

## IEETI-216 – PLC/HMI Experiments

NAME: \_\_\_\_\_

Last update 3 February 2026

Activity #	Due by theory session #12	Score
01 PLC Trainer Build	Building your own personal PLC trainer unit	
02 Assessment	Program a PLC-based logic function	
03 Assessment	Design and build a simple PLC-based motor controller	

Activity #	Due by theory session #24	Score
04 PLC Trainer Build	Adding an HMI to the PLC trainer unit	
05 Assessment	PLC/HMI-based motor controller with counter	
06 Assessment	Troubleshoot a PLC/HMI-based motor controller	

Activity #	Due by theory session #36	Score
07 System Build	PLC-controlled system using timers and counters	
08 Assessment	PLC/HMI-based motor controller with timer	
09 Assessment	Troubleshoot a PLC/HMI-based motor controller	

Activity #	Due by theory session #48	Score
10 System Build	PLC-based analog measurement system	
11 Assessment	Design and build custom-scaled PLC/HMI analog signal monitor	
12 Assessment	Troubleshoot PLC-based analog measurement system	

Activity #	Due by theory session #60	Score
13 System Build	Networked PLC/HMI control system	
14 Assessment	Network diagnostic demonstrations	
15 Assessment	Troubleshoot networked PLC/HMI control system	

16 Lab clean-up	Clean and organize the entire lab room	
-----------------	--	--

### **Essential information about this course:**

- This is a laboratory course, but *the experiments are multi-concept systems rather than single-concept experiments.*
- These experimental systems follow concepts taught in that semester's theory course, which means the theory course outline and source texts are essential resources for you here.
- Half of your course grade comes from the number of attempts necessary to correctly complete each demonstration. Clarifying questions are welcome, but all results presented to the instructor will be assessed just like answers submitted on an exam.
- Half of your course grade comes from the number of attempts necessary to correctly complete each assessment activity (e.g. skills demonstrations and circuit troubleshooting). Just like the experiments, each assessment activity must be eventually completed correctly to pass the course.
- You should budget a minimum of 12 hours per week for this course. The lab is yours to use during all open-school hours. *Do not let the lack of fixed hours for this course lead to procrastination – manage your time wisely!*
- Successful students (1) review relevant theory before planning experiments, (2) prioritize hands-on lab time, (3) read instructions carefully, (4) build up their experiments step by step instead of trying to do it all at once, (5) document data neatly and accurately, and (6) double-check their work for correctness before submitting for assessment.

### **EET Program Learning Outcomes**

- (1) **COMMUNICATION and TEAMWORK** – Accurately communicate ideas across a variety of media (oral, written, graphical) to both technical and non-technical audiences; Function effectively as a member of a technical team.
- (2) **SELF-MANAGEMENT** – Arrive on time and prepared; Work diligently until the job is done; Budget resources appropriately to achieve objectives.
- (3) **SAFE WORK HABITS** – Comply with relevant national, state, local, and college safety regulations when designing, prototyping, building, and testing systems.
- (4) **ANALYSIS and DIAGNOSIS** – Select and apply appropriate principles and techniques for both qualitative and quantitative circuit analysis; Devise and execute appropriate tests to evaluate electronic system performance; Identify root causes of electronic system malfunctions.
- (5) **PROBLEM-SOLVING** – Devise and implement solutions for technical problems appropriate to the discipline.
- (6) **DOCUMENTATION** – Interpret and create technical documents (e.g. electronic schematic diagrams, block diagrams, graphs, reports) relevant to the discipline.
- (7) **INDEPENDENT LEARNING** – Select and research information sources to learn new principles, technologies, and/or techniques.

file eet\_outcomes

## Values

This educational program exists for one purpose: to empower you with a comprehensive set of knowledge, skills, and habits to unlock opportunities in your chosen profession. The following values articulate personal attitudes guaranteed to fulfill this purpose, and the principles upon which this program is designed. They embody what I like to call a *strong learning ethic*, similar to a strong work ethic but applied to the learning process rather than a job.

**Ownership** – you are the sole proprietor of your education, of your career, and to a great extent your quality of life. No one can force you to learn, make you have a great career, or grant you a fulfilling life – these accomplishments are possible only when you accept responsibility for them.

**Responsibility** – *ensuring* the desired outcome, not just *attempting* to achieve the outcome. Responsibility is how we secure rights and privileges.

**Initiative** – independently recognizing needs and taking responsibility to meet them.

**Integrity** – living in a consistently principled manner, communicating clearly and honestly, applying your best effort, and never trying to advance at the expense of others. Integrity is the key to trust, and trust is the glue that binds all relationships personal, professional, and societal.

**Perspective** – prioritizing your attention and actions to the things we will all care about for years to come. Never letting short-term concerns eclipse the long-term.

**Humility** – no one is perfect, and there is always something new to learn. Making mistakes is a symptom of living, and for this reason we need to be gracious to ourselves and to others.

