installed equipment at other than 24 VDC, so this was used for all the sensors.

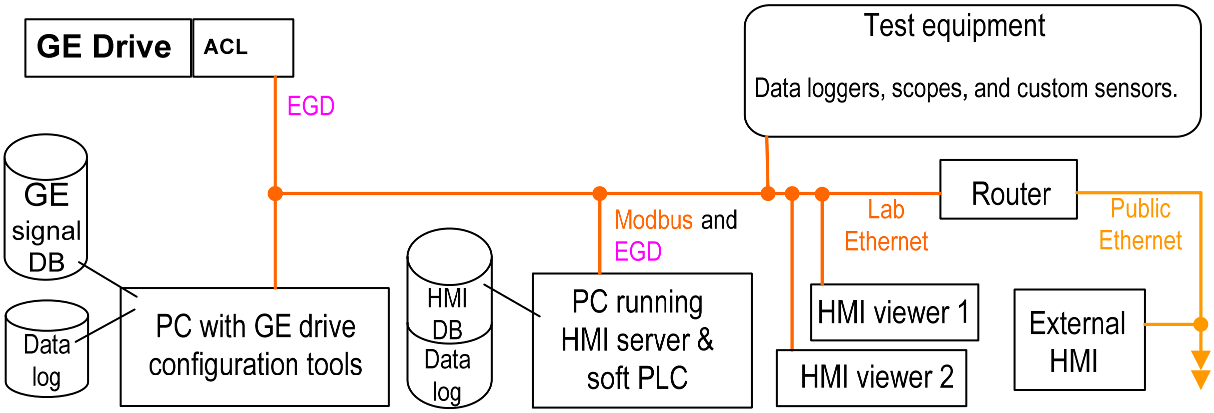


**Figure 3-4; Ethernet backbone.**

Given the prior discussion of the advantages of Ethernet connections, we began by planning an Ethernet backbone for the lab, figure Figure 3-4. A router was the first necessity, providing security and privacy by keeping external traffic off of the control network. We have installed an 8-port Linksys model for this purpose. Category 6e cable was used throughout, because it is shielded, and we expect a great deal of EMI. Crimp-on RJ-11 plugs were installed as needed. Even though the proper crimp tool was used to attach the plugs, they have been a repeated point of failure. Patch panels with insulation-displacement sockets would probably work better.

Any PC will, of course, be able to use Ethernet, so connecting a soft-PLC and HMI viewers was no problem. The GE drive supports a direct Ethernet connection via its ACL coprocessor, using a proprietary EGD protocol. This is inconvenient, but still feasible. No test equipment is connected at the moment, but Ethernet connections on such are commonplace.

The Square-D switchgear is connected to Ethernet via a bridge. The switchgear uses a 4-conductor serial Modbus internally, with two drops for "cradle-communications modules," on the network. These modules interface to the trip units' CPUs. Communications to the switchgear are addressed by the Ethernet IP address of the bridge, the Modbus drop address of the CCM, and finally, the register address of interest. User software can query the switchgear directly, or go through the HMI which does the bookkeeping for us. Square-D recommends that the power supply for the communications be independent of the controlled line voltage to avoid ground loops presumably. Further instructions, including wiring diagrams and data addressing, can be found in the manufacturer's datasheets[29,30,31] and in the Users Manual, Appendix-B.

Modbus is a daisy-chain serial network, so adding drops is straightforward[32] provided the total number of drops nor the total length of cable is excessive. Termination resistors are required at both ends.

The switchgear also has hardwired connections to E-stop and to "safety" I/O which consists of interlocks to loads (typically "permission to energize" signals), Figure 3-5.

[29] Power-Zone4® Instruction Bulletin, Bulletin No. 80298-002-05, December 2003, © 1999–2003 Schneider Electric, 8821 Garners Ferry Road, Columbia, SC 29209 USA, 1-888-SquareD, (1-888-778-2733), www.us.SquareD.com

[30] Power-Zone4® Catalog 05, © 2005 Schneider Electric, Ibid.

[31] Instruction Bulletin, MICROLOGIC® 5.0P and 6.0P Electronic Trip Units v 7.522, Bulletin No. 48049-137-02 06/01, © 2000–2001 Schneider Electric, Square D Company, PO Box 3069, 3700 Sixth St SW, Cedar Rapids IA 52406-3069 USA, 1-888-SquareD (1-888-778-2733), www.SquareD.com

[32] There are four wires and a shield/ground, which are color coded for identification. Connect a new device's wires in series with existing wiring. If adding to an end, move the terminating resistor to the new end of the chain.

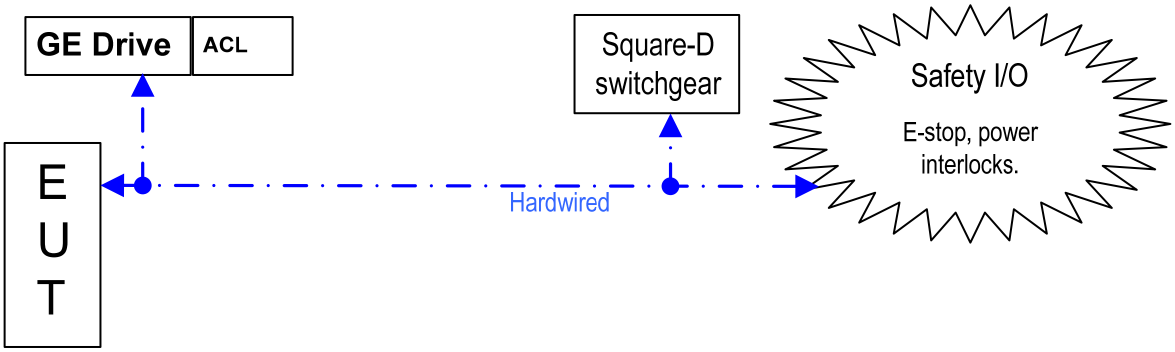


**Figure 3-5; Hardwired interlocks.**

From chapter 2 the reader will recall that I/O in general is connected to Ethernet-enabled communications modules – bridges in their own right between serially connected signal conditioning hardware and a network interface/logical unit. The complete Ethernet topology is shown in Figure 3-6.

All fixed devices are configured with static IP addresses. The addresses are used at compile-time to set up runtime messages between devices, so changing a device's address will lead to dangerously garbled communications. In general, there is no call for a permanent device to change address.

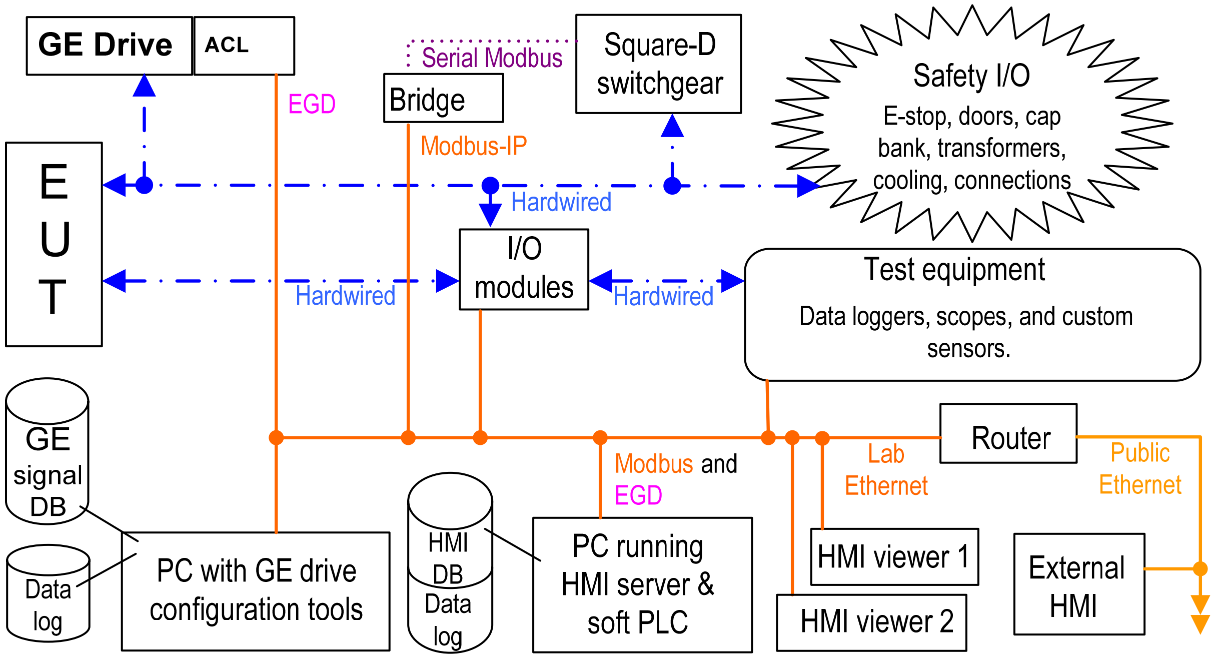


**Figure 3-6; Complete Ethernet topology.**

Finally, serial links and special purpose wiring complete the system communications topology, Figure 3-7. This is a maximum-case build out, showing all likely connections. With this scheme, the facility is both observable (either directly or in analog) and controllable. Communication delays are quantifiable and data integrity can be verified. It remains to good practice to control the system well, but the necessary resources are available.

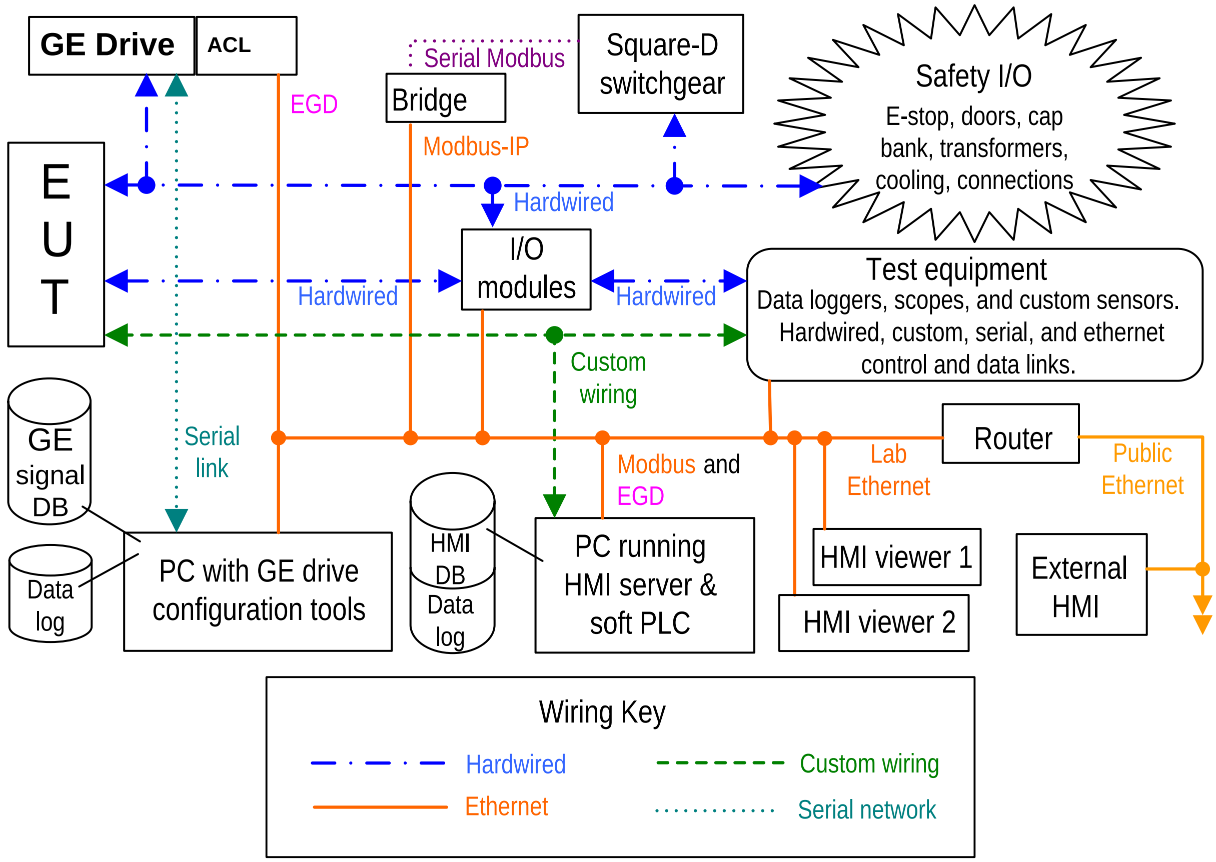


**Figure 3-7; Complete communications topology.**

## 3.4 Procedures

Industrial settings commonly allow for equipment to be energized automatically, but there are several reasons why this is not wise in an academic laboratory setting. Chief among these is that industrial systems are commissioned by safety-obsessed professionals and then *left alone*, not poked and prodded as an experimental setup will be. Even so, there are catastrophic failures.

The following operating procedures have been drafted with this in mind. They will need modification as the lab equipment changes, especially as energy storage elements and drive systems are added.

There are shortcomings in any mechanical layout or control system, so it is imperative that lab users understand what procedural safeguards are in place and abide by them. A major deficiency of the existing system is that there is no way to lock-out the 480VAC outside the room (the 4160VAC can be locked out at the switchgear). This could lead to someone closing a local disconnect that is "hot," courtesy of someone else's carelessness.

Procedure 3.4.1, Safing the room, must always be followed, even when only making minor changes (such as moving probes or changing low-voltage connections). People get killed by dropping tools or loose screws, or by touching the wrong thing accidentally.

Procedures 3.4.2 and 3.4.3 for energizing the room give the opportunity to check for faults. If a system has been operating normally and has had only minor changes (such as moving a probe), then it can probably just be switched on again.

### 3.4.1 Safing the room

1) **Set all loads to "off" state.**
2) **Open the appropriate supply connections by selecting "manual trip" from HMI.**
3) **Verify that appropriate supply connections are open:**
   - **Visually verify that 480VAC main breaker in room 161/171 is open.**
   - **Check switchgear status indication for 4160VAC supply on HMI is open.**
4) **If 480VAC is used, manually open appropriate branch feed circuit breaker.**
5) **If stored energy is present in the experiment bay wait for it to dissipate.** Electrical codes state that capacitors should be configured to bleed off to < 50 VDC within five minutes, but don't trust this to actually happen – check!
6) **If 480VAC supply is connected, post a guard on the main breaker.** An energy source that can not be locked-out must be guarded by a person.

It is now safe to enter the experiment bay ***<u>unless</u>*** there are energy storage devices in use.

**3.4.1a, No energy storage or alternate sources present.**

No large capacitors, no other power connections (120/208 VAC, etc.).

7) **Latch doors open.**
8) **If any adjustments are to be made to the 480VAC connections, lock/tag them out. The guard on the main breaker may be relieved once this is done.**

**3.4.1b, Energy storage present.**

- **Put on proper protective gear (glasses, flame-retardant smock, hard hat).**
- **Latch doors open.**
- **Disconnect alternate energy sources.**
- **Use a voltage-level appropriate hot-stick and voltage tester to check that storage devices are discharged.**

- **Install safety grounds.**
  Ground conductors, as a rule, should be no more than two wire sizes smaller that the supply wiring. Anything less is not a ground, it is a fuse!
- **If any adjustments are to be made to the 480VAC connections, lock/tag them out. The guard on the main breaker may be relieved once this is done.**

**End.**

### 3.4.2 Energizing 480VAC wall outlets

1) **Safe the room.**
   All energy sources must be safe before entering the room, even ones not directly associated with the circuit to be energized.
2) **Post a guard on the main breaker.**
   There is no way to lock-out the main breaker, so a person must be delegated to watch it.
3) **Open the knife switch (if it is not already locked-out).**
   In case there is current present, this will contain the arc. Stand to one side of the panel and look away when you operate the switch.
4) **Open the panel door.**
5) **Verify that both the line- and load-side voltages are zero.**
   Be sure that what you think you turned off is really off.
6) **Remove safety grounds.**
7) **Test for load side shorts between phases and to ground.**
   Bad connections, bad insulation, or other damaged components, tools or debris left in equipment, can all cause explosions when equipment is energized.[33]
8) **Shut the panel.**
9) **Double check that the main breaker is still off.**
10) **Remove locks/tags from wall switch.**
    Everyone who has been working on this circuit has to personally remove their own safety lock-out.
11) **Close the knife switch.**
12) **Verify that proper status is reflected on the HMI.**
    The SCADA can not protect anything without proper feedback.
13) **Energize auxiliary sources.**
14) **Clear everyone from the room and close the main doors.**
    There is no way at present for the SCADA to know if the room is truly empty – be careful.
15) **Verify permission to energize status on HMI.**
    This ensures that there are no other system faults
16) **Close the main 480VAC breaker.**
17) **Close the branch feed breaker.**

**End.**

[33] Professionals literally count their tools before and after working on high power equipment to make sure nothing has been left inside.

### 3.4.3 Energizing MV supply

1) **Safe the room.**
   All energy sources must be safe before entering the room, even ones not directly associated with the circuit to be energized.
2) **If 480VAC connections are in use, perform steps 1-12 of the 480VAC checklist.**
   Check everything and get it ready to energize.
3) **Open the knife switch (if it is not already locked-out).**
   In case there is current present, this will contain the arc. Stand to one side of the panel and look away when you operate the switch.
4) **Open the panel door.**
5) **Verify that both the line- and load-side voltages are zero.**
   Be sure that what you think you turned off is really off.
6) **Remove safety grounds.**
7) **Test for load side shorts between phases and to ground.**
   Bad connections, bad insulation or other damaged components, tools or debris left in equipment, can all cause explosions when equipment is energized.
8) **Shut the panel.**
9) **Double check that the main breaker is still off.**
10) **Remove locks/tags from wall switch.**
    Everyone who has been working on this circuit has to personally remove their own safety lock-out.
11) **Close the knife switch.**
12) **Verify that proper status is reflected on the HMI.**
    The SCADA can not protect anything without proper feedback.
13) **Energize auxiliary sources.**
14) **Clear everyone from the room and close the main doors.**
    There is no way at present for the SCADA to know if the room is truly empty – be careful.
15) **Verify permission to energize status on HMI.**
    This ensures that there are no other system faults
16) **Lab administrator will go to room 139 and close the MV switchgear .**
17) **Close any 480VAC circuit breakers as required.**

**End.**

# Chapter 4 – Testing and Verification.

This chapter will document the commissioning of the control system hardware and software. This is how we tested the new equipment and software, so this chapter can serve as a guide for others re-verifying system integrity after it has been modified.

The fastest way to find faults in a causal chain is with a binary search. Cut the chain in half and test a segment. If that segment is OK, then the fault must be in the other segment (and you have just avoided testing half of the nodes). Cut the faulted segment(s) in half and repeat. An obvious first cut in this system is the dividing line between hardware and software.

We begin then with how we tested the hardware

## *4.1 Hardware commissioning*

This is not complicated, but does require strict attention to detail. Wires have to go to the *correct* terminals, jumpers and switches must likewise be correct, proper network terminations are required, etc. Wiring elementary diagrams, produced during the project planning stage, are a guide to where connections should be made. These must be updated to "as-built" status to reflect changes made during construction, and must show every single wire connection point to be any use in troubleshooting later problems.

The hardware has its own division point at the network level. All the "field" devices, the actual sensors and actuators are on one side and the network connectivity is on the other.

### 4.1.1 I/O, field device to module

Testing I/O devices is usually just an exercise in applied linear networks. Signals have to start somewhere; trace the power from source through the device and back to ground. Observe that the correct things happen along the way. Success is unambiguous. Actuating a sensor makes an indicator light up on an input module (Figure 4-1), or a working output "makes something go klunk."

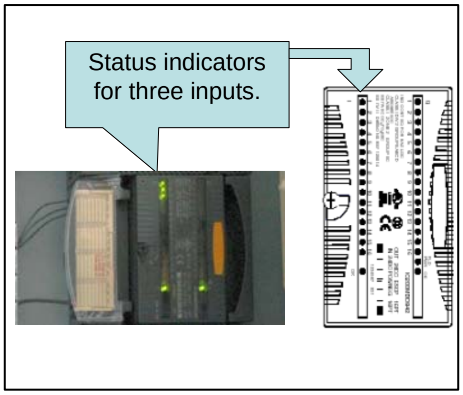


**Figure 4-1; I/O module (and schematic) showing three "true" inputs.**

#### 4.1.1.1 Discrete inputs

Discrete inputs are the simplest of the simple - in essence just a switch. They are tested by opening and closing, the current through them being read by an I/O module. If opening is impractical, then a wire can be taken loose to break the circuit. We did this with the transformer over-temperature sensors (which are inaccessible) to check we had them identified correctly.

Start tracing problems by testing the mechanical hardware. Is the switch actually opening/closing mechanically (the switch body may have moved)? Does it open/close electrically? Does it have proper incoming voltage? If not, trace back to the supply. Is the signaling voltage correct? Trace it to the input terminal. Is there noise voltage on the signal, especially transients, when some other load is switched?

#### 4.1.1.2 Discrete outputs

Discrete output problems can be localized to field or module very easily by shorting around the module terminals with a wire jumper.