**Safety** – assessing hazards and avoiding unnecessary risk to yourself and to others.

**Competence** – your ability to consistently and independently apply knowledge and skill to the solution of practical problems. Competence includes the ability to verify the appropriateness of your solutions and the ability to communicate so that others understand how and why your solutions work.

**Diligence** – exercising self-discipline and persistence in learning, accepting the fact there is no easy way to absorb complex knowledge, master new skills, or overcome limiting habits. Diligence in work means the job is not done until it is done *correctly*: all objectives achieved, all documentation complete, and all root-causes of problems identified and corrected.

**Community** – your actions impact other peoples' lives, for good or for ill. Conduct yourself not just for your own interests, but also for the best interests of those whose lives you affect.

**Respect** is the acknowledgment of others' intrinsic capabilities, responsibilities, and worth. Everyone has something valuable to contribute, and everyone deserves to fully *own* their lives.

file eet\_values

## **Course description**

This course reinforces the content of IETTI-215 (PLC/HMI Theory) by way of experimental system design and construction, starting with only a block diagram and a general description of the system's intended function. Students then engineer all details of the system, building and testing it in stages. Students also apply foundational circuit and digital networking principles to the diagnosis of simulated and real faults in these same types of circuits. Mastery standards applied to all experimental and diagnostic activities guarantee attainment of learning outcomes.

## **Course learning outcomes**

- Complete experimental systems and assessments on or before their associated deadlines, communicating professionally with instructors for the timely certification of those activities. (Addresses Program Learning Outcomes 1, 2)
- Design, construct, document, and test electrical systems using PLCs and HMIs to monitor and control parameters based on general specifications and high-level design ideas developed in partnership with the instructor, these systems incorporating electric motors, sequencing, digital logic functions, analog signaling, and digital networking. (Addresses Program Learning Outcomes 1, 2, 3, 4, 5, 6, 7)
- Correctly wire PLC I/O points to real-world devices, using manufacturer literature to identify voltage and current limitations as well as current sinking/sourcing direction. (Addresses Program Learning Outcomes 2, 3, 4, 5, 6, 7)
- Edit and write ladder-diagram code to implement monitoring and control functions in PLCs, and also create graphical monitoring screens for HMIs. (Addresses Program Learning Outcomes 2, 4, 6, 7)
- Construct PLC-based circuits meeting specifications set by the instructor and demonstrate proper test equipment usage including multimeters, PLC and HMI programming software, and network packet sniffing software. (Addresses Program Learning Outcomes 4, 5, 6)
- Troubleshoot real PLC- and HMI-controlled electrical systems from measurements taken at test points with circuit components and connections hidden from view. (Addresses Program Learning Outcomes 4, 6)
- Articulate diagnostic reasoning while troubleshooting these same circuits. (Addresses Program Learning Outcomes 1, 3)
- Create, maintain, and reference textual records (i.e. an Experiment Journal) documenting important principles and lessons learned. (Addresses Program Learning Outcomes 1, 6)

## Required Tools, Supplies, and Software

Listed by IETTI course number and course type (**Thy** = theory, **Exp** = Experiments, **Prj** = Projects).

Semester 1 = IETTI-101 (Theory), 103 (Experiments), and 102 (Projects)

Semester 2 = IETTI-104 (Theory), 112 (Experiments), and 105 (Projects)

Semester 3 = IETTI-222 (Theory), 221 (Experiments), and 220 (Projects)

Semester 4 = IETTI-223 (Theory), 225 (Experiments), and 236 (Projects)

Tool, Supply, or Software installation	Thy 101	Exp 103	Prj 102	Thy 104	Exp 112	Prj 105	Thy 222	Exp 221	Prj 220	Thy 223	Exp 225	Prj 236
\$5 mechanical pencil and eraser	X	X	X	X	X	X	X	X	X	X	X	X
\$5 permanent marker			X			X			X			X
\$25 scientific calculator	X	X	X	X	X	X	X	X	X	X	X	X
\$300 personal computer	X	X	X	X	X	X	X	X	X	X	X	X
\$10 USB “flash” drive	X	X	X	X	X	X	X	X	X	X	X	X
\$50-\$400 digital multimeter		X	X		X	X		X	X		X	X
\$190 USB-based oscilloscope		X	X		X	X		X	X		X	X
\$10 solderless breadboard (e.g. Busboard BB830)		X	X		X	X		X	X		X	X
\$25 grounding wrist strap		X	X		X	X		X	X		X	X
\$10 slotted screwdrivers (1/8",1/4")	X	X			X	X		X	X		X	X
\$10 Phillips screwdrivers (#1,#2)	X	X			X	X		X	X		X	X
\$10 jeweler’s screwdriver set	X	X			X	X		X	X		X	X
\$10 wire strippers, 18-24 AWG	X	X			X	X		X	X		X	X
\$10 needle-nose pliers	X	X			X	X		X	X		X	X
\$20 diagonal wire cutters	X	X			X	X		X	X		X	X
\$10 metal rule (inches & mm)	X	X			X	X		X	X		X	X
\$10 alligator-clip jumper wires (package of at least ten)		X	X		X	X		X	X		X	X
\$15 batteries: 6 Volt and 9 Volt	X	X			X	X		X	X		X	X
\$15 illuminated jeweler’s loupe	X	X			X	X		X	X		X	X
\$10 safety glasses		X			X	X			X			X
\$25-\$100 soldering iron (pencil-tip), 30 Watts or less			X		X	X			X			X
\$15 tube/spool of rosin-core solder			X		X	X			X			X
\$90 PLC I/O card	X				X			X			X	
\$15 microcontroller and software								X			X	X
\$0 software: Python interpreter	X			X			X			X	X	X
\$0 software: schematic editor		X	X		X	X		X	X		X	X
\$0 software: text editor		X			X			X			X	
\$0 software: NGSPICE circuit sim.	X				X			X			X	
\$0 software: WSL	X	X		X	X		X	X			X	
\$0 software: tshoot fault sim.	X	X		X	X			X			X	
\$0 software: sp simulator	X			X			X					
\$0 software: packet-sniffing software										X	X	X
\$0 archive: EETREF collection	X			X			X			X		