There are two types of discrete output modules installed, the isolated relay contact (which is again, just a switch) and a current source/sink which provides +/- 24 VDC, depending on how it is wired. These can be tested with a multimeter right at the module, though the load may need to be disconnected.

Discrete loads can be relay coils (as with the signals to the E-stop and switchgear), or solid-state devices as in the light-trees. In either case, the impedance can be checked at the module end to see if the load is as expected.

#### 4.1.1.3 Analog signals

We do not have any analog signals in the lab safety system (yet), but commissioning/troubleshooting techniques for them are covered here for completeness sake. There really is not too much difference with discretes, continuity must be maintained just the same.

The biggest difference with analog signals is that checking them requires corroborating field measurements with the state of the I/O module. Use the HMI to set or monitor the module's state, so you know what to expect to measure. External precision current/voltage sources can be used to signal either a load or input module to see if they behave properly. Outputs can be validated directly[34] with a multimeter.

The most difficult case will be checking an analog sensor, because the state it is sensing must be known.

### 4.1.2 I/O, network connectivity

The second half of getting I/O to work properly is communicating with and configuring the interface modules. At an early stage in this project, we set the network communications modules on a convenient workbench, hooked in dummy inputs, and went through the manufacturer's network configuration procedure. The HMI has an online monitoring mode that let us verify that we had communications. We then were able, over the network, to configure the modules to expect appropriate field connections. This accomplished, we observed the dummy I/O states faithfully reported to the HMI.

[34] A word of caution. One of the author's former colleagues tripped a very large utility substation by measuring voltage on a multimeter configured for current. This should not happen here, but you never know.

Satisfied they would communicate, we mounted the modules in the field and repeated the dummy I/O tests, this time checking the newly installed network. This lead to much practice re-doing Ethernet connectors.

In the case of the switchgear, we monitored the date and time status messages from the trip units to prove we were reading good data.

If an I/O problem persists after the field wiring is checked, verify that the wire has not come loose in the module terminal by pulling on the wire (loose wiring terminals are *the* most common connection problem).

Above the individual signal wire termination, the chain of network connectivity goes from module internal values, through network bridge, router, PC and finally PLC software configuration. Failures anywhere along this chain will affect multiple signals, so looking for commonalities between malfunctioning signals is a good troubleshooting technique.

### 4.1.3 (Re)verification

Verifying that a system, once commissioned, still works just requires exercising all the devices and seeing them behave properly. Think of it as a pre-flight inspection, "Flaps? Check. Rudder? Check…" It is easier than commissioning, because there may not *be* any faults, or at least probably only one in a given channel.

In practice, this means the signal from physical connection, all the way through to logical representation, can be checked at once. One must still go and move each physical actuator (doors, disconnects), but one can then just check the system log to see that each transitioned when and only when it was tested.

An example in the following, Figure 4-2, shows how automatically recorded data can be retrieved from the system log.

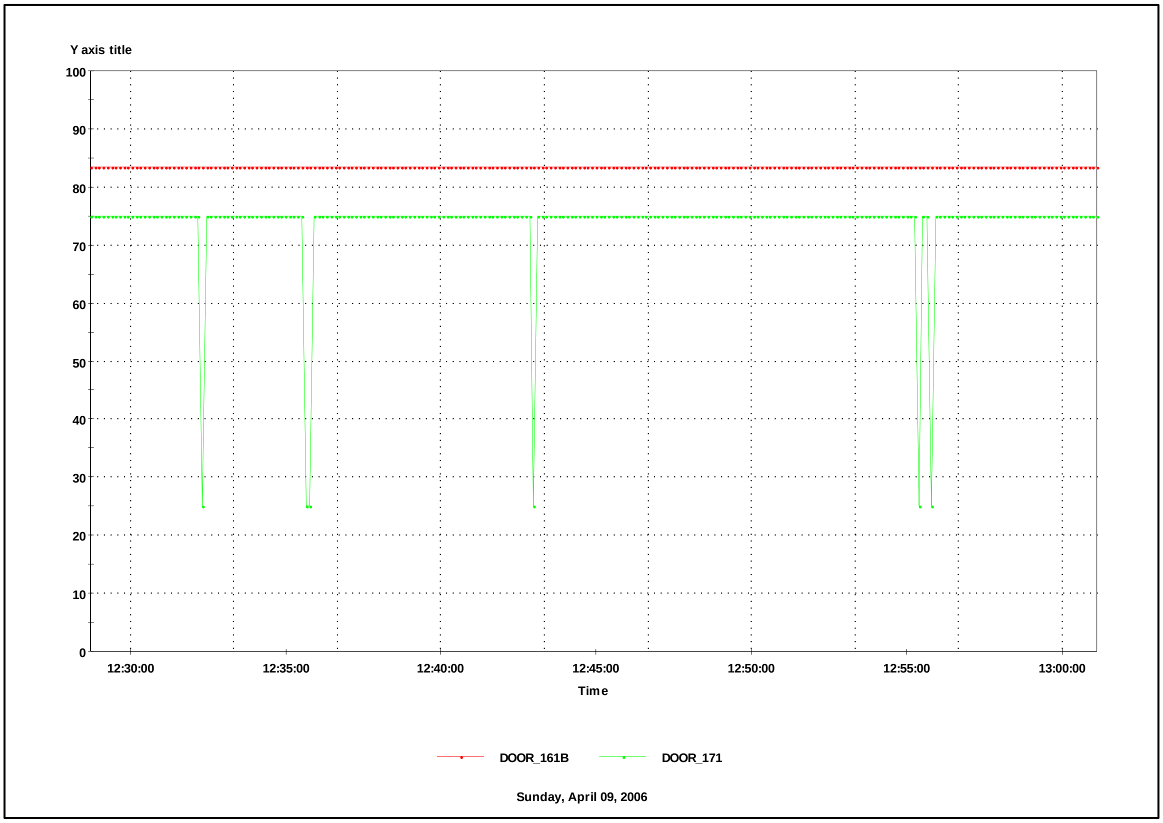


**Figure 4-2; Historical data retrieved from the system log, showing traffic into one lab bay. These are binary signals so the Y-axis scale is arbitrary and has been set for readability.**

The lab administrator may well require this log, satisfying a checklist, before allowing the lab to be energized.

Obviously a friend watching an HMI status screen, which displayed the checklist progress and indicated which equipment to test next, would be helpful.

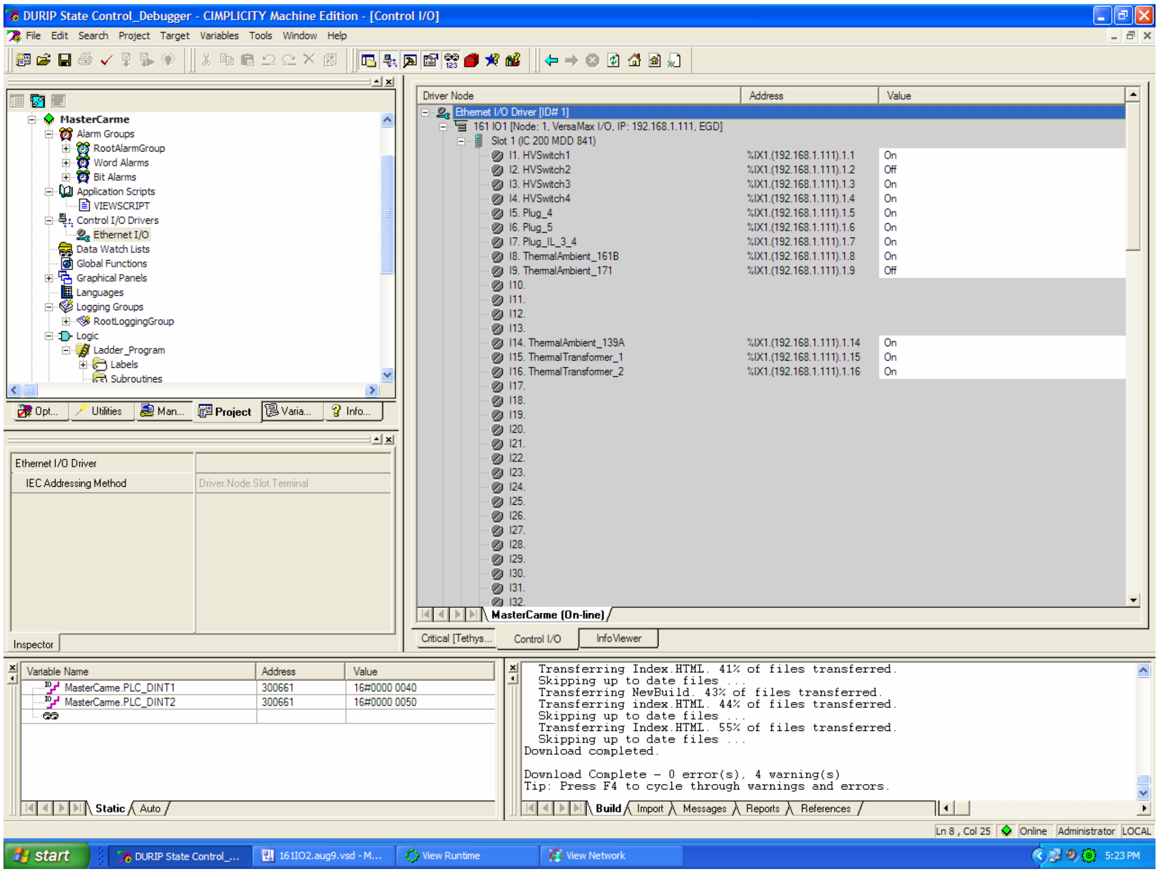
## *4.2 Software commissioning*

Having succeeded in getting all the I/O signals into the system is a major step. It still remains to be seen that the data is received and interpreted properly and that logic functions correctly.

### 4.2.1 Variable assignments.

The correct logical meaning must be assigned to each received signal. As-built wiring elementaries are a crucial check at this stage, showing what is *really* connected to each I/O point.

The true test, though, comes in watching a live HMI view as devices are actuated one by one. Figure 4-3 below is a screen capture of the HMI online monitor, showing status of a group of inputs.


DURIP State Control_Debugger - CIMPLICITY Machine Edition - [Control I/O]
File Edit Search Project Target Variables Tools Window Help
MasterCarme
Alarm Groups
RootAlarmGroup
Word Alarms
Bit Alarms
Application Scripts
VIEWSCRIPT
Control I/O Drivers
Ethernet I/O
Data Watch Lists
Global Functions
Graphical Panels
Languages
Logging Groups
RootLoggingGroup
Logic
Ladder_Program
Labels
Subroutines
Opt... Utilities Man... Project Varia... Info...
Inspector
MasterCarme (On-line)
Critical [Tethys... Control I/O InfoViewer
Static Auto
Build Import Messages Reports References
Ln 8 , Col 25 Online Administrator LOCAL
start DURIP State Control_... 161IO2.aug9.vsd - M... View Runtime View Network 5:23 PM

| Driver Node | Address | Value |
|---|---|---|
| Ethernet I/O Driver [ID# 1] | | |
| 161 IO1 [Node: 1, VersaMax I/O, IP: 192.168.1.111, EGD] | | |
| Slot 1 (IC 200 MDD 841) | | |
| I1. HVSwitch1 | %IX1.(192.168.1.111).1.1 | On |
| I2. HVSwitch2 | %IX1.(192.168.1.111).1.2 | Off |
| I3. HVSwitch3 | %IX1.(192.168.1.111).1.3 | On |
| I4. HVSwitch4 | %IX1.(192.168.1.111).1.4 | On |
| I5. Plug_4 | %IX1.(192.168.1.111).1.5 | On |
| I6. Plug_5 | %IX1.(192.168.1.111).1.6 | On |
| I7. Plug_IL_3_4 | %IX1.(192.168.1.111).1.7 | On |
| I8. ThermalAmbient_161B | %IX1.(192.168.1.111).1.8 | On |
| I9. ThermalAmbient_171 | %IX1.(192.168.1.111).1.9 | Off |
| I10. | | |
| I11. | | |
| I12. | | |
| I13. | | |
| I14. ThermalAmbient_139A | %IX1.(192.168.1.111).1.14 | On |
| I15. ThermalTransformer_1 | %IX1.(192.168.1.111).1.15 | On |
| I16. ThermalTransformer_2 | %IX1.(192.168.1.111).1.16 | On |
| I17. | | |
| I18. | | |
| I19. | | |
| I20. | | |
| I21. | | |
| I22. | | |
| I23. | | |
| I24. | | |
| I25. | | |
| I26. | | |
| I27. | | |
| I28. | | |
| I29. | | |
| I30. | | |
| I31. | | |
| I32. | | |

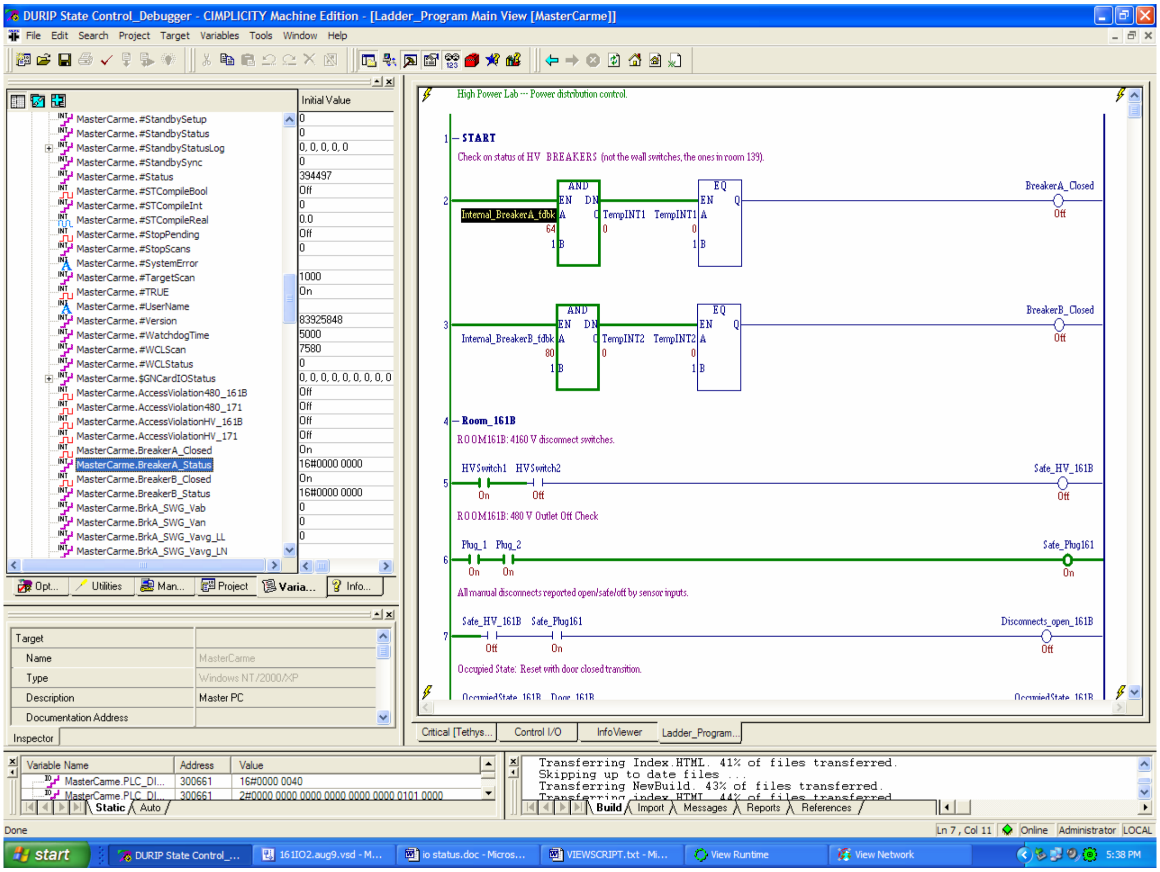
| Ethernet I/O Driver | |
|---|---|
| IEC Addressing Method | Driver.Node.Slot.Terminal |

| Variable Name | Address | Value |
|---|---|---|
| MasterCarme.PLC_DINT1 | 300661 | 16#0000 0040 |
| MasterCarme.PLC_DINT2 | 300661 | 16#0000 0050 |

```
  Transferring Index.HTML. 41% of files transferred.
  Skipping up to date files ...
  Transferring NewBuild. 43% of files transferred.
  Transferring index.HTML. 44% of files transferred.
  Skipping up to date files ...
  Transferring Index.HTML. 55% of files transferred.
  Skipping up to date files ...
Download completed.

Download Complete - 0 error(s), 4 warning(s)
Tip: Press F4 to cycle through warnings and errors.
```

**Figure 4-3; Screen capture of control system editor, in online monitor mode, showing I/O status on the upper right for several devices.**

Figure 4-4 shows another perspective, with the logic editor online. "True" inputs are highlighted, which makes debugging easier.



| | Initial Value |
|---|---|
| MasterCarme.#StandbySetup | 0 |
| MasterCarme.#StandbyStatus | 0 |
| MasterCarme.#StandbyStatusLog | 0, 0, 0, 0, 0 |
| MasterCarme.#StandbySync | 0 |
| MasterCarme.#Status | 394497 |
| MasterCarme.#STCompileBool | Off |
| MasterCarme.#STCompileInt | 0 |
| MasterCarme.#STCompileReal | 0.0 |
| MasterCarme.#StopPending | Off |
| MasterCarme.#StopScans | 0 |
| MasterCarme.#SystemError | |
| MasterCarme.#TargetScan | 1000 |
| MasterCarme.#TRUE | On |
| MasterCarme.#UserName | |
| MasterCarme.#Version | 83925848 |
| MasterCarme.#WatchdogTime | 5000 |
| MasterCarme.#WCLScan | 7580 |
| MasterCarme.#WCLStatus | 0 |
| MasterCarme.$GNCardIOStatus | 0, 0, 0, 0, 0, 0, 0, 0, 0 |
| MasterCarme.AccessViolation480_161B | Off |
| MasterCarme.AccessViolation480_171 | Off |
| MasterCarme.AccessViolationHV_161B | Off |
| MasterCarme.AccessViolationHV_171 | Off |
| MasterCarme.BreakerA_Closed | On |
| MasterCarme.BreakerA_Status | 16#0000 0000 |
| MasterCarme.BreakerB_Closed | On |
| MasterCarme.BreakerB_Status | 16#0000 0000 |
| MasterCarme.BrkA_SWG_Vab | 0 |
| MasterCarme.BrkA_SWG_Van | 0 |
| MasterCarme.BrkA_SWG_Vavg_LL | 0 |
| MasterCarme.BrkA_SWG_Vavg_LN | |

| Target | |
|---|---|
| Name | MasterCarme |
| Type | Windows NT/2000/XP |
| Description | Master PC |
| Documentation Address | |

High Power Lab --- Power distribution control.

1 — START

Check on status of HV BREAKERS (not the wall switches, the ones in room 139).

2

3

4 — Room_161B

ROOM161B: 4160 V disconnect switches.

5

ROOM161B: 480 V Outlet Off Check

6

All manual disconnects reported open/safe/off by sensor inputs.

7

Occupied State: Reset with door closed transition.

| Variable Name | Address | Value |
|---|---|---|
| MasterCarme.PLC_DI... | 300661 | 16#0000 0040 |
| MasterCarme.PLC_DI... | 300661 | 2#0000 0000 0000 0000 0000 0000 0101 0000 |

```
Transferring Index.HTML. 41% of files transferred.
Skipping up to date files ...
Transferring NewBuild. 43% of files transferred.
Transferring index.HTML. 44% of files transferred.
```

**Figure 4-4; Screen capture of control system editor, in online monitor mode, showing I/O status as used in logic. Bold, green highlighting indicates that the signal is active.**

It is worth reiterating that fail-safe signals show a positive feedback when in a healthy condition and a null when the system is inoperable. Invariably, the sense of some input will be reversed (showing true for inoperative, false for OK), and go undetected until software checkout. The temptation will be to reverse the logic at this point, but this must *not* be done. "Not unhealthy" is not equivalent to "definitely healthy." Go and change the wiring.

## 4.2.2 Logical tests

As with any other software, the time comes to do live testing on the system by setting up realistic scenarios and observing how the system reacts.

The following Figure 4-5 illustrates two different "access violation" conditions, where a person may be in proximity to energized equipment. The line-side utility feed to the switchgear is not energized, and the breakers are "racked-out" to the logic test position where they operate, but do not make contact with the main power buss. This is a safe condition, but our controls do not consider these factors, so the test is realistic.

Initially, the High-Voltage disconnect switch feed from CB #1is open, so the room is safe (as far as this breaker is concerned). The switch is closed at ~22:51:40, shown by the signal "HV_Switch" going false. The breaker should open at this point, because a person must be in the room to close the switch, *ergo* the room is occupied. The breaker does *not* open in this case, revealing a bug which was soon remedied. We are reminded again, testing is important.

Continuing with the test analysis then, two seconds prior to the cursor position, the door to room 161B is opened and "Door_161B" goes false. This time, the access violation is recognized and the output, "Energize_Breaker_A," which permits the breaker to stay closed, goes false immediately. The breaker confirms that it is open within one second. When the room door closes again at 22:52:00, the access violation condition is resolved, but the breaker remains open. Recall that the breaker is behind a locked door and must be manually reset.