## Required Tools, Supplies, and Software

**Scientific calculator** – at minimum your calculator must perform trigonometric functions (sine, cosine, tangent, etc.), offer multiple memory registers, and display values in both scientific and “engineering” notations. I recommend either the Texas Instruments model TI-36X Pro or the Casio model fx-115ES because they easily perform complex-number arithmetic necessary for AC circuit analysis and are inexpensive.

**Personal computer** – all course materials are available in electronic format and are free (most are also open-source), making a portable computer extremely useful. The school provides personal computers for on-campus use, but having your own will enable you to work outside of school. Most operating systems, size of hard drive, amount of RAM memory, and screen size is appropriate, but your computer must have a keyboard and mouse (i.e. no tablets). Avoid ChromeBooks. Useful features worth higher cost include an RJ-45 Ethernet port and an EIA/TIA-232 (9-pin) serial port.

**Multimeter** – this is your first and most important electronic test instrument. At minimum it must measure DC and AC voltage, DC and AC current (milliAmpere range), resistance, and “diode check” voltage drop. Cost is a strong function of accuracy, additional features, frequency range, and safety (“Category” ratings for over-voltage exposure). The Fluke model 17B+ is an excellent choice for digital multimeters, and the Simpson 260 is an excellent professional-grade choice for analog multimeters. The best *minimal-cost* multimeter I’ve found is the model DT-830B which sells for less than \$20, with the test leads being the weakest component (but easily upgraded).

**Oscilloscope** – once too expensive for student purchase, entry-level USB-based oscilloscopes now cost less than a textbook. Pico Technology is an excellent brand, and their model 2204A comes with high-quality probes as well. Plugged into your personal computer using a USB cable, the Picoscope turns your computer’s monitor into a high-resolution oscilloscope display. Features include two measurement channels, 10 MHz bandwidth, built-in arbitrary waveform generator (AWG),  $\pm 100$  Volt over-voltage protection, digital “cursors” for precise interpretation of amplitude and frequency, meter-style measurement capability, Fast Fourier Transform algorithm for frequency-domain measurement, export ability to several graphic image formats as well as comma-separated variable (.csv) files, and serial communications signal decoding. Together with your multimeter, solderless breadboard and Development Board (which you will construct in the IETTI-102 Project course and is yours to keep) this forms a complete electronics laboratory for doing experiments and projects outside of school.

**Soldering** – the equipment you purchase for soldering need not be expensive, *if* you purchase the right solder. For electronics work you *must* use rosin-core solder. Kester is an excellent brand, and you should avoid cheap imported solders. For lead-based solder, a 63% tin and 37% lead alloy (Sn63/Pb37) works very well. A one-pound roll is likely more solder than you will need in these courses, so I recommend buying just a small tube or small roll. I recommend a fine-tipped soldering iron (15 Watts continuous power, although some with adjustable temperature controls may have higher power ratings to get up to soldering temperature more quickly) and a solder diameter 0.031 inches or smaller for doing fine printed-circuit board work. Also, keep the tip of your soldering iron clean by wiping it against a damp sponge or paper towel when hot, and not leaving it hot any longer than necessary. Hakko, X-tronic, and Lonove are all recommended brands.

**PLC I/O card** – any DC combo input-output card designed for use with Automation Direct Koyo “CLICK” PLC processors, such as the model C0-16CDD1 or C0-16CDD2 eight-point DC I/O cards.

**Microcontroller** – these courses are not brand- or model-specific, but the Texas Instruments MSP430 series is highly recommended for their powerful features, modern design, and programmability in multiple languages (assembly, C, C++, and Sketch). I particularly recommend the model MSP-EXP430G2ET “LaunchPad” development board (MSP430G2553IN20 microcontroller chip) with Code Composer Studio for the IDE software.