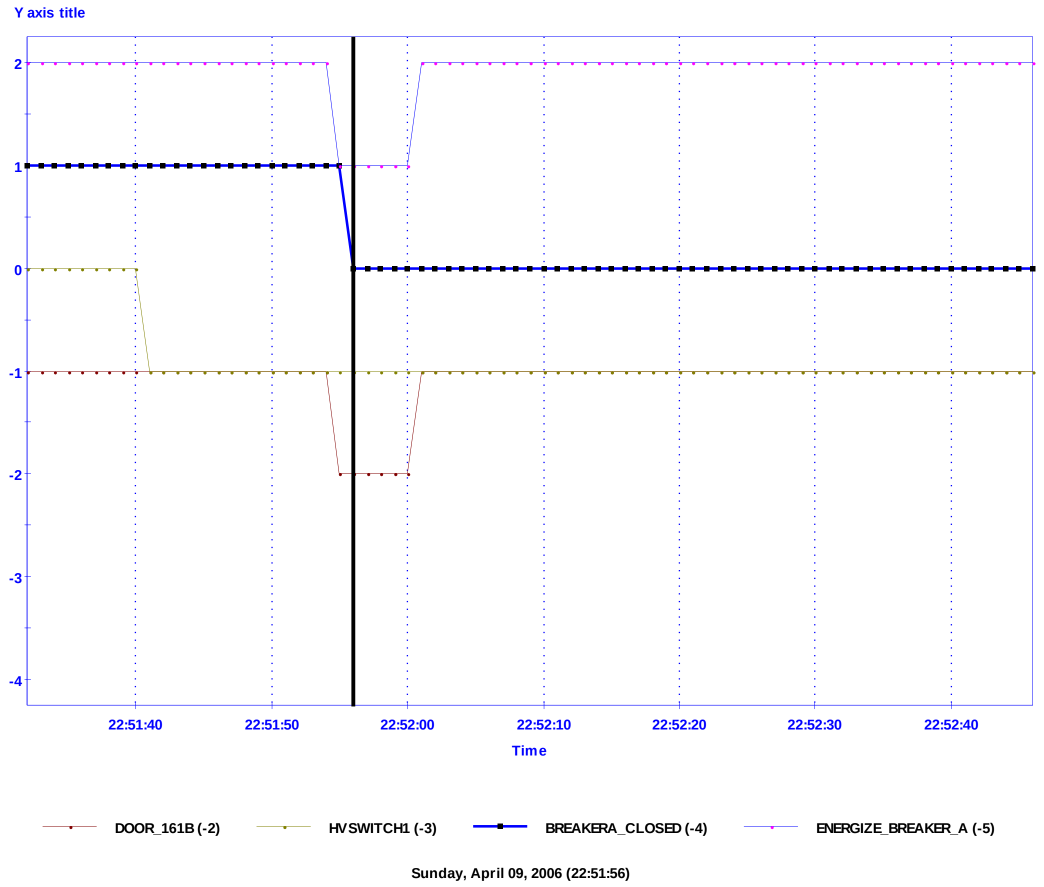


**Figure 4-5; Access violation testing of CB#1.**

The second test, Figure 4-6, is the control system reacting to an external fault (as opposed to the operator-error in the first test). From the circuit breaker's point of view it hardly matters – it must open.

Transformer #2 has been simulated overheating by breaking the fail-safe thermal overload detection circuit (just as would happen if the over-temperature klixon opened). This occurred at ~22:21:42. The cursor shows the next historical data record, one second later, after the signal “ThermalTransformer_2” has changed value to “0.”

There is a programmed thirty second lag to filter spurious events, after which (at ~22:22:13) the signal “Energize_Breaker_B” goes false. The breaker reports that it is indeed open at ~22:22:14.

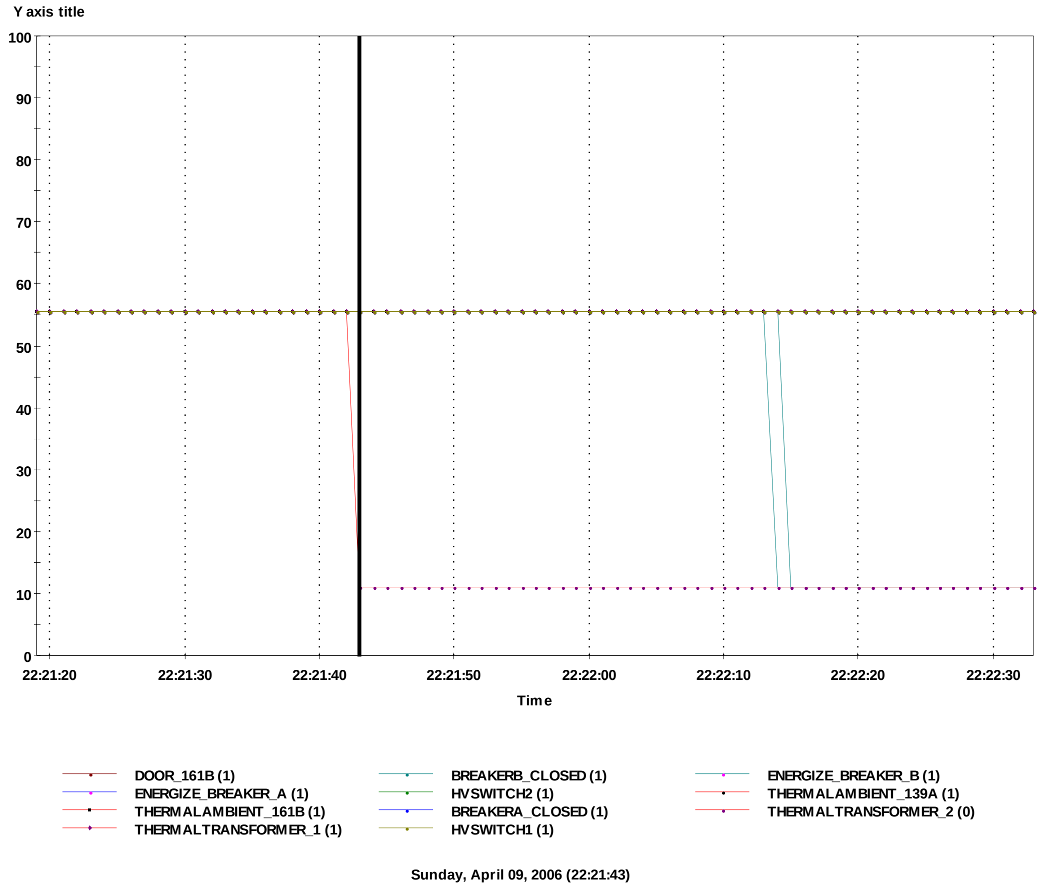


**Figure 4-6; Transformer fault test, CB#1.**

Testing for user-added fault conditions, as in support of a particular project, would look much the same.

# Chapter 5 – Results, Conclusions, and Guidance.

As with any major construction project, building the high-power lab has had things that went well and things that didn't. A unique aspect of building a laboratory space is that it is never really done. At this stage we have a proof of concept for each critical feature of the lab, and a rational framework for adapting the facility to meet the needs of future projects.

## *5.1 Weaknesses in the design.*

The first phase of our effort was to identify specifications for the lab and characteristics of the equipment. As a result of this analysis, we have identified three features of the lab which make its safe operation problematical. Overcoming these issues is possible, and we address them again here.

First, the SCADA system is unable to directly, positively observe that the lab bay is unoccupied. We have overcome this by designing operating procedures and hardware interlocks that require a person to always have the final decision to turn on the power.

Secondly, there is a danger of closing an energized disconnect switch. The control has successfully negated this problem by interlocking the doorways with the power-enabling outputs. If procedure is followed, the room will be empty before there is any power available, and if someone enters (a necessary step to close a disconnect), then the feeders will trip. One exception to this is if one leg of the 4160 is in use, and someone closes the switch for that same leg in the other bay. The only prevention for this is an administrative padlock on the switch, again a procedural point.

The final problem is that all the incoming circuit breakers require energy to open. Control power is typically from a separate supply, meaning that it is vulnerable to interruption while the main power remains on, regardless of what the control system commands. There is not much that can be done about this now. Adding a UPS would help, but is still not a guarantee.

These issues are all serious but not insurmountable. They require more care in drafting operating procedures, including more exhaustive pre-energization checks and more attention from the lab administrator. Attention to safety procedures and a healthy paranoia are highly recommended – just because something is supposed to be off is no proof that it is.

## 5.2 Successes

The preceding thoughts aside, the construction project has been quite successful. We have identified the requirements of the users and the capabilities of the equipment. The communications, I/O, and commercial SCADA package we have implemented give us the capability to meet those needs and make the most of our resources.

We have demonstrated automation functions that enforce operations interlocks that adapt to the equipment in use. Students have designed and implemented their own graphical HMI screens for remote equipment monitoring and control. There is an alarm system to annunciate events of interest and a data logger to keep a record. The system supports a variety of coding languages for user functions which increases ease of use.

Importantly, it is possible for users to play with graphics and controls without compromising the administrative software. The system is secure both physically and electronically, and it can not be energized accidentally. The E-stop interlocks are fail-safe. Loss of either external E-stop or internal system integrity will trip the power, as will failures of the controller or network.

Operations procedures have been drafted for the use of the facility. We have compiled a body of documentation for future users and written project planning aids, including commissioning guidelines.

In short, we have demonstrated that this is a viable control system, containing all the elements necessary to meet present and future needs. At a minimum, the control system will protect people and the power distribution hardware. The infrastructure exists to support additional I/O and logic that will be required to provide custom protection and instrumentation / analysis for future research projects.

## 5.3 Future

The possibility for future enhancements is basically unlimited by the system design. The SCADA software supports 75 I/O points at present, but this is only a licensing restriction. It can expand to thousands, as can the I/O hardware. Since we use a soft-PLC, we can always upgrade the computer as processing speed or other requirements dictate.

What follows is a wish-list of features that could be added with little additional investment. First, much more could be done with the HMI to graphically illustrate the health of the I/O and give diagnostic assistance to users. Next, we have access to data from the second floor substation via Ethernet and should consider adding this to our displays and interlocks. We can also set up the HMI to publish data to remote users outside the lab. Lastly, we can use existing I/O to coordinate data capture by external devices and integrate the result with our existing data log.

Since the lab already has water cooling installed, it would be reasonable to instrument the flow as an interlock to energize a liquid-cooled load. We could also monitor temperature and perform valve control as required.

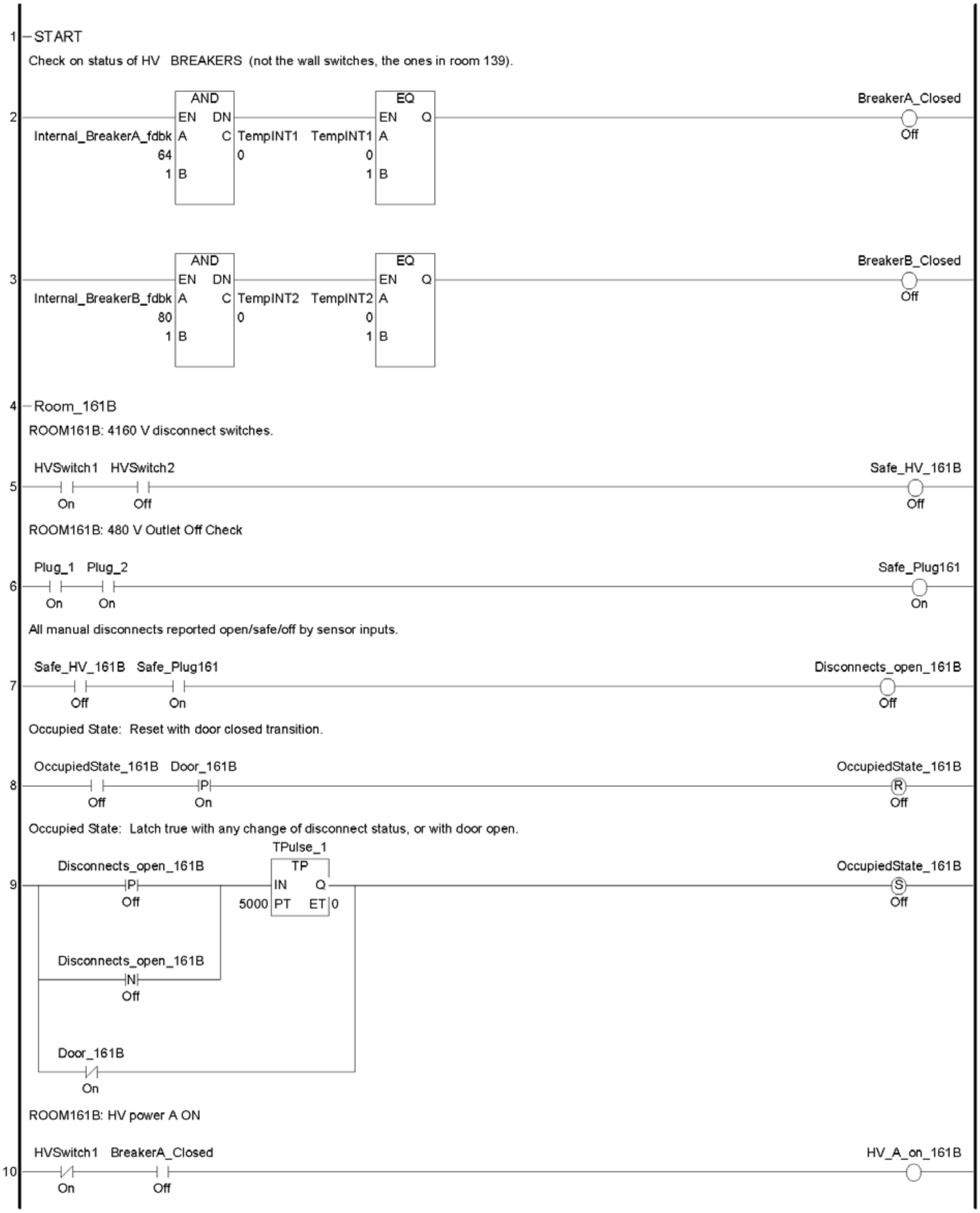
The existing lab protections would benefit from having redundant/complimentary sensors to improve reliability. Secure power for the trip circuits is important to add. Lastly, the mechanical fittings for lock-out grounding could be improved with ball fittings for clamping to.

This may seem like a long list, but that is a good thing because it indicates we have possibilities available. If the lab were a static, inflexible construct it would be much less useful.

# Appendix A – Ladder Program

This is the ladder logic program and I/O map extracted from the lab controller, along with a data-transfer script that runs as a background process on the controller.

## 6.1 Ladder Program



START

Check on status of HV BREAKERS (not the wall switches, the ones in room 139).

Room_161B

ROOM161B: 4160 V disconnect switches.

ROOM161B: 480 V Outlet Off Check

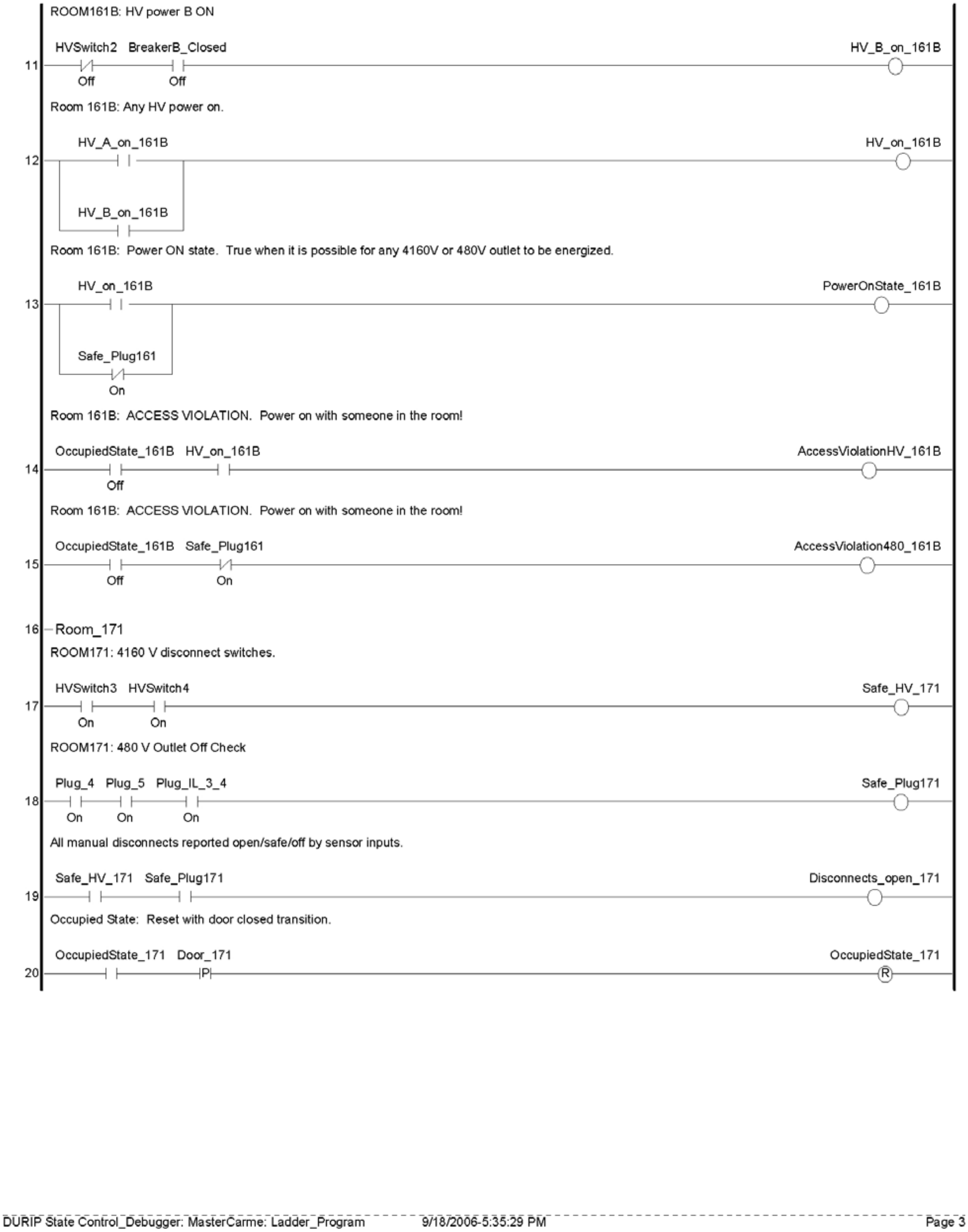
All manual disconnects reported open/safe/off by sensor inputs.

Occupied State: Reset with door closed transition.

Occupied State: Latch true with any change of disconnect status, or with door open.