## Required Tools, Supplies, and Software

All software and documentation required for these courses is free, and much of it is open-source.

**Python interpreter** – available from [python.org](http://python.org) as a free download, Python is an interpreted programming language that works exceptionally as a scientific calculator. When installing, be sure to enable “Add python.exe to PATH” option.

**Schematic editor** – this is used to draft schematic diagrams for circuits. **KiCad** offers an excellent schematic editor as well as other tools useful for creating printed circuit board (PCB) layouts.

**Text editor** – this is used to create plain-text files, kind of like a word processor but lacking formatting features such as typeface, font size, etc. **Notepad++** and Microsoft’s **VS Code** are both excellent options.

**NGSPICE** – this is a modern adaptation of the venerable SPICE circuit simulator which uses a text-coded “netlist” rather than a visual schematic diagram to describe circuits. The installer lacks sophistication, being nothing more than a compressed (zip) file that you unpack. Once installed, you should instruct your computer’s operating system to automatically associate any files ending in the extension **.cir** with the NGSPICE executable file **ngspice.exe** so that all of your netlist files will appear with the NGSPICE icon and will automatically load into NGSPICE when double-clicked.

**WSL** – Windows Subsystem for Linux is a “virtual machine” Linux operating system that runs within the Windows operating system, giving you a command-line user environment mimicking that of a Unix operating system. It is a free application from Microsoft, with instructions available from Microsoft on how to install. I recommend installing the “Debian” distribution of WSL. Once installed, you will issue these commands in the following order to install all the necessary programming tools:

- `sudo apt update`
- `sudo apt install build-essential`

**tshoot** – this is a specialized circuit-simulator program that inserts faults into circuits and tests your ability to locate them. The download consists of one “tar” file you unpack and compile using the following three commands within a Unix-type operating system or within WSL. The fourth command runs the application:

- `mkdir tshoot ; mv -v *.tar tshoot ; cd tshoot`
- `tar xvf *.tar`
- `make`
- `./tshoot`

**sp** – this is a specialized circuit-simulator program for series/parallel resistor circuit analysis practice. The download consists of a single C-language source file that you may paste into an online C compiler to run, or compile on your own computer within WSL. The first command compiles, and the second one runs:

- `gcc -o sp sp.c -lm`
- `./sp`

**EETREF document collection** – an archive of manufacturer datasheets, manuals, tutorials, application notes, whitepapers, regulatory standards, and other documents in electronic form stored on every computer in the lab, in a folder marked “EETREF”. Copy this collection to your own computer for future reference.

**Packet-sniffing software** – this is specialized software for monitoring network communications. An excellent (and free) option is **Wireshark**.

**IDE software** – an “Integrated Development Environment” is a software package used to write code, and for our purposes this would be code meant to run on a microcontroller. For the Texas Instruments MSP430 we use **Code Composer Studio** as it supports assembly language, C, and C++. A third-party add-on called **Energia** supports Sketch programming, identical to that used by the popular Arduino microcontroller.

file eet\_tools

## Grading standards for Experiment courses

Your grade for this course is based on percentage scores (in every calculation rounded *down* to whole-numbered values), with each category weighted as follows:

- Experiment scores = 50% (Note: all Experiments are mastery-based, which means they must be eventually completed at 100% competence in order to pass the course)
- Assessment scores = 50% (Note: all Assessments are mastery-based, which means they must be eventually completed at 100% competence in order to pass the course)

Please note the importance of completing all Experiments and all Assessments on or before their respective deadline dates. If any Experiment or Assessment is incomplete by the end of the school day of the deadline date, it will receive a 0% score. If any Experiment or Assessment is incomplete by the end of the last day of the course, you will earn a failing grade (F) for the course. All Experiments and Assessments must be complete by the end of the last day of the course to receive a passing grade for the course.

Carefully follow the instructions associated with each experiment! If there is a prescribed order, you must follow this order of steps or else you will receive a 0% score for that experiment. The same is true if you plagiarize any part of it (i.e. present someone else's work as your own). For every experiment there will be multiple demonstrations, explanations, and/or challenges you must complete in the instructor's presence, and these are all mastery-based which means they all must be completed with 100% competence. Multiple opportunities are given to master each, with a point deduction levied for every re-try. You are welcome and encouraged at any time to seek clarification from the instructor so that you understand what is being asked of you, but you are solely responsible for doing the work and solving the problems. This is similar to the policy during written exams in a Theory course: the instructor will happily clarify expectations, but will not solve the problems for you nor confirm if an answer is correct prior to scoring it.

Electronic submissions of Experiments and Assessments are acceptable for full credit. The standards are just as high for electronic submissions as for face-to-face demonstrations. For Experiments, video documentation of you completing all objectives in their proper order will count as full credit. For Assessments your work must either be videorecorded in one seamless take or performed on a live video stream so the instructor is able to ensure you are doing the work yourself with no aid.