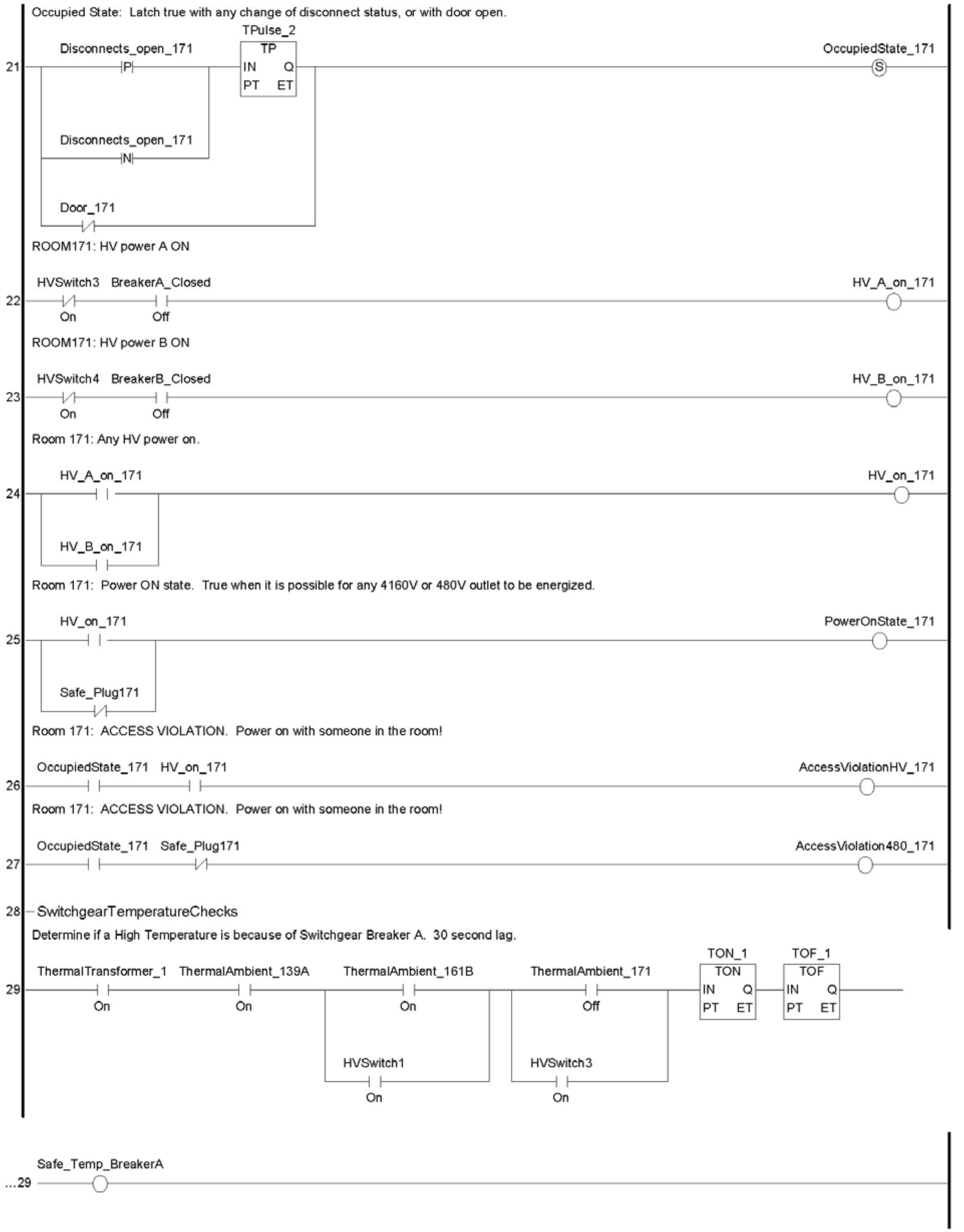
ROOM161B: HV power A ON

DURIP State Control_Debugger: MasterCarme: Ladder_Program 9/18/2006-5:35:29 PM Page 2



```
ROOM161B: HV power B ON
11  HVSwitch2 (-|/|-, Off)  BreakerB_Closed (-| |-, Off)  --------------------  HV_B_on_161B -( )-

Room 161B: Any HV power on.
12  [ HV_A_on_161B (-| |-)  OR  HV_B_on_161B (-| |-) ]  --------------------  HV_on_161B -( )-

Room 161B:  Power ON state.  True when it is possible for any 4160V or 480V outlet to be energized.
13  [ HV_on_161B (-| |-)  OR  Safe_Plug161 (-|/|-, On) ]  --------------------  PowerOnState_161B -( )-

Room 161B:  ACCESS VIOLATION.  Power on with someone in the room!
14  OccupiedState_161B (-| |-, Off)  HV_on_161B (-| |-)  --------------------  AccessViolationHV_161B -( )-

Room 161B:  ACCESS VIOLATION.  Power on with someone in the room!
15  OccupiedState_161B (-| |-, Off)  Safe_Plug161 (-|/|-, On)  --------------------  AccessViolation480_161B -( )-

16  Room_171
ROOM171: 4160 V disconnect switches.
17  HVSwitch3 (-| |-, On)  HVSwitch4 (-| |-, On)  --------------------  Safe_HV_171 -( )-

ROOM171: 480 V Outlet Off Check
18  Plug_4 (-| |-, On)  Plug_5 (-| |-, On)  Plug_IL_3_4 (-| |-, On)  --------------------  Safe_Plug171 -( )-

All manual disconnects reported open/safe/off by sensor inputs.
19  Safe_HV_171 (-| |-)  Safe_Plug171 (-| |-)  --------------------  Disconnects_open_171 -( )-

Occupied State:  Reset with door closed transition.
20  OccupiedState_171 (-| |-)  Door_171 (-|P|-)  --------------------  OccupiedState_171 -(R)-
```

DURIP State Control_Debugger: MasterCarme: Ladder_Program    9/18/2006-5:35:29 PM    Page 3



Occupied State: Latch true with any change of disconnect status, or with door open.

ROOM171: HV power A ON

ROOM171: HV power B ON

Room 171: Any HV power on.

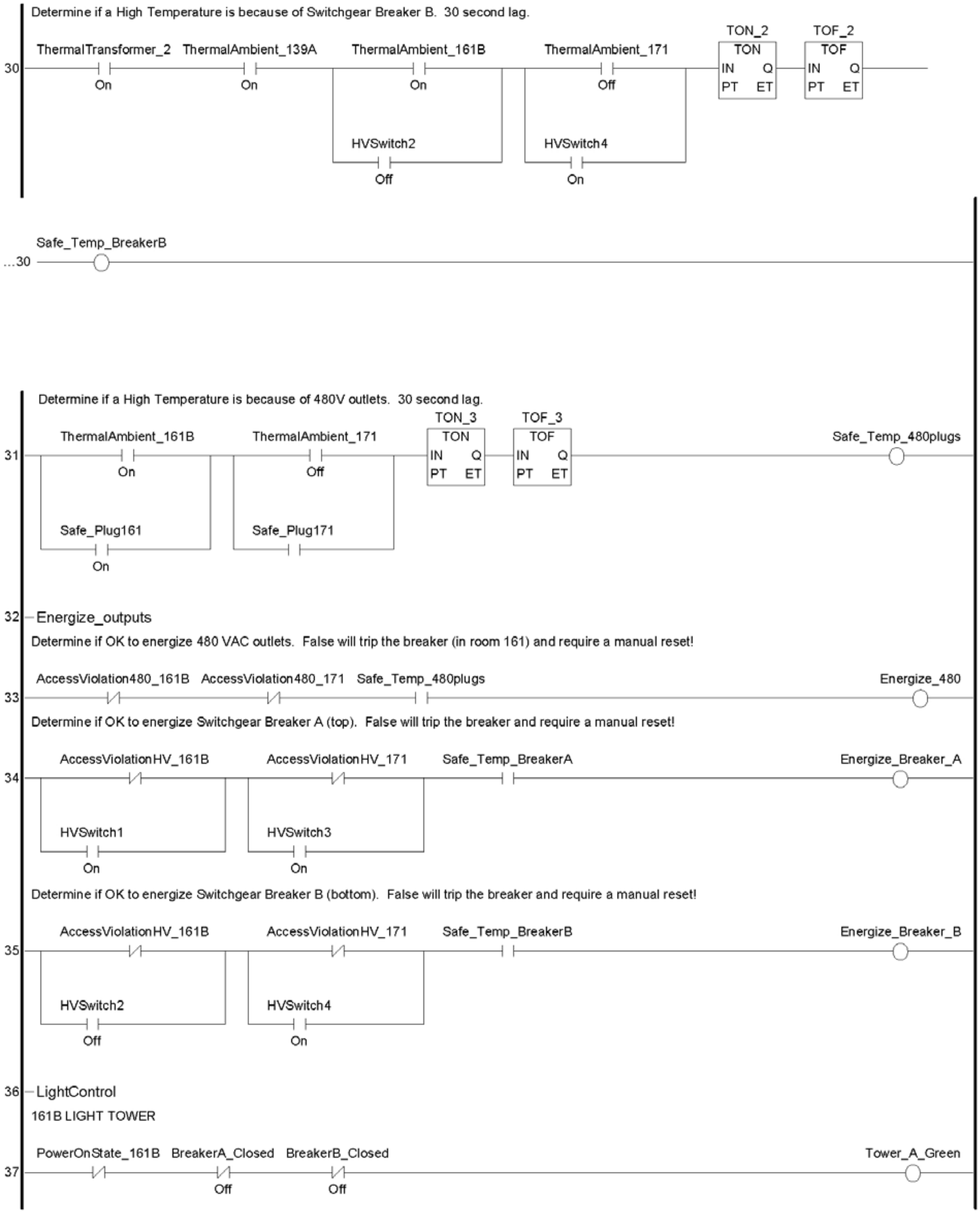
Room 171: Power ON state. True when it is possible for any 4160V or 480V outlet to be energized.

Room 171: ACCESS VIOLATION. Power on with someone in the room!

Room 171: ACCESS VIOLATION. Power on with someone in the room!

Determine if a High Temperature is because of Switchgear Breaker A. 30 second lag.

DURIP State Control_Debugger: MasterCarme: Ladder_Program 9/18/2006-5:35:29 PM Page 4



Determine if a High Temperature is because of Switchgear Breaker B. 30 second lag.

Determine if a High Temperature is because of 480V outlets. 30 second lag.

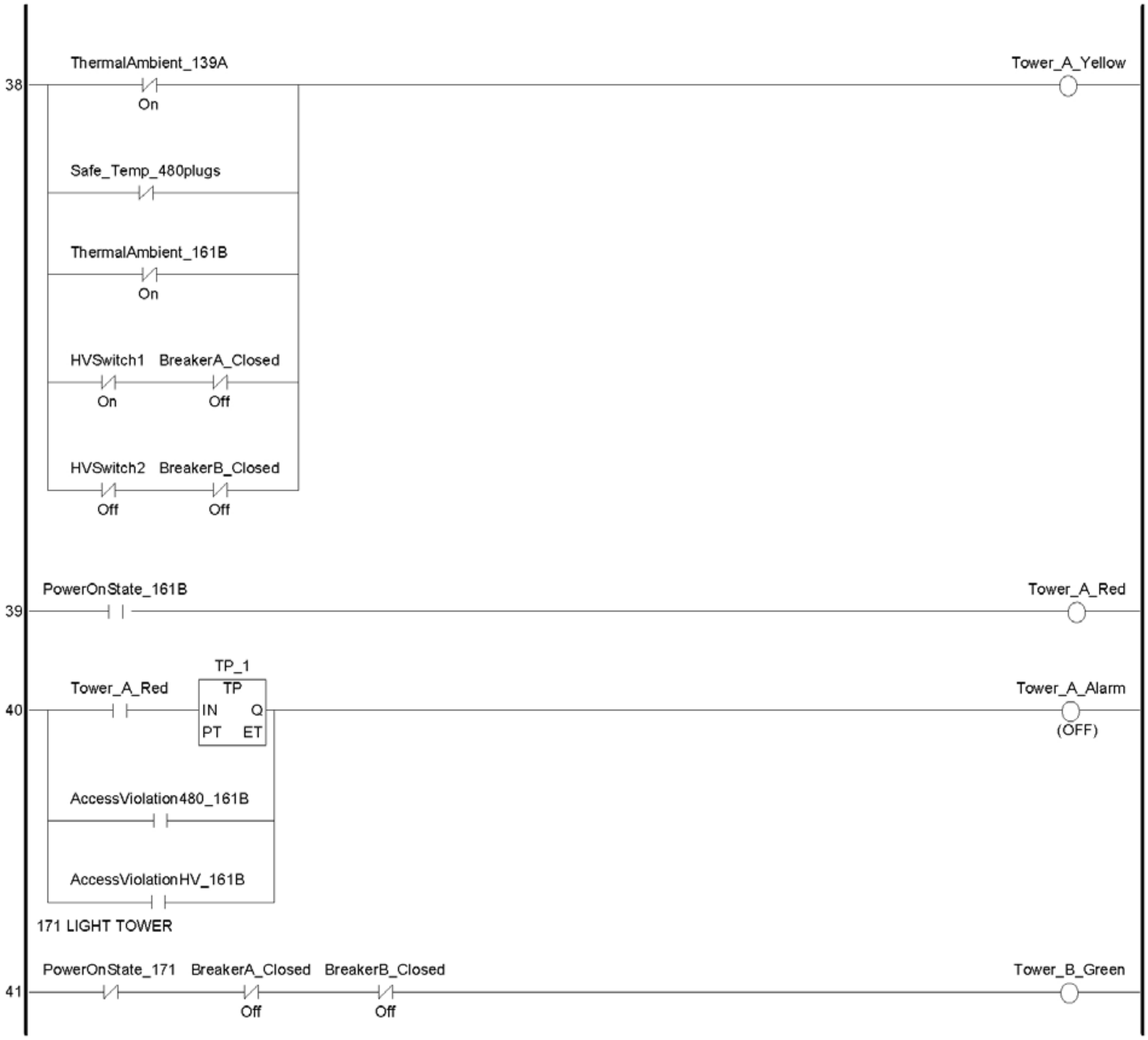
Determine if OK to energize 480 VAC outlets. False will trip the breaker (in room 161) and require a manual reset!

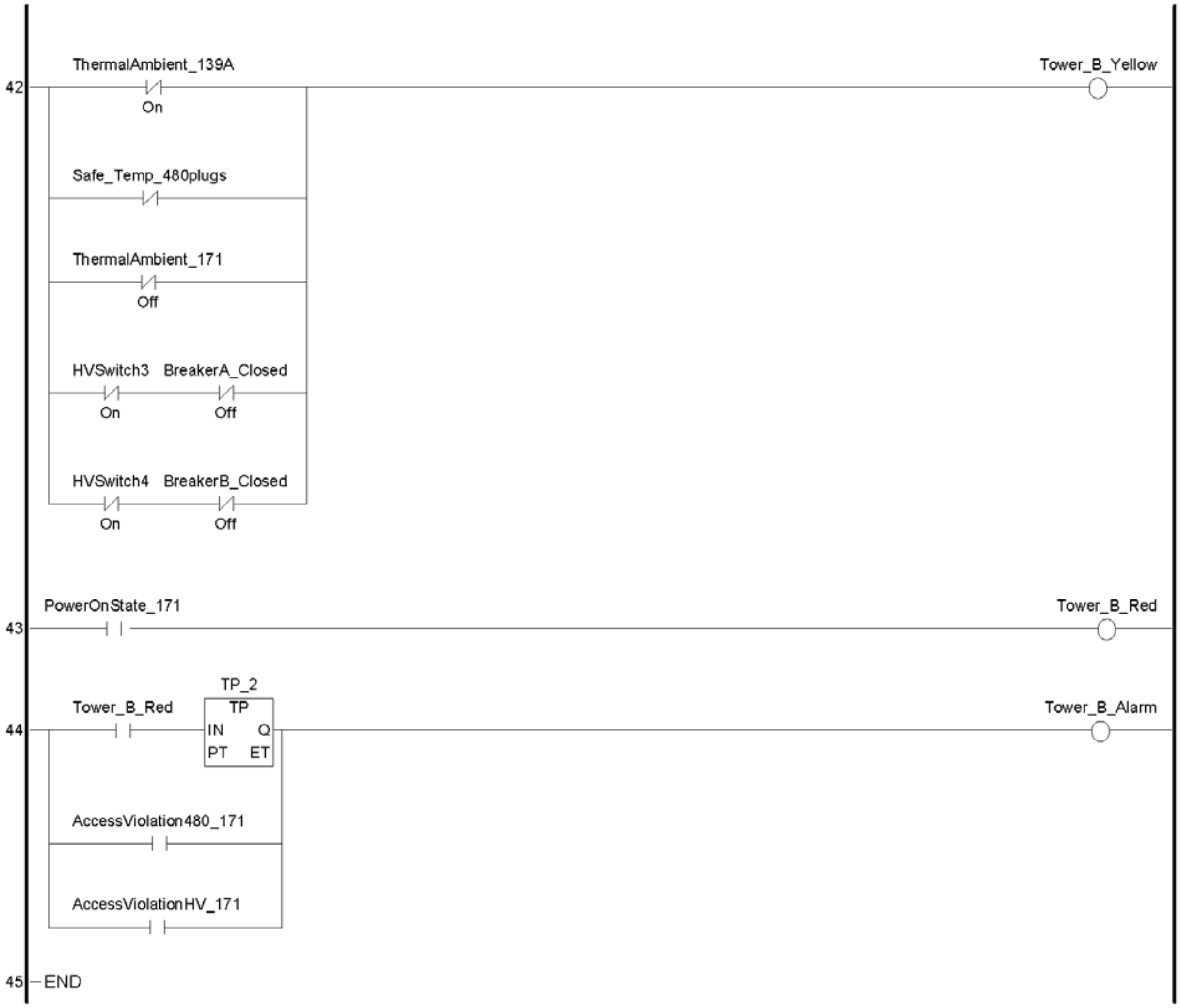
Determine if OK to energize Switchgear Breaker A (top). False will trip the breaker and require a manual reset!

Determine if OK to energize Switchgear Breaker B (bottom). False will trip the breaker and require a manual reset!

161B LIGHT TOWER

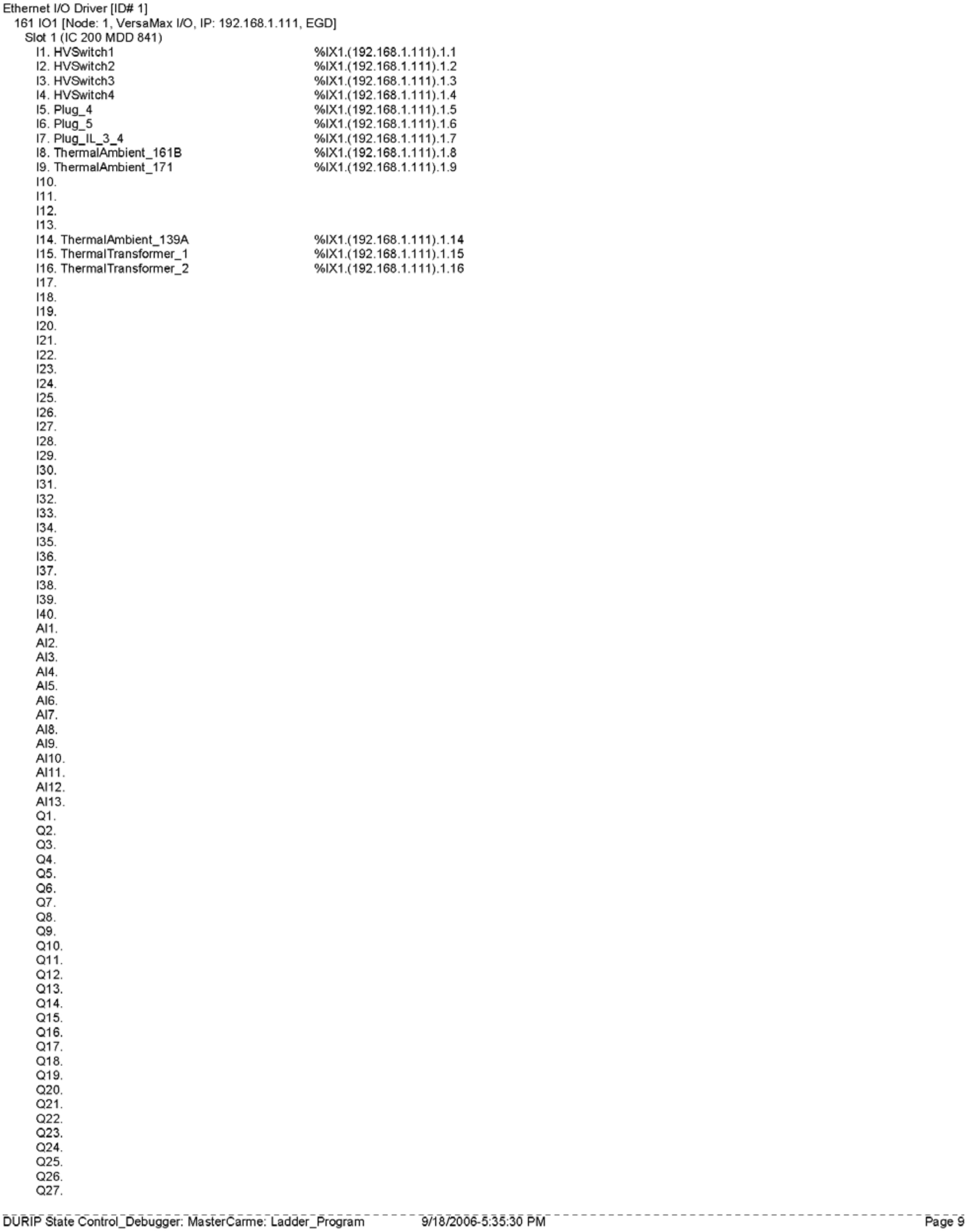
DURIP State Control_Debugger: MasterCarme: Ladder_Program 9/18/2006-5:35:29 PM Page 6


38
ThermalAmbient_139A
On
Tower_A_Yellow
Safe_Temp_480plugs
ThermalAmbient_161B
On
HVSwitch1
BreakerA_Closed
On
Off
HVSwitch2
BreakerB_Closed
Off
Off
39
PowerOnState_161B
Tower_A_Red
40
TP_1
Tower_A_Red
TP
IN
Q
PT
ET
Tower_A_Alarm
(OFF)
AccessViolation480_161B
AccessViolationHV_161B
171 LIGHT TOWER
41
PowerOnState_171
BreakerA_Closed
BreakerB_Closed
Tower_B_Green
Off
Off



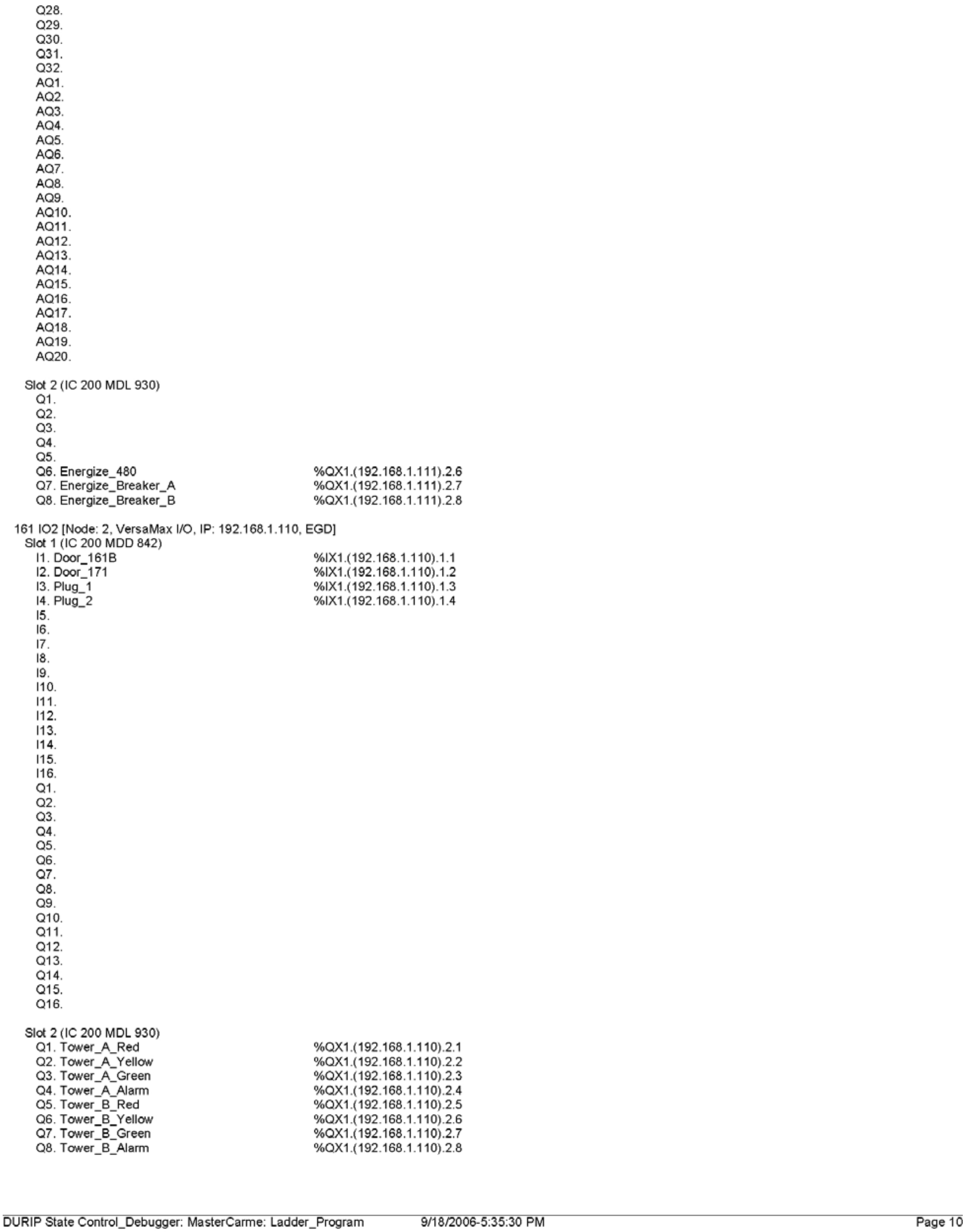
DURIP State Control_Debugger: MasterCarme: Ladder_Program 9/18/2006-5:35:29 PM Page 8

## 6.2 I/O listing



```
Ethernet I/O Driver [ID# 1]
  161 IO1 [Node: 1, VersaMax I/O, IP: 192.168.1.111, EGD]
    Slot 1 (IC 200 MDD 841)
      I1. HVSwitch1                       %IX1.(192.168.1.111).1.1
      I2. HVSwitch2                       %IX1.(192.168.1.111).1.2
      I3. HVSwitch3                       %IX1.(192.168.1.111).1.3
      I4. HVSwitch4                       %IX1.(192.168.1.111).1.4
      I5. Plug_4                          %IX1.(192.168.1.111).1.5
      I6. Plug_5                          %IX1.(192.168.1.111).1.6
      I7. Plug_IL_3_4                     %IX1.(192.168.1.111).1.7
      I8. ThermalAmbient_161B             %IX1.(192.168.1.111).1.8
      I9. ThermalAmbient_171              %IX1.(192.168.1.111).1.9
      I10.
      I11.
      I12.
      I13.
      I14. ThermalAmbient_139A            %IX1.(192.168.1.111).1.14
      I15. ThermalTransformer_1           %IX1.(192.168.1.111).1.15
      I16. ThermalTransformer_2           %IX1.(192.168.1.111).1.16
      I17.
      I18.
      I19.
      I20.
      I21.
      I22.
      I23.
      I24.
      I25.
      I26.
      I27.
      I28.
      I29.
      I30.
      I31.
      I32.
      I33.
      I34.
      I35.
      I36.
      I37.
      I38.
      I39.
      I40.
      AI1.
      AI2.
      AI3.
      AI4.
      AI5.
      AI6.
      AI7.
      AI8.
      AI9.
      AI10.
      AI11.
      AI12.
      AI13.
      Q1.
      Q2.
      Q3.
      Q4.
      Q5.
      Q6.
      Q7.
      Q8.
      Q9.
      Q10.
      Q11.
      Q12.
      Q13.
      Q14.
      Q15.
      Q16.
      Q17.
      Q18.
      Q19.
      Q20.
      Q21.
      Q22.
      Q23.
      Q24.
      Q25.
      Q26.
      Q27.
```

DURIP State Control_Debugger: MasterCarme: Ladder_Program 9/18/2006-5:35:30 PM Page 9



```
      Q28.
      Q29.
      Q30.
      Q31.
      Q32.
      AQ1.
      AQ2.
      AQ3.
      AQ4.
      AQ5.
      AQ6.
      AQ7.
      AQ8.
      AQ9.
      AQ10.
      AQ11.
      AQ12.
      AQ13.
      AQ14.
      AQ15.
      AQ16.
      AQ17.
      AQ18.
      AQ19.
      AQ20.

    Slot 2 (IC 200 MDL 930)
      Q1.
      Q2.
      Q3.
      Q4.
      Q5.
      Q6. Energize_480                   %QX1.(192.168.1.111).2.6
      Q7. Energize_Breaker_A             %QX1.(192.168.1.111).2.7
      Q8. Energize_Breaker_B             %QX1.(192.168.1.111).2.8

  161 IO2 [Node: 2, VersaMax I/O, IP: 192.168.1.110, EGD]
    Slot 1 (IC 200 MDD 842)
      I1. Door_161B                      %IX1.(192.168.1.110).1.1
      I2. Door_171                       %IX1.(192.168.1.110).1.2
      I3. Plug_1                         %IX1.(192.168.1.110).1.3
      I4. Plug_2                         %IX1.(192.168.1.110).1.4
      I5.
      I6.
      I7.
      I8.
      I9.
      I10.
      I11.
      I12.
      I13.
      I14.
      I15.
      I16.
      Q1.
      Q2.
      Q3.
      Q4.
      Q5.
      Q6.
      Q7.
      Q8.
      Q9.
      Q10.
      Q11.
      Q12.
      Q13.
      Q14.
      Q15.
      Q16.

    Slot 2 (IC 200 MDL 930)
      Q1. Tower_A_Red                    %QX1.(192.168.1.110).2.1
      Q2. Tower_A_Yellow                 %QX1.(192.168.1.110).2.2
      Q3. Tower_A_Green                  %QX1.(192.168.1.110).2.3
      Q4. Tower_A_Alarm                  %QX1.(192.168.1.110).2.4
      Q5. Tower_B_Red                    %QX1.(192.168.1.110).2.5
      Q6. Tower_B_Yellow                 %QX1.(192.168.1.110).2.6
      Q7. Tower_B_Green                  %QX1.(192.168.1.110).2.7
      Q8. Tower_B_Alarm                  %QX1.(192.168.1.110).2.8
```

## 6.3 Data transfer script

Moves data from the switchgear (serial ModBus protocol devices), to an internal variable as a background process. This frees the main program from waiting for a response to a request for data from a ModBus device. This runs every second.

```
'------------------------------------
' Script Created: Aug 30, 2006
'
' Description:
'
'------------------------------------
Internal_BreakerA_fdbk := PLC_DINT1
Internal_BreakerB_fdbk := PLC_DINT2
```

# Appendix B - VT High-Power Lab User's Guide

## VT High-Power Lab

## User's Guide

**Author: Dhane Ross and William Gatune**
**Revisions: Clint Perdue, 1.1**
**Date: 16 September 2006**

## B-1 Introduction

The High-Power Lab facility is designed to supply loads in excess of 250 kVA continuous power. The lab is controlled using GE-Fanuc Proficy (formerly Cimplicity) Machine Edition, an industrial supervisory control and data acquisition (SCADA) software package. This system consists of Proficy-Control, a soft-PLC (a programmable logic controller running on a generic PC instead of special hardware) for logic functions, and Proficy-View, a graphical operator screen engine. Both of these run on PCs in the lab.

The control system applications logic is designed to determine whether or not it is safe for the lab to be energized, and to shut down the power if there is a fault or if a person enters an energized area. This software is connected to the lab space instrumentation using a distributed network of I/O modules that communicate via Ethernet.

The purpose of this guide is, first, to document the specific configuration of lab equipment that is unique to this project and point users toward the pertinent OEM documentation. Copies of all the relevant documentation have been assembled into a project directory; ask the lab administrator for access. Secondly, this guide will help users configure the system for their needs.

If a user simply requires power, the system can operate as is. The lab administrator will enforce the safety procedures and checklists and can explain the default operator environment.

In many cases, users will wish to add custom interlocks to the system. The system has provisions for this, and we will describe them here. The control also supports custom user programming for referencing, data collection, and other tasks as desired.

## B-2 Communications

The following section details how the various elements of the lab communicate. Figure B-1 shows topology including Ethernet, serial networks, and hardwired discrete signals.

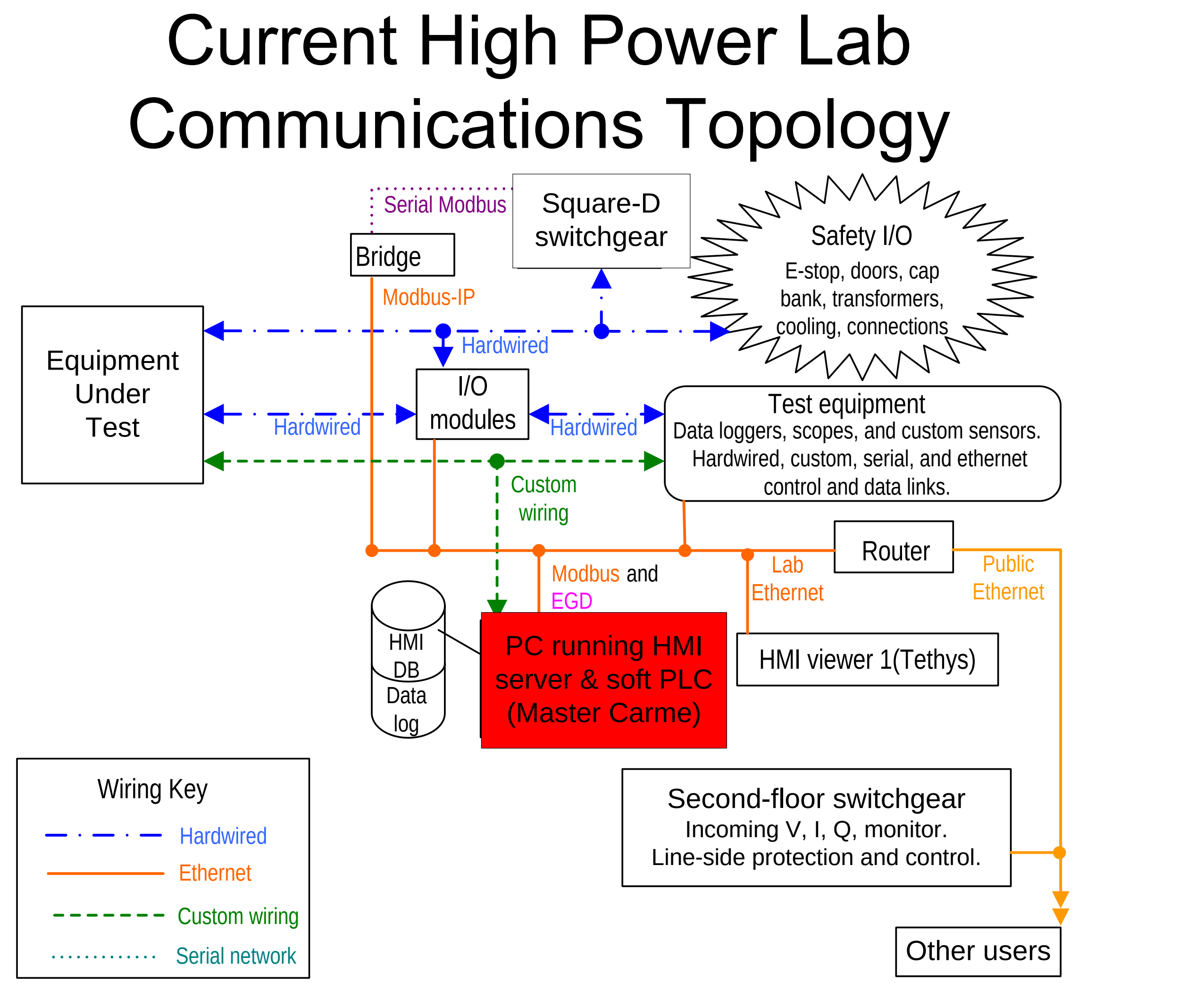


**Figure B-1; Communications topology. Master computer, Carme, is in red.**

### *B-2.1 Ethernet Devices*

We employ an Ethernet network of hardwired devices which communicate through a router, and also a hub for devices within the Switchgear Room, room 131. Our private network has all been wired with red CAT-5e Ethernet cables – everything else in the area is blue. Below in Table B-1 is the network configuration information for the Ethernet devices.

**Table B-1; Ethernet configuration data.**

| Device Name | Purpose | MAC Address | IP Address | *Physical Location | Additional Information |
|---|---|---|---|---|---|
| ROUTER - LAN side | Provide lab Ethernet. | 00-12-17-4c-da-08 | 192.168.1.1 | 161B- Near door. | Login:Admin<br>Pass:Admin |
| ROUTER - WAN side | Internet connection using cloned MAC. | - | 128.173.90.202 | - | Subnet Mask 255.255.252.0 |
| Carme | Soft PLC controller | 00-07-e9-7d-19-5e | 192.168.1.100 | 161b window | Registered developer Station |
| Tethys | Designated HMI | 00-07-e9-7d-61-cb | 192.168.1.102 | 161- Center bench | HMI node |
| 161_IO1_NIU | Cabinet Module Communications | 00-09-91-00-4d-6b | 192.168.1.111 | 161B- Far wall | VersaMax NIU |
| 161_IO2_NIU | Cabinet Module Communications | 00-09-91-00-4d-be | 192.168.1.110 | 161B- Near window | VersaMax NIU |
| 139_IO1_NIU | Cabinet Module Communications | 00-09-91-00-4d-de | 192.168.1.112 | 139- beside switchgear. | VersaMax NIU |
| “CCM Gateway” | Communicates with switchgear | 00-80-67-80-4b | 192.168.1.130 | 139_IO1 | Login:Administrator<br>Pass: Gateway |
| 139_IO1_HUB | Connects 131_IO1_NIU and “CCM Gateway” | N/A | N/A | 139_IO1 | |

***Important: Physical locations are subject to change. Be sure to update this list whenever any of these locations find a new temporary or permanent home. Red letters indicate a known temporary position.**

The Router has 8 cable ports, each with clear number of markings on the case. At present, 5 Ethernet cables are connected.

**Table B-2; Router port assignments.**

| Physical Port Number | Device/Connecting-Location | Additional Information |
|---|---|---|
| 1 | 161_IO1 | 1 Stripe on Cable |
| 2 | 161_IO2 | 2 Stripes on Cable |
| 3 | 139_IO1 | 3 Stripes on Cable |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | Tethys | HMI viewnode. |
| 8 | Carme | Master computer. Runs logic, HMI server. |

### *B-2.2 Modbus serial network*

The lab switchgear in room 139 uses Modbus, a serial network designed to imitate hardwired relay logic, to communicate between the circuit-breakers. We can get this data into the control and HMI via a bridge device located in 139IO1. Modbus is a serial, daisy-chain network. New drops are added by putting another unit in parallel with the existing ones on the four communications wires and relocating an appropriate termination resistor if the new drop is on the end of the chain. The wires are color-coded, just match the pattern. Addressing this data is covered in the switchgear section.

### *B-2.3 Wiring Cabinets*

As of Rev. 1.1, the I/O modules are located in three wiring cabinets: 161IO1, 161IO2, and 139IO1. Note that the cabinets are named according to their physical location and function. For example, 161IO1 is located in room 161, is an I/O cabinet (with electronics, versus a junction box, JB, with only wire connections), and is box number 1 in that area.

### B-2.3.1 Functions of Cabinets

**Table B-3; Cabinet functionality.**

| **Cabinet Designation** | **Physical Location** | **Purpose** | **Devices Controlled/Connected** | **Internal Hardware** |
|---|---|---|---|---|
| 161_IO1 | Far wall of 161B | Acts as a pull-box for communications with room 171, the Switchgear, the high voltage switches, and any devices on the far side of the room | HV Switch: 1,2,3,4<br>480 Outlets: 3-4,5,4 (all 171)<br>Thermal Sensors 171,161B ambient<br>temporary functions | VersaMax NIU<br>Power Supply<br>IC200MDD841<br>IC200MDL930 |
| 161_IO2 | Beside window in 161B | Monitors Doors, 480 outlets in 161B. Controls both light trees. | Door 161B and 171, 480v Outlets 1, 2. | VersaMax NIU<br>Power Supply<br>IC200MDD842<br>IC200MDL930 |
| 161_JB1 | Beside door in 161 | Mounting and connections for light-tree. | Room 161B light tree. | None |
| 171_JB1 | On side of HVSwitch4 enclosure. | Brings together the four 480 Switches and thermal sensor into a larger multiconductor cable | 480 Outlets: 3-4,5,4 (all 171)<br>Thermal Sensor 171 ambient | None |
| 139_IO1 | Adjacent to the HV Fuses and beside the switchgear on the wall. | To control communications with the switchgear and gather information concerning 139A | SquareD Gateway-switchgear CCM<br>Thermal Sensors: Transformer 1, 2, 139 ambient. | VersaMax NIU<br>Power Supply<br>IC200MDD842<br>IC200MDL930<br>Power Logic EGX100 |

### B-2.3.2 Cabinet Construction

In each cabinet there are devices that allow access to the surrounding environment. The cabinets themselves are NEMA-1 steel enclosures, grounded to provide shielding from radiated EMI in the lab bays. The general layout of the I/O cabinets is as seen in Figure B-2, with a real example in Figure B-3. The following appear as a series of black boxes:

- Network Interface Unit (“NIU”): The network communication module uses an Ethernet connection to distribute data from subordinate modules and essentially interface with them. It must be programmed to have a specific IP address to operate properly.
- Power Supply: The power supply is connected to the surface of the NIU and appears as a thinner black box with power wiring and ground wires connecting to it.
- Modules: Each module has different specific functions according to the specifications it was purchased for. The modules must sit inside of a base or "carrier" which connects to the NIU through a backplane, and provides wire termination points for external devices. Carriers can be configured to connect to any module type by way of turn dials. The modules present are an isolated relay output module (marked red) and a 24 VDC Discrete I/O module (yellow).