This course is based on experiments and hands-on assessments, and does not have scheduled meeting times as is the case with instructor-facilitated theory sessions. However, your punctual and consistent attendance is important for your success, as these activities require significant time-on-task to complete.

If you must be late or absent, it is imperative that you contact your instructor as well as any classmates you may be coordinating with so plans may be adjusted. It is still your responsibility to meet all deadlines.

A failing (F) grade will be earned for the entire course if any experiment or assessment is unfinished by the deadline date, or for any behavior that would result in being fired from a job, including false testimony (lying), cheating on any assignment or assessment, plagiarism (presenting another's work as your own, including output generated by artificial intelligence), willful violation of a safety policy, habitual tardiness or absenteeism, theft, harassment, sabotage, destruction of property, or intoxication.

## Getting help in the lab

Success in this career is largely a function of your ability to *independently* solve complex problems and to *empirically* learn new concepts and techniques. You will find the design, construction, testing, and diagnosis of real circuits and systems catalyze these abilities. During this course you may at times find yourself faced with a seemingly unsolvable problem. Your instructor's job is to provide practical advice useful for solving these problems, without directly solving them for you. Although this may feel frustrating for your instructor to withhold a solution from you, know that letting others solve these problems for you actually hinders your own professional development and limits your career opportunities.

Any time you request help, your instructor will first check to see you have done the following:

- **Simplify** – make problems manageable by building and testing your system in stages, by dividing large systems into smaller sections to isolate problems, and by avoiding complexity that isn't absolutely necessary to the function you are trying to obtain at that time. A common mistake is to build large systems without testing along the way in an attempt to save time, but this strategy usually creates problems that are more difficult to solve than if the system were built and tested piece by piece.
- **Document** – draw a diagram before building any system, and use this to guide your analysis and design decisions. If and when you make changes to your circuit, update your diagrams accordingly. Also, have any other relevant documents available for reference, such as datasheets, tutorials, etc. A common mistake is to begin building systems without first sketching what is to be built, and also not using the documentation for troubleshooting.
- **Measure and record** – take measurements using appropriate test equipment, because these measurements will reveal what your senses alone cannot perceive. Record those test results so you will not have to rely on memory to recall what you did.
- **First Principles** – always apply First Principles such as the guaranteed effects of opens and shorts, Ohm's Law, Joule's Law, Conservation of Energy and of Electric Charge, Kirchhoff's Laws, properties of series and of parallel networks, etc. when solving problems. Reference a list of these if necessary, checking them one by one to prompt and guide your reasoning.

A good way to remember these is to use the phrase, "*Slow Down My Friend!*" (Simplify, Document, Measure/record, and First Principles) which is also a reminder that problem-solving requires patience. A great many student problems are caused by impatience and an urge to take shortcuts!

A final suggestion to help you complete experiment and project assignments is to consult the Case Tutorial, Tutorial, and Questions chapters of the learning modules referenced your Theory course sessions for circuit ideas. If you find yourself unsure of where to begin for assigned experiments and/or projects, these references are almost always sure to help because they directly relate to the foundational concepts and principles taught in this program. Be sure to consult these references *before* requesting help from your instructor and/or classmates.

## Expectations for each experimental system

In the EET program the initial Experiments courses are based on small-scale experiments, each one designed to explore a single concept. Here in the later Experiments courses you build experimental *systems* involving multiple concepts, requiring research and design similar to that of real challenges on the job. The purpose of the introductory Experiments courses is to build competence in discrete skills; here the purpose is to develop a higher level of problem-solving and professionalism.

The systems you will build in this course borrow heavily from concepts and designs explored in previous courses, so you will find your previous coursework extremely useful as reference material.

Stand-alone demonstrations should be performed using *minimal* hardware – meaning the bare minimum of components connected together to perform each demonstration, ideally with that portion of the system functioning in complete isolation from all others. The purpose of this is to build the habit of building and testing large systems in stages which is a proven way to develop complex systems with as little troubleshooting as necessary. A deduction of –5% per re-try is levied on each unsuccessful stand-alone demonstration.

Complete system demonstrations must be performed after every stand-alone demonstration has been completed for that experimental system. All portions of the system must be in good working order, proven by having you demonstrate all bullet items listed for the complete system demonstration. If any portion of the complete system demonstration is faulty or otherwise incorrect, the instructor's inspection will stop and a deduction of –10% will be levied. You will then revisit your system and make all necessary corrections and reviews, and are welcome to re-try as many times as necessary. The same –10% deduction applies to all failed re-tries.