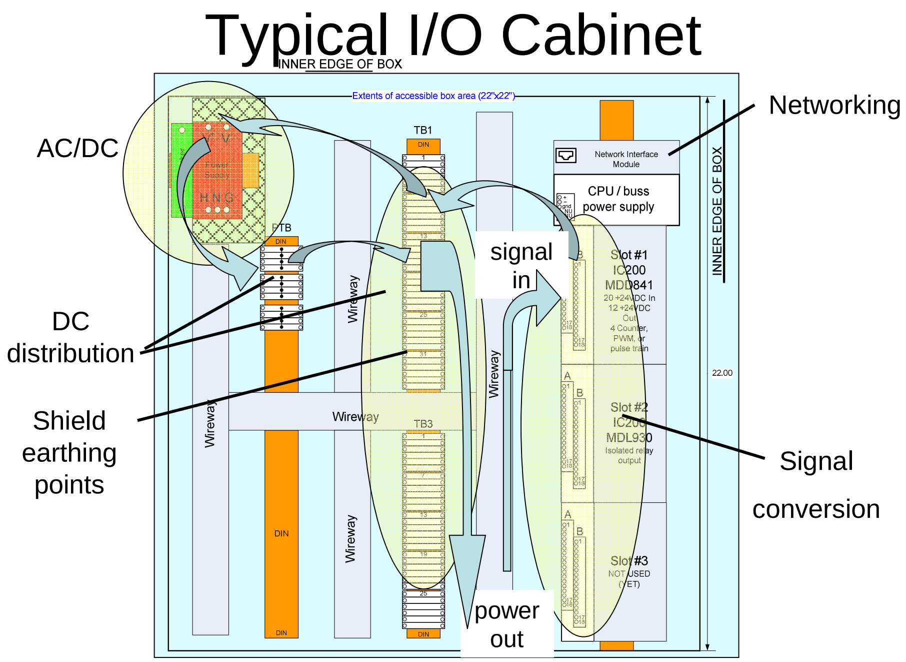


**Figure B-2; Typical I/O box with signal path shown.**

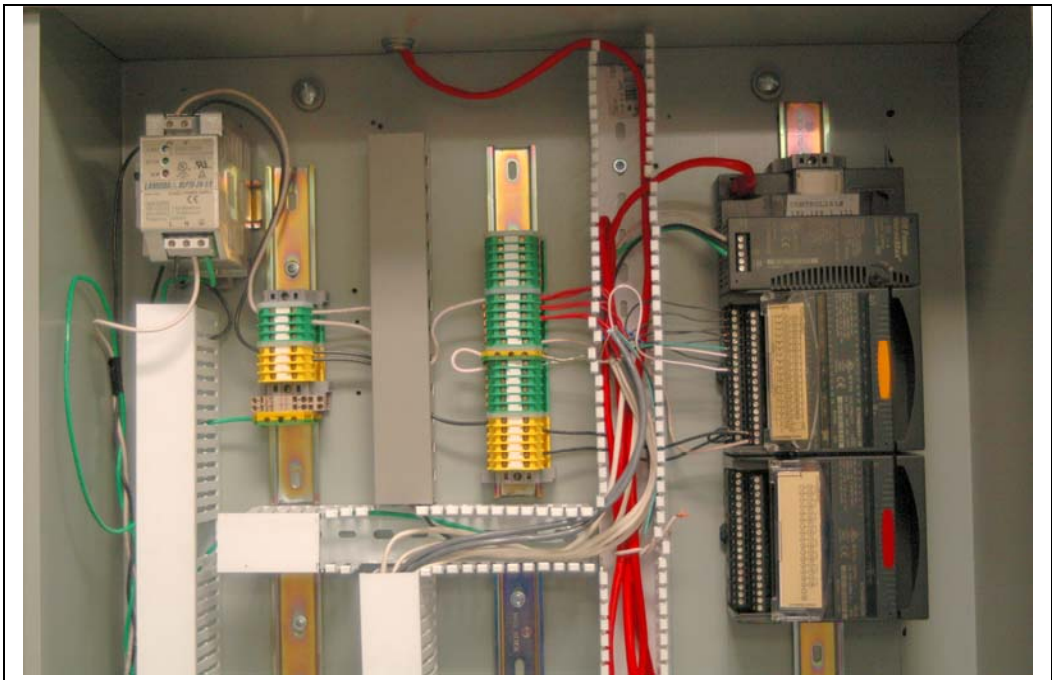


**Figure B-3; typical I/O cabinet**

*Currently Installed Modules:*

- Power: 24vdc Power Supply, IC200PWR002C
- NIU: VersaMax Ethernet NIU, IC200EBI001

- Carrier/Base: Box-Style I/0 Carrier, IC200CHS002
- Mixed I/O 24VDC Discrete/High-Speed Counter Module, IC200MDD841
- Mixed I/O24 VDC Discrete Module, IC200MDD842
- Relay Output Module, IC200MDL930

Further details about these items can be found in the VersaMax® Modules, Power Supplies, and Carriers User's Manual – GFK-1504K, March 2003, GE Fanuc Automation, Programmable Control Products. An electronic copy of this document is located in the DURIP system directory.

### *B-2.4 Wiring*

Note that field wiring is shielded. Power to the field device comes from the terminal strip to the left of the modules, as close as possible to where the return signal conductor is routed. The cable shield wire is tied to earth ground to the left side as well, keeping the shield intact up to where the other conductors part.

The best resource for good wiring practice we have available is the instructions GE publishes for installing their drives and control systems, Installation Guidance For Innovation Series™ Drive Systems - GEH-6380, © 1999 General Electric Company, USA. Chapters 4,5,6,7, and 9 are particularly useful. Some general points to keep in mind:

- Keep power and signal wires as far apart as possible and cross them only at 90 degree angles, and never wrap one around the other.

- Cable shields must be tied to ground, but at one point only – usually the end near the I/O module is most convenient.

- Conductors must be sized for the expected current they will carry, so wire for relays and solenoids will be larger than for sensors. The AWG wire size is the log of the resistive power lost for a given current. Three sizes larger is 3dB more loss, or twice the power (so then twice the resistance). If you get too much voltage drop due to wire resistance in a #16 wire, #13 will cut the drop in half.

### *B-2.5 Sensors*

Most of the sensors in the lab have a pair of complimentary contacts, one set is open when the sensor is in its default/quiescent/"normal" state, and the other is closed. We have elected to use the set which is normally open, so that the sensor, when actuated, indicates by a closed circuit that it detects a "safe" condition. If the input circuit should fail at any point, a fault will be indicated, making the system fail-safe. Any new system interlocks should be set up the same way. Table B-4 below lists the sensor types and their "safe" outputs.

**Table B-4; Sensor outputs.**

| **Sensor** | **State** | **Output** |
|---|---|---|
| High Voltage Switch | Off | 1 |
| 480 Voltage Switch | Off | 1 |
| Door Sensor | Closed | 1 |
| Ambient Thermal Sensors | Cool - ok | 1 |
| Transformer Sensors | Cool - ok | 1 |

## B-3 Switchgear

One of the most important components of the system is the switchgear located in room 139. There are two units, one on each distribution leg feeding the lab bays, see Figure B-4. Both are Square-D Power-Zone® 4 models, with MICROLOGIC® 5.0P electronic trip units. We refer to them improperly as the high-voltage switchgear even though they operate at only 480VAC because they control the current that becomes the 4160VAC feed to the labs (and 4160 is really classified as medium voltage by utility people).

In brief, these are circuit breakers with built in power analyzers. They can determine up to the 40th harmonic of power system complex voltages and currents and publish this data, via ModBus and network bridge, to the control system. The breakers themselves can be controlled remotely, either by network commands or through traditional discrete hardware inputs.

We have disabled the remote-close capability (closing is a manual-only, administrator prerogative), but have connected their remote-trip request, discrete hardware input to a set of normally-closed relay contacts, that are held open by an "energize-OK" output from the control. If this output goes away for any reason, including control failure, the relay will deenergize and trip the breaker – see the E-stop wiring elementaries.

Each trip circuit pulls over 9 amps at 24 VDC for a few milliseconds to actually trip the main circuit breaker. We have placed ride-through capacitors in the switchgear upper equipment bay to buffer this load.

There is a similar power analyzer on the second floor feeder to the lab, and its data is also available to the control system, but has not yet been utilized.

There is also a remote-trippable circuit breaker located in room 161 for the 480 VAC feed to the lab bays. The trip control for this is similar in principle to that of the other units, but it does not have a power analyzer, and students are allowed to close it for themselves. The rest of this section deals exclusively with the previous units unless specifically stated otherwise.

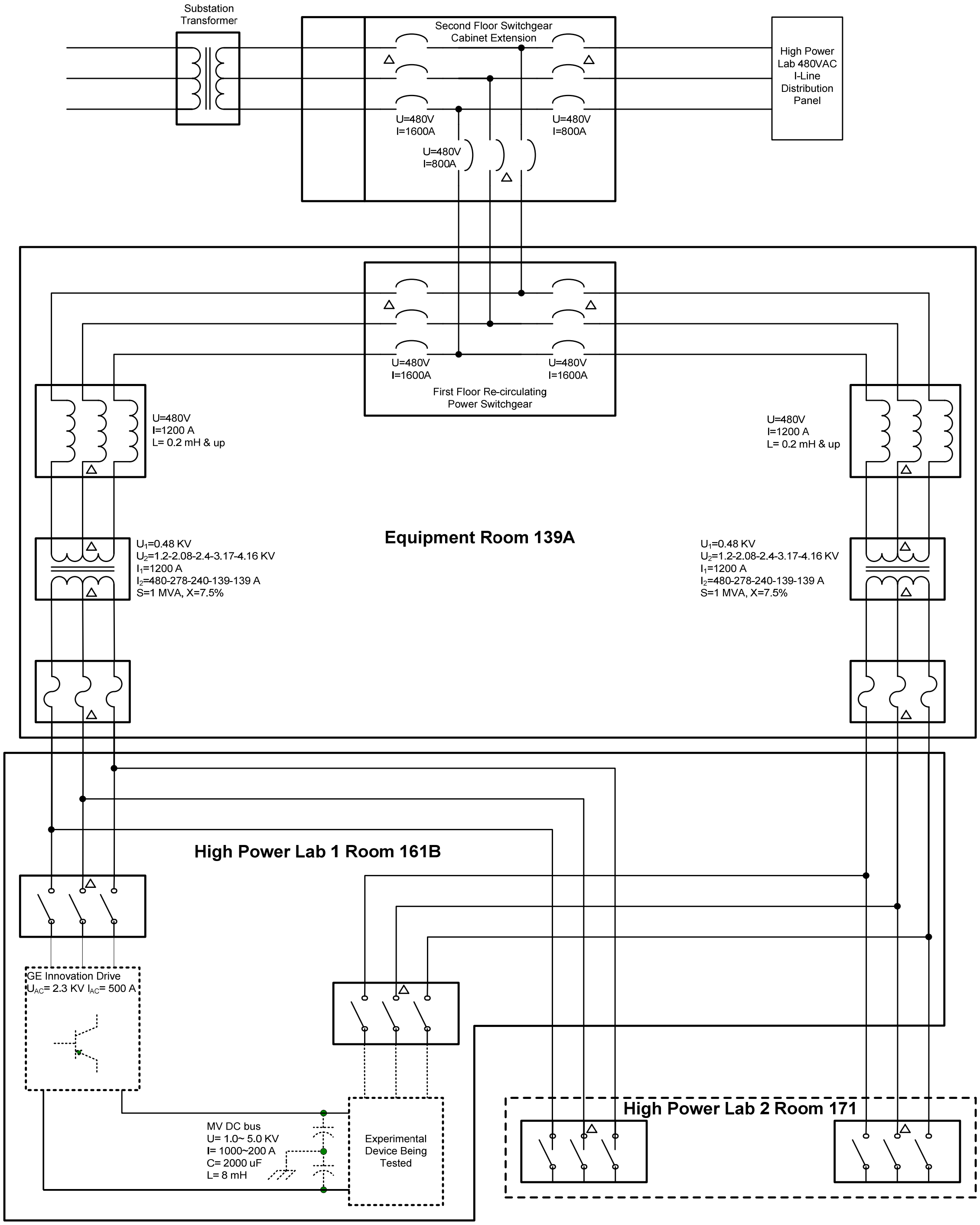

Substation Transformer
Second Floor Switchgear Cabinet Extension
U=480V I=1600A
U=480V I=800A
U=480V I=800A
High Power Lab 480VAC I-Line Distribution Panel
U=480V I=1600A
U=480V I=1600A
First Floor Re-circulating Power Switchgear
U=480V I=1200 A L= 0.2 mH & up
U=480V I=1200 A L= 0.2 mH & up
Equipment Room 139A
U₁=0.48 KV U₂=1.2-2.08-2.4-3.17-4.16 KV I₁=1200 A I₂=480-278-240-139-139 A S=1 MVA, X=7.5%
U₁=0.48 KV U₂=1.2-2.08-2.4-3.17-4.16 KV I₁=1200 A I₂=480-278-240-139-139 A S=1 MVA, X=7.5%
High Power Lab 1 Room 161B
GE Innovation Drive U_AC= 2.3 KV I_AC= 500 A
MV DC bus U= 1.0~ 5.0 KV I= 1000~200 A C= 2000 uF L= 8 mH
Experimental Device Being Tested
High Power Lab 2 Room 171

**Figure B-4; power distribution elementary.**

## B-3.1 Reading Switchgear Register Information

Each trip unit has an internal, controlling computer that keeps vital information about the status of the system and the power passing through it. Getting to this data is a bit of a basic programming exercise, requiring reading and interpreting specific registers. Addresses and further instructions can be found in "MICROLOGIC® 5.0P and 6.0P Electronic Trip Units, Instruction Bulletin 48049-137-02", Appendix-C, table-17, pp. 69-75.

Recall that the trip units communicate via ModBus-serial and so has its own drop on that network, but share a single Ethernet IP address when communicating with the control system. Therefore, addressing trip unit registers requires using the proper unit-ID, see below.

**Table B-5; Switchgear identification.**

| Unit ID | Breaker name/position | Associated transformer | Associated HV disconnects |
|---|---|---|---|
| 45 | A / top | 1 | 1, 3 |
| 47 | B / bottom | 2 | 2, 4 |

### B-3.1.1 Register Configuration

When configuring the master computer to read the registers of the switchgear, it is important to note the following:

- **Note 1 on Switchgear Offset:** The slot information configures the offset and the length of the register in *bytes* not registers. This is important because the slot information displays the entered Register Offset in terms of the total bytes. Example: Register offset = 32. Register Size = 16-Bits. The reported offset next to the slot number will be 64 because 32 registers corresponds to 32 registers at 2 bytes (8+8 bits) each.

- **Note 2 on Switchgear Offset:** The register offset declares the shift in read registers. The first terminal to be read from this offset starts one register afterwards. So, if you are attempting to read register 671, you need to set the register offset to 670. Possibly because the register offset only provides the base, and the register count starts after that point.

### B-3.1.2 Method 1: Reading generic source data directly into the program.

**WARNING** what follows is an example based on our first efforts to get switchgear data into the control program. It works, but has serious consequences for execution time. This method is no longer recommended for communicating with the switchgear, but is included because it may be useful for connecting to some other device. The reasoning will be made clear in the next section.

**Example:** Showing detailed instructions of how to read the time and date from the switchgear. Time is a good, dynamic, easily verified signal to use for diagnostics.

a. From inside the control project development software, double click at the Control I/O drivers then add a node.

b. At the properties window the node is configured according to the setup we have. This is shown in the next steps.

c. The node type is a Generic device since we are communicating to the switchgear. The switchgear communicates directly to the computer without using a control box and hence it does not use the VersaMax NIU.

d. Select Modbus/TCP for the protocol because that is the protocol for the switchgear. All the other devices are made by GE and hence use GE’s unique protocol known as EGD.

e. The IP address of the switchgear has been configured at the router settings to be 192.168.1.130 .

f. Enter 1 for the number of slots. The number of slots designates the number of modules. In our case, since it’s the switchgear, the slot is used to set a base to start reading the registers, and we only want one base, so enter 1.

g. Message time-out is explained later in the "Operating Cimplicity" section, for now just input 1000.

h. Reconnect time-out is explained later in the operating simplicity section, for now just input 2000.

i. Time between reconnect is explained later in the operating simplicity section, for now just input 1000.

j. Input 45 for unit ID. This is how the switchgear has just been configured. Unit 45 represents breaker A of the switchgear and unit 47 represents breaker B.

k. Enter 678 for input register offset. Register 679 is the one to be read since it stores the date and time information. See note 2 above to know why we enter the register offset to be 678 instead of 679.

l. Leave output register offset to be zero, because we are reading data not writing.

m. The slot now has to be configured. Double click on it. Note that the offset on the slot will seem to be "off" but note 1 above explains.

n. At the properties make sure it's on analog input.

o. Click on the tab for module properties. Number of terminals corresponds to the number of registers (4 in this case for the date and time). Since the last register is showing milliseconds, and that is in no ones particular interest just enter 3.

p. The switchgear is composed of 16-bit registers therefore select 16-bit.

q. Now you are ready to run the program. Refer to Table 3-2, showing key registers in order to understand how to read the data coming from the registers.

### B-3.1.3 Method 2: Reading serial-source data via a buffer with a script.

In brief, what GE Tech Support recommends is periodically transferring serial-sourced data to/from a buffer variable using an "applications script," which runs periodically in the background. The ladder (or other process control) logic will use the current value of the buffer instead of waiting for an update. We have tried this, and it seems to work, but we do not have a final version as of this revision. The conversation with GE is below:

> Q: Scan time for the ladder program is hundreds of milliseconds - why?
>
> A: The issue is the Ethernet to Modubs bridge.
>
> The Modbus TCP/IP communication is SYNCHRONOUS and is a request / response protocol. When you put serial on the I/O system, you make the scan times HUGE [the logic scan waits for the message response CLP].
>
> Use the View Modbus driver to read this data and pass it to the controller.
>
> GE Fanuc Tech Support

Q: Is the the View Modbus driver just another element of PC-control that I can turn on, or do we need to run some other program (view runtime?) on the same PC (or another PC, i.e., a viewnode)?

A: The View modbus driver is part of the Control IO setup that you have configured. It would be running with the PC Control side of the software

GE Fanuc Tech Support

Q: Ok, we are getting somewhere. We found the help for setting up PLC access with a Modicon TCP/IP driver, and think it is configured with the proper modbus drop and address.... We also have a signal which was previously coming from a generic Ethernet driver (under control I/O drivers) associated with the new driver. However, we get Error 324 at build time, it seems that the ladder logic can not receive data directly from a PLC source.

Question then, how can we get the PLC source data into the ladder program? I gather that I may have to set up a script running in the background, or set internal values as an action on a screen object.

A: You are not able to use a variable connected to the PLC Access Variable in the LDPC portion of the product.

You are correct about being able to use a script to bring the PLC Access variables into the LDPC logic. You can create a application script that will periodically copy the PLC Access variables into an internal variable. You will then be able to use these internal variables in your ladder logic.

GE Fanuc Tech Support

Q: About the script - OK, that is what I thought. Now how do I do it? Can you please send me an example script and tell me how to get it to run. It would be best if it did not depend on a particular screen or object being visible.

A: The best way to do the script is to create a application script that runs periodically. Application script will run regardless of which panel is currently being displayed.

In the script, you would do the assignments of the PLC Access variables to the internals variables

ie

```
InternalVar1 := PLCVar1
```

InternalVar2 := PLCVar2

You can setup the script to go other other way. Internal variables being using in the LDPC to the PLC Access variables

ie

PLCVar3 := InternalVar3
PLCVar4 := InternalVar4

The one problem with this is that the communications will only go one way. If you try and send variables back and forth between two variables, you will run into problems.

GE Fanuc Tech Support

Q: One final question; What is the recommended way to accomplish bi-directional communications?

A: If you are talking about bidirectional communications through scripting. It can be done, but there are a few things that you need to be careful about.

If the data only flows one way in some variables and the other way in the other variables. There shouldn't be a problem. You just have you script written like:

InternalVar1 := PLCVar1
PLCVar2 := InternalVar2

But if you want the data to go both ways on the same tag, there may be some problems. If the value on both sides change at the same time, you will need to determine which side to write from. You also don't want to continually write frmo one side to the other.

ie

InternalVar1 := PLCVar1

If there is a new value in InternalVar1, it will be overwritten by the previous value in PLCVar1. So you will have to determine when a value changes and only to write to the other side when it changes to avoid overwritting a new value on the other end.

GE Fanuc Tech Support

**Table B-6; Key registers in the switchgear.**

| Register Number | Number of Registers | Description | Additional Information | |
|---|---|---|---|---|
| 679 | 4 | Shows Date and time | Register1:<br>Byte1:Month (1-12)<br>Byte2:Day (1-31)<br>**Mask Bits 14 & 15 on the Month/Day register | Register2:<br>Byte1: Year(0-199) +1900<br>Byte2: Hour(0-23)<br>Register3:<br>Byte1: Minutes(0-59)<br>Byte2: Seconds(0-59) |
| 661 | 1 | Breaker Status | Bit 0: Position | On = Closed<br>Off = Open |
| 661 | 1 | Breaker Position | Bit8 = Disconnected<br>Bit9 = Connected<br>Bit10 = test position | |
| 661 | 1 | Breaker Trip Status | Bit 2: Tripped | On = tripped<br>Off = not tripped |

# B-4 Operating Cimplicity

## *B-4.1 Connection Setup*

For every device on the network, one must add that device to the Cimplicity Project- this includes the controller (this computer), the view node and every other module on the network.

Adding and Configuring Network Modules:

1) Right click in the project window on your project's name and "Add Target"
    a. From there you will be able to chose various devices to add including a windows station, PLC drivers, and Modules
2) To add a service Panel Module:
    a. After reaching "add target," click GE Fanuc Remote I/O-> VersaMax Ethernet
3) You must designate:
    a. The Power Supply Model
    b. Network Interface Unit Model
        i. After picking the module you must configure the network setup. Be careful when specifying information in the Produced and Consumed Exchange tabs.
            1. Network Tab-Specifies network connection settings.
                a. Designate the IP address
                b. Subnet Mask (255.255.255.0)
                c. Gateway IP (192.168.1.1)
                d. Mode:
                    i. EGD- For use with GE Devices Such as the NIU.
                    ii. ModBus TCP/IP if you use a generic device.
            2. Produced Exchange Tab- The *Produced Exchange* refers to the information that is *generated by the module* that will be exchanged with something else (the PLC).
                a. Exchange ID: Always remains 1
                b. Consumer Type: Designates how the device(s) that *consume* the *produced exchange* will receive the information.
                    i. IP Address: Set the IP address of the *computer that will accept the produced exchange.*
                    ii. Group ID: By specifying a group number you can allow information to be sent to all devices associated with that group.
                c. Producer Period(ms): How often the module will produce information.