The “schematic diagram challenge” deserves elaboration here. Sketching a complete and accurate schematic diagram is vitally important for every system you design and build, but it takes too long for the instructor to perform a wire-by-wire inspection of every student's diagram on every system built. Instead, the instructor tests the accuracy of your schematic diagram by selecting random portions of your system and having you predict and measure signals there. These could be individual components or test points, they could be assuming normal operation or some special condition (e.g. a simulated fault), they could be selected from the physical system with you having to reference the schematic diagram, or vice-versa. In all cases, you must show an accurate correspondence between what you have built and what you have drawn, and demonstrate a thorough understanding of your system's behavior. Preparation consists of thoroughly reviewing both your system and your diagram, predicting what kinds of signals you ought to measure at all points in your system and making absolutely sure your schematic diagram is complete and accurate.

Each experimental system you build will be a testament to your knowledge and skill as an electronics technician, and will be thoroughly documented in your Experiment Journal which you may show to employers as a portfolio of your experimental capabilities.

As always, questions are welcome throughout the research, design, and construction process. You will never lose points for asking a question prior to an assessment, and your instructor will always clarify expectations so that you know what is being asked of you, but the moment you deem any part of your system ready the instructor's job is to grade your work. This is the same as on a job: when you present a finished product to a supervisor or client, it must actually be ready. The time to ask for clarification is before the inspection, not during.

## 01 PLC Trainer

NAME: \_\_\_\_\_ DUE DATE: \_\_\_\_\_

Design and build a “PLC trainer” suitable for PLC-based experimentation. This trainer module needs to be self-contained with a PLC processor, power supply operating from a standard 120 VAC electrical source, discrete-signal inputs, discrete-signal outputs, switches for driving discrete inputs, electric lamps for indicating discrete output status. Optionally, the PLC trainer may have analog-signal I/O as well, along with potentiometers for driving analog inputs. It is recommended that you use terminal blocks to “marshall” all the I/O channels so that when you connect and disconnect devices for various experiments all the wear and tear occurs at the replaceable terminal block rather than at the expensive I/O card.

Any brand or model of PLC is acceptable for your trainer. A PLC processor having Ethernet communication capability is *strongly* recommended for its high speed and easy ability to network with other devices such as HMI panels and other PLCs.

*Instructor consultation* – Discuss design options and strategies, responsibilities and expectations (including conditions for receiving help), safety considerations, etc. Be sure to maintain all records in a single “Experiment Journal” digital document, including test data, design changes, and lessons learned.

**Stand-alone demonstrations** (may be completed in any order; –5% per re-try, each):

*Instructor certifies* – Demonstrate how to properly interpret the I/O datasheet to determine discrete input voltage/current limits, as well as how to properly connect the input channels to devices such as switches.

*Instructor certifies* – Demonstrate how to properly interpret the I/O datasheet to determine discrete output voltage/current limits, as well as how to properly connect the output channels to load devices.

*Instructor certifies* – Demonstrate PLC processor powering up from 120 VAC source, using ohmmeter to verify robust chassis connection to the power cord’s ground pin (0.1 Ohms or less).

*Instructor certifies* – Demonstrate PLC programming software communicating with the PLC processor.

*Instructor certifies* – Demonstrate at least two discrete inputs being driven by discrete devices such as pushbutton switches.

*Instructor certifies* – Demonstrate at least two discrete outputs driving discrete loads such as LED indicating lamps.

**Complete system demonstration** (must be completed last, in a single session; –10% per re-try):

*Instructor certifies*

- All discrete inputs function properly (verified by bit states in PLC memory)
- All discrete outputs function properly (verified by electrical statuses at output terminals)
- All relevant Construction Standards met in the final build
- A single *Experiment Journal* document containing all information about the system
- Schematic diagram challenge

## 02 Assessment

NAME: \_\_\_\_\_ DUE DATE: \_\_\_\_\_

Design, build, and demonstrate a PLC-based function implementing a two-input logical function such as AND, OR, NAND, NOR, etc. This exercise tests your ability to program a PLC to implement a logical function using switches and lamps pre-wired to the PLC's discrete input and output channels, respectively. You will use your own PLC trainer that you built in the previous course activity.

Logic function type (instructor chooses) = \_\_\_\_\_

**SEQUENCE:** (1) Instructor chooses the logic function for you; (2) You write the PLC program (offline) and test it for syntax errors without connecting to a PLC; (3) Instructor inspects your program for completion prior to downloading to PLC; (4) You prove to the instructor that the program implements the specified logic function.