3. Consumed Exchange Tab- The *Consumed Exchange* refers to the information that is *received by the module*.
    a. Exchange ID: One would guess it is the group from which the module receives its instructions from- including all devices that are part of the group. In this Lab, we only want the master computer to send instructions to the modules.
        i. Avoid using default values for anything.
    b. Producer ID: Refers to the *source* that the information the module consumes *originates from*. In this setup, it's the master computer's IP address.
        *i.* *Note* *yes…We know it doesn't seem right to designate the Master computer's IP address for two different information directions both times. That's the way it is*
    c. Group ID: Designate the Group that the produced information will be a part of.

c. To Add Subsequent Modules
    i. Add Carrier/Base
        1. IC200CHS002: I/O Carrier Box Style
            a. Chose a Module

### *B-4.2 Some Basic Procedures:*

## The interface

Adding Devices

Cimplicity has various functions that make it attractive: It is capable of taking in various signal types if it knows what type of data to accept, produce signals, produce "view nodes" and regulate other properties. Like the modules in the cabinets, the GUI of Cimplicity imitates the screw terminals of each module or even for virtual devices such as a modbus device.

To add a module, view the Control I/O Drivers section and Ethernet I/O sublisting. From the Ethernet IO screen, you made add a 'Node'. Each node is a different physical device, or in the case of a Modbus connection, a different section of registers.

## *Communication Checklist*

- Message Time-Out: During normal communication, this is the time the computer expects to receive a message reply before it decides that the device is no longer communicating properly.
- Reconnect Time-Out: The time allowed for the reconnecting device to reply to the reconnect request.
- Time Between Reconnect: The duration of the pause between trying to reconnect. The controller will always ask to reconnect.

### *B-4.3 Ladder Logic*

Ladder logic is an instruction language based on the flow of an electrical signal across various "rungs." Each rung is a row in which the signal begins from the left side and travels to the right side. The flow of logic is from top to bottom and executes each rung in that order. The logical flow is similar to assuming a basic current or signal that originates on the left, and the instructions you insert into the rung act like switches closing or opening the circuit. At the end of a rung, you may output the final value to a variable. In essence the ladder logic simulates a series of relay contacts which affect an output signal which could also be seen as another relay elsewhere.

To understand the instructions, knowledge of normally open and normally closed relays is required. The term ‘normally’ describes the state of the device while there is no power applied to relay. Below is a brief description of these operations.

> Normally Open (NO) contacts: The contact will remain open without any given power. When power (or a signal) is applied, the device closes.
>
> Normally Closed (NC) contact: The contact will remain closed without a signal or power and closes in the presence of a signal or power.

In the ladder logic program, the NO and NC contacts are represented by a capacitor-like symbol, and the same with a strike through it respectively. Regardless of what the actual sensor is, you may still chose either a NO or NC instruction in the ladder logic. For example: the door sensor physically is a NO switch. Assume the variable assigned to the sensor is called Door1. When placing an instruction in a rung of the ladder logic, despite Door1 originating from a NO sensor, I can still chose either a NO or NC instruction to place in the rung.

**It is important to understand that ladder logic is just a software language designed to *look* like physical relay logic. What you write does not necessarily exist physically. The instructions you place are generic, and the variable that controls them may or may not physically exist. Bottom line: *The ladder logic instructions do not model the physical switches associated with the controlling variable. The variables assigned to an instruction simply* <u>*control*</u> *that instruction that you place*. All that is important with these variables are their value and how it changes your instructions; you are not modeling the physical switches.

Each NO or NC contact changes states depending on the variable assigned to it. Variables in Cimplicity can take one of two states, although the actual labeling of the states may vary. For instance, the following values you may choose for a Boolean variable are all the same, but with different names: on/off, 1/0, true/false, I/O, Open/Closed. Regardless of which designation you chose, the NO or NC contact changes state so long as the variable associated with it changes to true, 1, I, on, or open as if the relays had power applied to them.

Traditionally, at the end of each rung an 'output' relay is placed so that whatever value the signal takes (true, on, 1, etc., whatever form you choose) may be outputted to another variable. You may place as many output relays in any location as you see fit however. You may place instructions in parallel by click-dragging an empty segment of rung and pulling the dotted line that comes from it across to the other side of one or more instructions.

<u>"AND"ing and "OR"ing</u>: Since ladder logic operates like a circuit, you can put two NO or NC contacts in series or in parallel. If both contacts become closed (NO contact variable = 1, NC contact variable = 0), then the signal is allowed to pass through. The **AND** is written by having two NO contacts. A **NAND** is written by putting two NC contacts in series. By placing two NO contacts in parallel you'll write an **OR**.

### B-4.3.1 Adding instructions

1) When adding instructions, you may click on an empty space of rung and type in the command if you know it or select it from the list of instructions. If you know the instruction name, you only need to have the empty rung highlighted and begin typing.

2) After the command is added, the next dialogue box that appears is the variable selector, which you use to associate a variable with the instruction. Again, if you know the variable, you may type it in or begin typing it, and it will match letters with variables on the list as you type.

### B-4.3.2 Basic Useful Instructions:

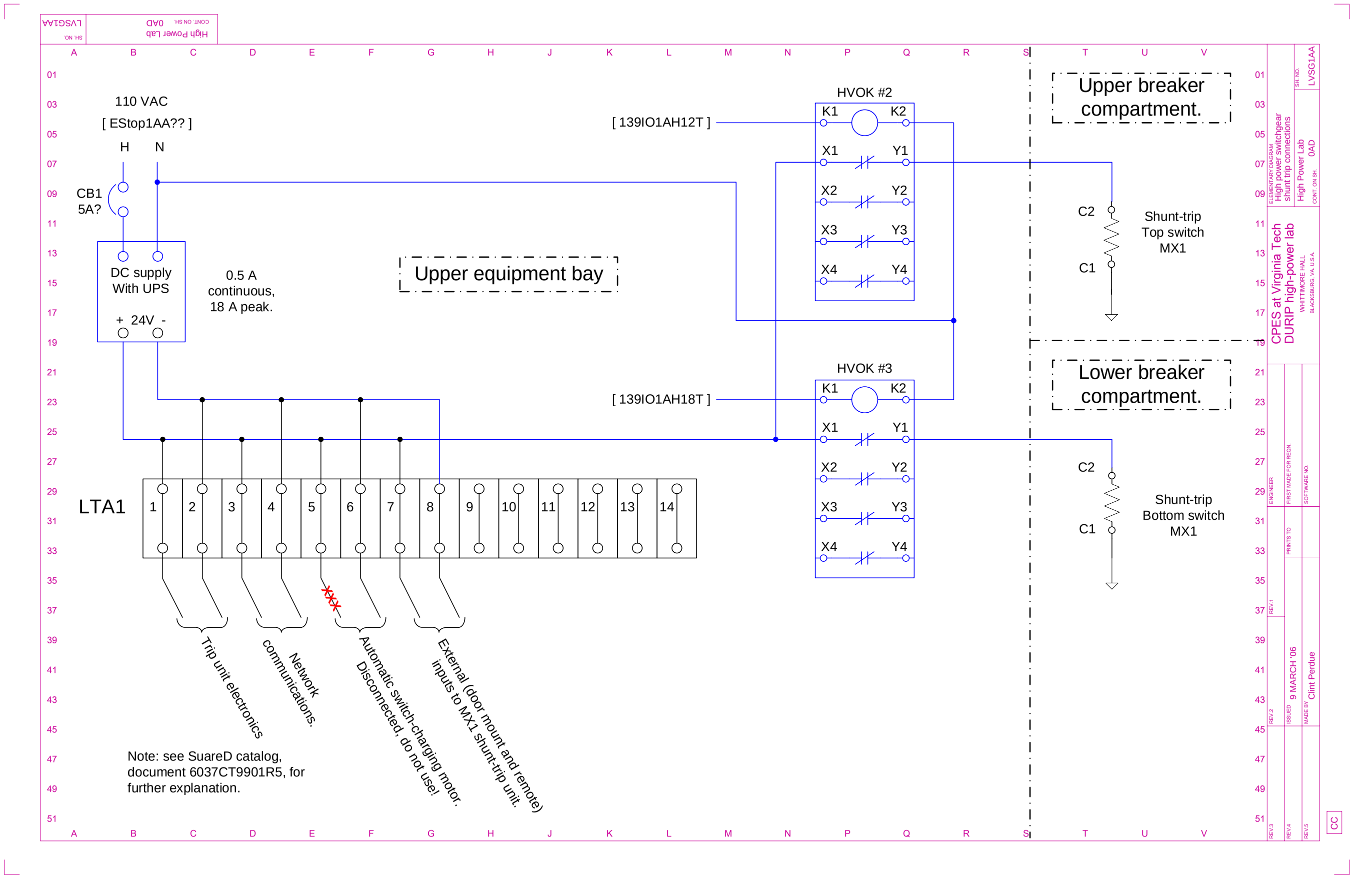
a. **NO:** Normally Open contact
b. **NC:** Normally Closed contact
c. **Out:** Output Relay
d. **TON:** Time-delayed turn ON. Time filter on rising edge. Note that name of instruction instance is a scratch variable for the time elapsed – don't reuse by mistake!
e. **TOF:** Time-delayed turn OFF. Time filter on falling edge, as above.
f. **TP:** Time-pulse. Output is true for time duration after rising edge of input. Again, instance name is the scratch/time elapsed variable.

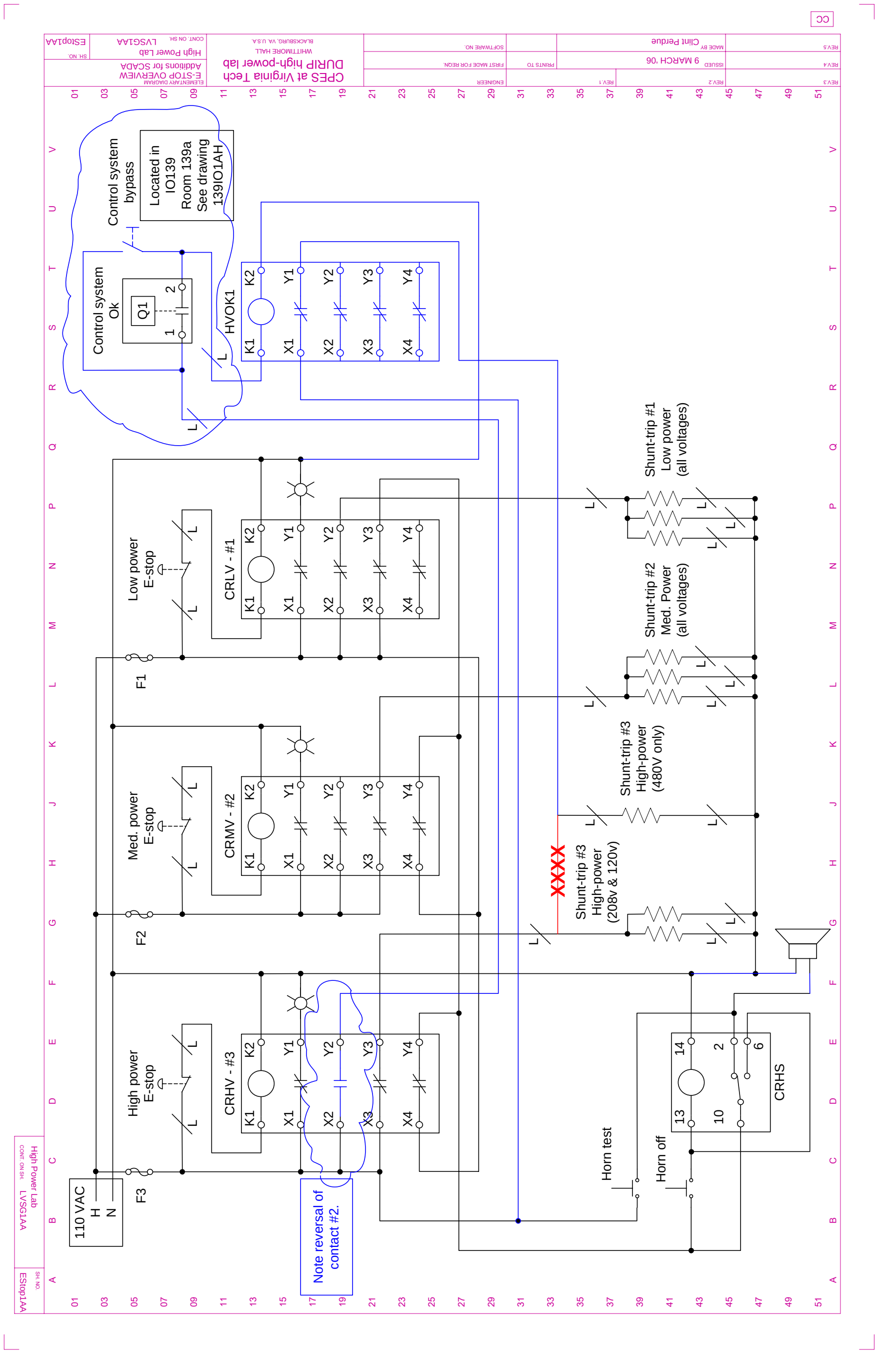
### B-4.3.3 Contact and coil types:

a. **P:** Positive edge triggered pulse (input goes true, output pulses true for one scan)
b. **N:** Negative edge triggered pulse (input goes false, output pulses true for one scan)
c. **S:** Set output <name>. <name> will latch true.
d. **R:** Reset output <name>. <name> will latch false.

# Appendix C – E-stop Interlock

This is the interlock between the existing lab E-stop, the switchgear, and the HV lab control. Per convention, black is existing, blue is new to be added, red is existing to be deleted.


110 VAC
[ EStop1AA?? ]
H
N
CB1
5A?
DC supply
With UPS
+ 24V -
0.5 A
continuous,
18 A peak.
Upper equipment bay
[ 139IO1AH12T ]
[ 139IO1AH18T ]
HVOK #2
HVOK #3
K1
K2
X1
Y1
X2
Y2
X3
Y3
X4
Y4
Upper breaker
compartment.
Lower breaker
compartment.
C2
C1
Shunt-trip
Top switch
MX1
Shunt-trip
Bottom switch
MX1
LTA1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
XXX
Trip unit electronics
Network
communications.
Automatic switch-charging motor.
Disconnected, do not use!
External (door mount and remote)
inputs to MX1 shunt-trip unit.
Note: see SuareD catalog,
document 6037CT9901R5, for
further explanation.
ELEMENTARY DIAGRAM
High power switchgear
shunt trip connections
SH. NO.
LVSG1AA
High Power Lab
0AD
CONT. ON SH.
CPES at Virginia Tech
DURIP high-power lab
WHITTIMORE HALL
BLACKSBURG, VA. U.S.A.
ENGINEER
FIRST MADE FOR REQN.
SOFTWARE NO.
PRINTS TO
REV.1
REV.2
ISSUED
9 MARCH '06
MADE BY
Clint Perdue
REV.3
REV.4
REV.5
CC


CPES at Virginia Tech
DURIP high-power lab
WHITTMORE HALL
BLACKSBURG, VA. U.S.A.
ELEMENTARY DIAGRAM
E-STOP OVERVIEW
Additions for SCADA
High Power Lab
CONT. ON SH.
LVSG1AA
SH. NO.
EStop1AA
SOFTWARE NO.
FIRST MADE FOR REQN.
ENGINEER
PRINTS TO
MADE BY
Clint Perdue
ISSUED
9 MARCH '06
REV.1
REV.2
REV.3
REV.4
REV.5
CC
Control system bypass
Located in IO139 Room 139a See drawing 139IO1AH
Control system Ok
Q1
1
2
HVOK1
K1
K2
X1
Y1
X2
Y2
X3
Y3
X4
Y4
L
Low power E-stop
CRLV - #1
F1
Med. power E-stop
CRMV - #2
F2
High power E-stop
CRHV - #3
F3
110 VAC
H
N
Note reversal of contact #2.
Shunt-trip #1 Low power (all voltages)
Shunt-trip #2 Med. Power (all voltages)
Shunt-trip #3 High-power (480V only)
XXXX
Shunt-trip #3 High-power (208v & 120v)
Horn test
Horn off
CRHS
13
14
10
2
6
High Power Lab
CONT. ON SH.
LVSG1AA
SH. NO.
EStop1AA

# Appendix D - Design Schedule

The following table was created early in the design process, as a predecessor to a Gantt chart for the design and construction.

| Item # | Task | Details | Deliverables & Milestones |
|---|---|---|---|
| **1** | **I/O** | Planning. | |
| **1.1** | Understand system architecture & requirements. Decide what I/O is needed. | List operator inputs, physical interlocks, system control outputs and data feedback. Specify bandwidth/sample rate and accuracy. | Master I/O list (spreadsheet in project directory). |
| **1.2** | Layout rooms, locate devices. | Identify device mounting positions, cluster wiring into local groups. | Map of rooms. |
| **1.3** | Select I/O devices (sensors, outputs) | Criteria = range, sensitivity, rate. | BOM |
| **1.4** | Select I/O modules (comm. Transducers). | Criteria = protocol, diagnostics, cost. buss/communications | BOM |
| **1.5** | Layout I/O panels. | | Construction drawings for I/O. |
| **1.5.1** | Physical mounting. | Wall mountings, din rails, couplings, holes. | |
| **1.5.2** | Place power, communications | 110VAC, 24VDC, E-net, other networks, breakers, fuses, grounding. | |
| **1.5.3** | Place terminals, modules | Have a connection point for each wire. | Wiring elementary drawings |
| **1.5.4** | route wiring in panel. | Pay attention to voltage levels, shielding, grounding. | Physical wiring layout for construction |
| **1.5.5** | Cable schedule (low level) | # of conductors/cables, termination locations, separation, shielding. | BOM |
| **1.5.6** | Conduit schedule (low level) | Protection as required. | BOM |
| **1.5.7** | Bill Of Materials for each panel. | | purchase order |
| | | | |
| **2** | **CONSTRUCTION** | | **Working I/O** |
| **2.1** | Assemble panels | Install equipment, internal low-level wiring. | |
| **2.2** | Mount panels | | |

| | | | |
|---|---|---|---|
| **2.3** | Install conduit | | |
| **2.4** | Pull wiring | | |
| **2.5** | Terminate wiring in panels | | |
| **2.6** | Checkout | Stimulate sensors, verify that logical signals conform. | |
| | | | |
| | | | |
| 3 | **Software setup** | | **Functional system** |
| **3.1** | Set up network (router, firewall) | Specify static IP addresses for all control equipment | Communications working. |
| **3.2** | Set up PCs, install tools | SCADA logic and HMI | Ability to configure system elements, monitor I/O states. |
| **3.3** | Create signal data. | Each I/O point must be affiliated with a logical signal. | |
| **3.4** | Write HMI screens | | Operator's displays and controls |
| **3.5** | Write control code | | Automation |
| **3.6** | Set up data logging. | | |

# Appendix E – High-Voltage Grounding and Substation Design.

The following document comes from other contributors at GE who consulted on this project. I have included it for background, and to help maintain a cohesive record even though it is outside my scope. I am not sure who to attribute this to, but I thank them for their insites.

## High Impedance Ground Fault Detection Discussion

The ground fault protection in GE Drive System's 2300-6600 Vll rms power electronics equipment and that envisioned for the Va Tech high power lab is of the high impedance type, that is, **there is a single point in the power electronics which is tied to cabinet frame through a moderate impedance**. The cabinet frame is then tied to earth ground.

One purpose for the high impedance ground technique is that on a first ground fault, the fault current will be limited by the moderate impedance, this fault can be detected and the drive shut down before a second ground fault occurs which can result in large fault current and significant subsequential damage.

All GE high voltage (2300-6600 Vll rms) power electronics equipment is ohmic isolated by a transformer from the 60 Hz three phase source which is typically 4160 Vll rms. The transformer delta or wye secondary is connected three-phase, three-wire to the power electronics equipment, that is, if the transformer secondary is connected in a wye, the transformer neutral is not connected to the power electronics.

The two types of power electronics manufactured at GE which are most similar to the intended Va Tech high voltage test setup each have a source power converter and a load power inverter with energy storage means, either a DC link filter capacitor, or a DC link reactor separating the two power converters.

If the energy storage between the two power converters is a DC link reactor, the equipment is referred to as current source equipment. If the energy storage between the two power converters is a DC link capacitor, the equipment is referred to as voltage source equipment.

For either the current source or voltage source equipment, the purpose of the high impedance ground fault circuit is to protect against subsequential damage due to the occurrence of a ground faults in either the transformer secondary, the cabling from transformer secondary to source converter, ground faults in the source converter, the DC link, the load converter, the load cables and the load or load transformer secondary if that be the case *(such as for the pump back test configuration envisioned for the Va Tech facility).*

## Discussion of Several High Impedance Ground Fault Detection Approaches

Fig 1 illustrates a ground fault detection approach based on sensing unreturned current. The source neutral is connected to ground through some impedance which could be zero. Fig 1 shows all three line currents going through one common CT, but each line could have its own CT and the three current summed electronically or otherwise. In the absence of a ground fault in the LOAD (downstream of the CT's), the currents should sum to zero.

It is envisioned this technique will be used to used to detect a ground fault in the transformer 480 Vll rms primary winding assuming the 480 Vll house feed to the Square D or equivalent low voltage switchgear is a three-phase wye with neutral tied to ground. Is is also assumed the transformer primary ground fault detection will be derived from the three line CT signals in the low voltage switchgear and will be summed and compared to a threshold in the low voltage switchgear electronics.