Circuit design assessments are *mastery-based*, meaning every one must be competently completed in order to pass the course, and you will be given multiple opportunities to re-try if you do not pass on the first attempt. Each re-try requires new criteria (e.g. different circuit, different design parameters). Scoring is based on the number of attempts necessary to successfully design, build, and demonstrate a circuit (e.g. 1 attempt = 100% ; 2 attempts = 80% ; 3 attempts = 60% ; 4 attempts = 40% ; 5 attempts = 20% ; 6 or more attempts = 0%). Failing to follow instructions counts as an unsuccessful attempt and will require a re-try.

file we\_1127

### 03 Assessment

NAME: \_\_\_\_\_

DUE DATE: \_\_\_\_\_

Design, build, and demonstrate a circuit where a programmable logic controller (PLC) starts up a small electric motor when a “start” pushbutton is pressed, and shuts it down when a “stop” pushbutton is pressed. Your circuit must use an “ice-cube” relay to interpose between the PLC’s output channel and the electric motor, and the electric motor must be powered from a different source than what is powering the PLC so that the PLC’s supply voltage does not “sag” or become “noisy” as the load energizes. You must connect a “commutating” diode in parallel with the relay’s coil to prevent the phenomenon known as “inductive kickback,” which may otherwise damage the transistor output on a PLC. Note that incorrectly connecting this diode will present a short-circuit to the PLC, so you *must* get it right!

This exercise tests your ability to properly interface devices with a PLC, to create a ladder-logic program for a PLC, to interpose between the PLC’s current-limited output channels and a load requiring much more current, to properly install a commutating diode.

The following components and materials will be available to you: assorted “ice cube” **relays** with DC-rated coils and matching **sockets** ; **terminal blocks** ; 1N400X rectifying **diodes** ; assorted pushbutton switches ; lengths of **hook-up wire** ; batteries. You must supply your PLC trainer as well as your own screwdrivers and multimeter for assembling and testing the circuit at your desk, and a copy of this page for your instructor to mark conditions. The instructor chooses specific I/O channels for you to connect the switches and relay.

“Start” switch to input: \_\_\_\_\_ “Stop” switch to input: \_\_\_\_\_ Relay to output: \_\_\_\_\_

**PLC program** (instructor chooses one of these options):

- PBNO “start” switch and PBNO “stop” switch
- PBNO “start” switch and PBNC “stop” switch
- PBNC “start” switch and PBNO “stop” switch
- PBNC “start” switch and PBNC “stop” switch

**SEQUENCE:** (1) Instructor chooses criteria and verifies no devices are wired to the PLC’s I/O terminals; (2) You write the PLC program (offline) and test it for syntax errors without connecting to a PLC; (3) Instructor inspects your program for completion prior to wiring; (4) You build and test the complete circuit (using a multimeter) without any power sources at all; (5) Instructor observes circuit energizing for the very first time; (6) You prove to the instructor that the circuit performs as predicted by your program, using test equipment as necessary.

Quantitative results must fall within the tolerance of your circuit’s components to be considered correct. For students working remotely rather than in-person, the entire exercise will take place via videoconference to permit instructor observation from start to finish.

Circuit design assessments are *mastery-based*, meaning every one must be competently completed in order to pass the course, and you will be given multiple opportunities to re-try if you do not pass on the first attempt. Each re-try requires new criteria (e.g. different circuit, different design parameters). Scoring is based on the number of attempts necessary to successfully design, build, and demonstrate a circuit (e.g. 1 attempt = 100% ; 2 attempts = 80% ; 3 attempts = 60% ; 4 attempts = 40% ; 5 attempts = 20% ; 6 or more attempts = 0%). Failing to follow instructions counts as an unsuccessful attempt and will require a re-try.

file we\_1128

## 04 PLC/HMI Trainer

NAME: \_\_\_\_\_ DUE DATE: \_\_\_\_\_

Add a Human-Machine Interface (HMI) panel to the “PLC trainer” you previously built. Like the PLC, this new HMI panel must be powered by the same 120 VAC source and preserve the integrity of the trainer as a self-contained unit. Any brand or model of HMI panel is acceptable for your trainer, so long as it is able to communicate effectively and reliably with your PLC’s processor.

*Instructor consultation* – Discuss design options and strategies, responsibilities and expectations (including conditions for receiving help), safety considerations, etc. Be sure to maintain all records in a single “Experiment Journal” digital document, including test data, design changes, and lessons learned.

**Stand-alone demonstrations** (may be completed in any order; –5% per re-try, each):

*Instructor certifies* – Demonstrate PLC and HMI together powering up from 120 VAC source, using ohmmeter (again) to verify robust chassis connection to the power cord’s ground pin (0.1 Ohms or less).

*Instructor certifies* – Demonstrate HMI programming software communicating with the HMI panel.

*Instructor certifies* – Demonstrate the HMI panel displaying the bit states of at least two discrete inputs being driven by discrete devices such as pushbutton switches.

*Instructor certifies* – Demonstrate the HMI panel controlling at least two discrete outputs driving discrete loads such as LED indicating lamps.

*Instructor certifies* – Demonstrate a latching on/off control program in the PLC using standard contact and coil instructions.

*Instructor certifies* – Demonstrate a latching on/off control program in the PLC using retentive instructions (e.g. Set and Reset coils).