Note the cabling from the house feed to the low voltage switchgear is not included in this ground fault protection zone.

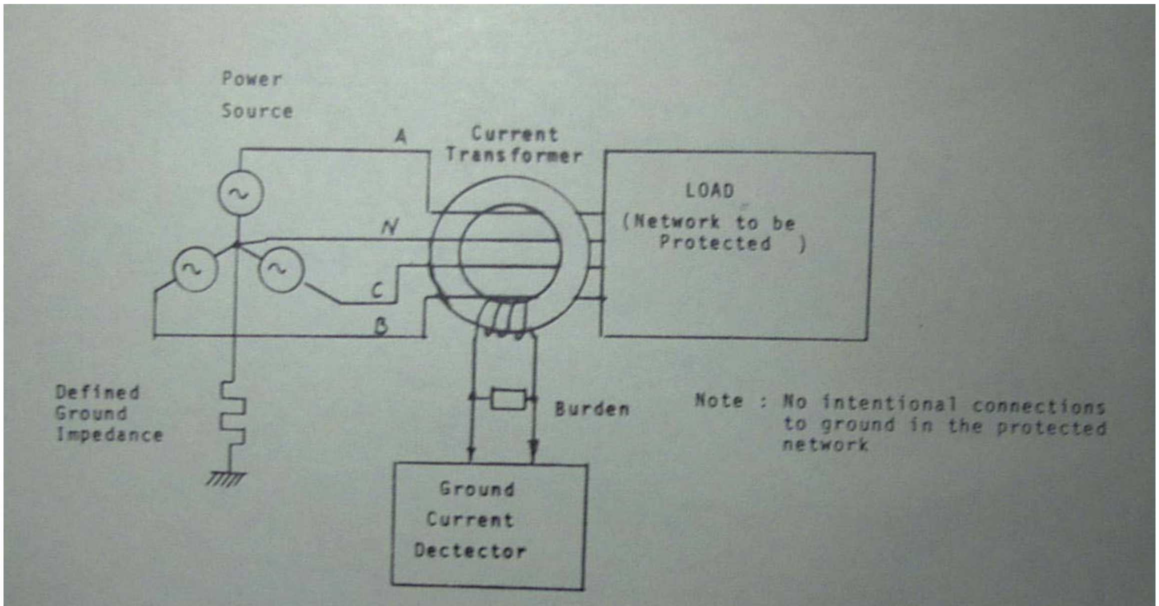


FIGURE 1

Prior Art Ground Fault Detector of the Type Which Senses Unreturned Current

Fig's 2a, 2b, 2c illustrate three high impedance ground fault detection schemes where voltage across or current through the grounding impedance is measured and compared to a threshold to detect a ground fault.

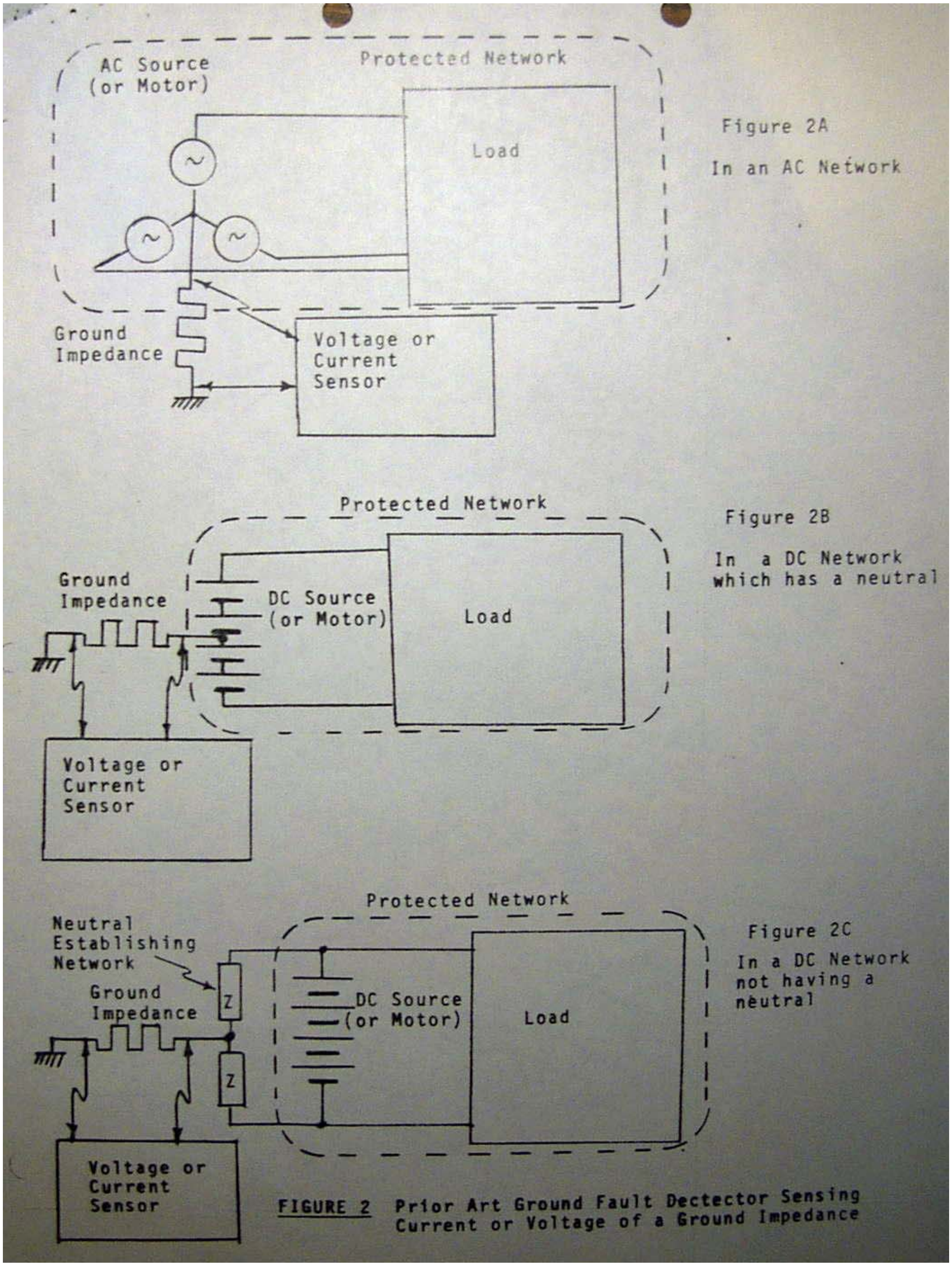


Figure 2A

In an AC Network

Figure 2B

In a DC Network which has a neutral

Figure 2C

In a DC Network not having a neutral

**FIGURE 2** Prior Art Ground Fault Dectector Sensing Current or Voltage of a Ground Impedance

Fig 3 illustrates a ground fault detection scheme used in all the GE high voltage (2300-6600 Vll rms) **current source power electronic equipment** implemented by using already existing RC line filter and line voltage sensing equipment. The neutral of the three phase wye connected RC filter is connected to ground. The line voltage sensing uses a high ohm (approximately 1 megohm) resistor string attenuator connected from line to ground to reduce high line-to-neutral voltage to a low signal level of approximately 2 volts. The three attenuated line voltages are summed and passed through a notch filter, then the absolute value of the notch filter is compared to an adjustable ground fault trip level. The purpose of the notch filter is to equalize the ground fault impedance trip level regardless of whether the ground fault occurs in the source, DC link, or load, and makes use of the fact that the ripple voltage amplitude and frequency on the summated line voltage signal signal will vary depending on where in the equipment the ground fault has occurred.

Fig 4 further expands on the equivalent circuit of this detection scheme and Fig 5 shows the detailed schematic.

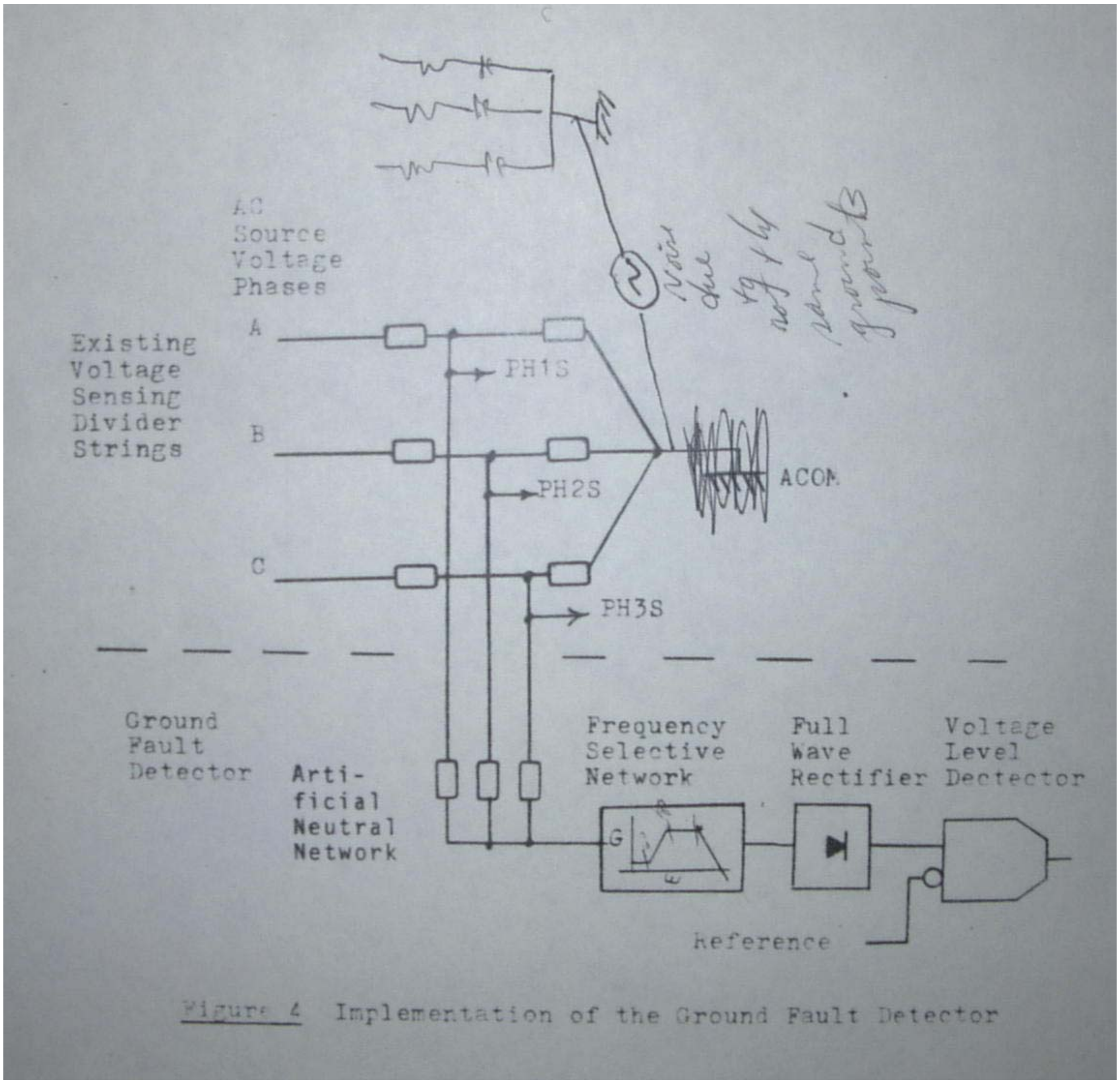

AC Source Voltage Phases
Existing Voltage Sensing Divider Strings
A
B
C
PH1S
PH2S
PH3S
ACOM
Noise due to not the same ground points
Ground Fault Detector
Artificial Neutral Network
Frequency Selective Network
G
Full Wave Rectifier
Voltage Level Dectector
Reference

Figure 4 Implementation of the Ground Fault Detector

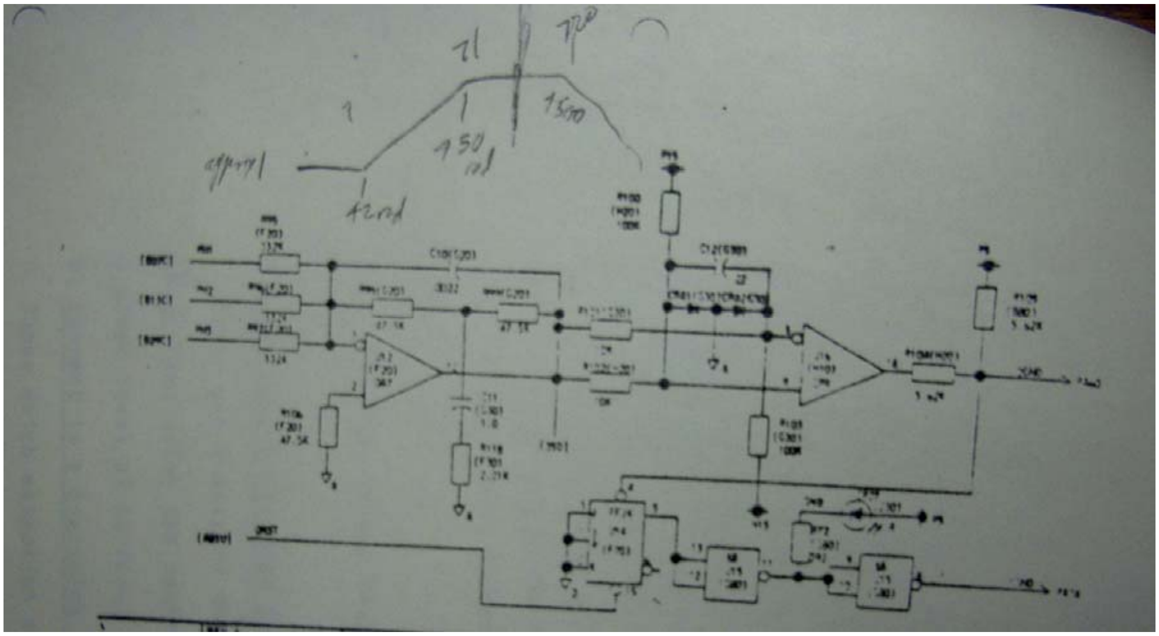


Fig 5
Detailed ground fault detector schematic

A different ground fault detection scheme in used in GE’s high voltage (2300-6600 Vll rms) **voltage source equipment**. This equipment uses either IGBT or IGCT based three-level PWM source and load converters separated by a center tapped DC link capacitor. The center tap of this DC link capacitor is connected to ground through a moderate impedance of approximately 40 ohms. If a ground fault occurs anywhere in the system, secondary of the transformer(s), source or load converter, DC link, or source or load AC line cables voltage will appear on this grounding resistor and when greater than a preset threshold annunciates a ground fault detected signal which can be used to shut down the equipment.

For redundancy, in conjunction with DC link ground resistor, the three phase AC line currents in the source converter are summed and when this summation signal is greater than a preset reference, a ground fault signal is annunciated that a ground fault has occurred somewhere upstream of the current sensing such as in the source transformer secondary or in the cabling from transformer secondary to source converter. In similar fashion, in conjunction with the same ground resistor, the summation of the three phase load converter line currents is compared to a preset threshold which when exceeded annunciates a ground fault signal upstream of the current sensors, that is, in the load transformer secondary or cabling from transformer secondary to load converter.

Fig 6 illustrates the power hardware used in the implementation of the ground fault detector in GE’s 6600 Vll rms IGCT voltage source power electronics equipment.

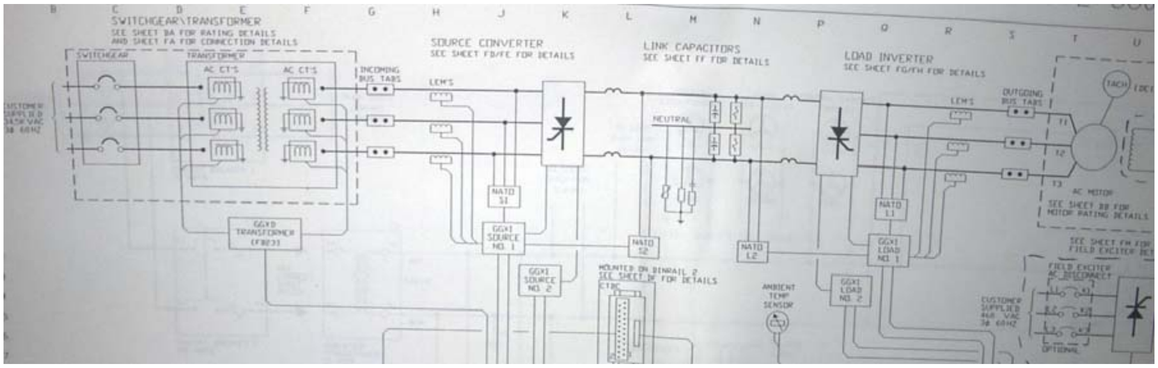


Fig 6
Illustration of ground fault detection hardware used in
GE voltage source power electronics equipment

This ground fault detection scheme exists in the 2300 Vll IGBT source converter that GE donated to Va Tech.

## 1MVA Transformer Discussion

For flexibility the 1MVA transformer secondary is reconnectable to the following options.

| Vll rms | Rated RMS Current @ 40 deg C ambient | MVA available in this connection |
|---|---|---|
| 1200 | 480.8 | 1 |
| 2078 | 277.6 | 1 |
| 2400 | 240.4 | 1 |
| 3174 | 138.8 | .76 |
| 4160 | 138.8 | 1 |

Assuming 480 Vll primary voltage, the nominal secondary voltage can be adjusted +/-5% by the primary taps.

The variable secondary voltage allows testing different voltage level equipments, or it allows higher voltage equipments to be initially tested at a lower, safer voltage, or it allows higher voltage to be tested at high voltage, low current, or at low voltage, high current. Therefore the rate of secondary winding reconnection might be higher than expected.

The transformer secondary cabling should be rated for maximum current times a factor of at least 1.25, 480.8*1.25 = 601 amp rms and for the maximum secondary voltage of 4160 Vll rms.

Rated 480 Vll rms primary line current is 1202.8 amp rms.

The installed transformer should have adequate clearance on all necessary sides to meet NEC requirements for cooling and to allow access for removal of panels necessary to make primary and secondary winding reconnection.

Allowable transformer acoustic noise at rated load has been specified to be 65 dbA. Since there are two identical transformers, if each is operating at rated load, the maximum acoustic noise for the two transformers is 65 + 3 = 68 dbA. I had requested that the iron core line reactors not add more than 1 dbA to total at rated load. Thus the maximum noise due to magnetics in transformer room is expected to be 69 dbA.

The maximum transformer loss at rated load is to be < 1.5% rated MVA. Thus watts loss to room due to the two transformers is 30 KW.

It would be reasonable to expect that heat runs will need to be conducted on the Equipment Under Test. Normally heats runs take about 4 hours for everything to stabilize. It is recommended that Va Tech calculate the expected transformer room temperature assuming 30 KW transformer losses and existing air conditioning. If forced air cooling is required, total cooling and magnetic acoustic noise should be estimated and compared to what the environment will tolerate.

## References:

GE Fanuc Automation. March 2003. VersaMax® Modules, Power Supplies, and Carriers User's Manual - GFK-1504K. USA.

GE Industrial Systems. 1999. Installation Guidance For Innovation Series™ Drive Systems - GEH-6380. General Electric Company. USA.

Schneider Electric Company. June 2001. MICROLOGIC® 5.0P and 6.0P Electronic Trip Units, Instruction Bulletin 48049-137-02. Cedar Rapids, IA, USA.

Schneider Electric Company. Nov. 2005. Power-Zone® 4 Low Voltage, Metal-Enclosed, Drawout Switchgear with Masterpact® NW and NT Low Voltage Power Circuit Breakers, Class 6037, Catalog 05. Cedar Rapids, IA, USA.

Schneider Electric Company. June 2001. Power-Zone® 4 Low Voltage, Metal-Enclosed, Drawout Switchgear with Masterpact® NW and NT Low Voltage Power Circuit Breakers, Class 6037, Instruction Bulletin. Cedar Rapids, IA, USA.

# Vita

Clinton Lee Perdue was born in Loudon, Virginia on September 20, 1970. He grew up on a small farm in Northern Virginia, developing an interest in the sciences through applied experience. Clinton matriculated to Virginia Polytechnic Institute and State University (Virginia Tech) in August of 1988 as an engineering student and member of the university's Corps of Cadets.

After receiving a BSEE in 1992 with concentrations in controls and microprocessors, Clinton accepted a position with General Electric Industrial Systems in Salem, Virginia. Over the next ten years this led to extended tours designing and commissioning control systems for heavy industrial facilities in The Peoples Republic of China, Brasil, Canada, The Netherlands, and domestically in the USA.

Clinton has served in the post of Royal Bard to Princess Susan of the Bunnies since 1992. He was made Knight-Order of the Shivering Duck in 2002 and been decorated with the Furtive Hare in 2005.

Clinton returned to Virginia Tech for a class in 2000, and to seek an MSEE in 2002. In 2005 he took a full-time position with Panaphase Technologies in Blacksburg, Virginia, developing drives and controls for switched-reluctance motors.