**Complete system demonstration** (must be completed last, in a single session; –10% per re-try):

*Instructor certifies*

- All discrete input statuses correctly shown by the HMI
- All discrete outputs controllable by virtual pushbuttons on the HMI
- HMI display screen must be comfortably visible from a seated position
- All relevant Construction Standards met in the final build
- A single *Experiment Journal* document containing all information about the system
- Schematic diagram challenge (again, on the PLC I/O portion of your trainer)

## 05 Assessment

NAME: \_\_\_\_\_

DUE DATE: \_\_\_\_\_

Design, build, and demonstrate a circuit where a programmable logic controller (PLC) controls the on/off status of a small electric motor according to the state of one or more pushbutton switches as well as a counter instruction within the PLC. Your circuit must use an “ice-cube” relay to interpose between the PLC’s output channel and the electric motor, and the electric motor must be powered from a different source than what is powering the PLC so that the PLC’s supply voltage does not “sag” or become “noisy” as the load energizes. You must connect a “commutating” diode in parallel with the relay’s coil to prevent the phenomenon known as “inductive kickback,” which may otherwise damage the transistor output on a PLC. Note that incorrectly connecting this diode will present a short-circuit to the PLC, so you *must* get it right!

This exercise tests your ability to properly interface devices with a PLC, to create a ladder-logic program for a PLC, to interpose between the PLC’s current-limited output channels and a load requiring much more current, to properly install a commutating diode.

The following components and materials will be available in addition to the PLC trainer you constructed for yourself: assorted “ice cube” **relays** with DC-rated coils and matching **sockets** ; **terminal blocks** ; 1N400X rectifying **diodes** ; assorted pushbutton switches ; lengths of **hook-up wire**. You will be expected to supply your own screwdrivers and multimeter for assembling and testing the circuit at your desk, as well as a copy of this page for your instructor to mark conditions. The instructor chooses specific I/O channels for you to connect switches and relay.

“Start” switch to input: \_\_\_\_\_ “Stop” switch to input: \_\_\_\_\_ Relay to output: \_\_\_\_\_

**PLC program** (instructor chooses one of these options):

\_\_\_\_\_ Motor starts after “start” switch pressed \_\_\_\_\_ times, stops with one press of “stop” switch

\_\_\_\_\_ Motor starts with one press of “start” switch, stops when “stop” switch pressed \_\_\_\_\_ times

\_\_\_\_\_ Motor starts and stops normally, but refuses to start again after \_\_\_\_\_ start/stop cycles  
(Feel free to add another switch to any input channel to reset this lockout function)

Your program must be written such that every start/stop cycle behaves the same.

**SEQUENCE:** (1) Instructor chooses criteria and verifies no devices are wired to the PLC’s I/O terminals; (2) You write the PLC program (offline) and test it for syntax errors without connecting to a PLC; (3) Instructor inspects your program for completion prior to wiring; (4) You build and test the complete circuit (using a multimeter) without any power sources at all; (5) Instructor observes circuit energizing for the very first time; (6) You prove to the instructor that the circuit performs as predicted by your program, using test equipment as necessary.

Quantitative results must fall within the tolerance of your circuit’s components to be considered correct. For students working remotely rather than in-person, the entire exercise will take place via videoconference to permit instructor observation from start to finish.

Circuit design assessments are *mastery-based*, meaning every one must be competently completed in order to pass the course, and you will be given multiple opportunities to re-try if you do not pass on the first attempt. Each re-try requires new criteria (e.g. different circuit, different design parameters). Scoring is based on the number of attempts necessary to successfully design, build, and demonstrate a circuit (e.g. 1 attempt = 100% ; 2 attempts = 80% ; 3 attempts = 60% ; 4 attempts = 40% ; 5 attempts = 20% ; 6 or more attempts = 0%). Failing to follow instructions counts as an unsuccessful attempt and will require a re-try.

file we\_1132

## 06 Assessment

NAME: \_\_\_\_\_ DUE DATE: \_\_\_\_\_

Troubleshoot a fault within a *PLC-controlled* motor circuit. This circuit shall be constructed in such a manner that all circuit components and simulated faults must be hidden from view (e.g. covering it up with a box or towel) but test points will be available for contact with a multimeter's probes. A schematic diagram showing the circuit and its test points will be allowed for use during the troubleshooting exercise. You will also be allowed to examine a "live view" of the PLC's program to observe bit states and other relevant variables within the PLC's memory.

Possible faults include:

- Any cable failed open
- Any cable failed shorted
- Any component failed open
- Any component failed shorted
- Any component value altered
- Any bit forced in the PLC
- PLC run/stop mode incorrect
- Error introduced into the PLC program

First, you will demonstrate that the system functions properly. Then the instructor will either set up or supervise other students setting up a random fault in that system (hidden from view) while you are out of the room. You will then have a limited amount of time to independently perform measurements and other tests while under the continuous observation of the instructor. A successful troubleshooting exercise consists of both correctly identifying the location and nature of the fault, as well as logically defending the necessity of each diagnostic step. Incorrect fault identification, unnecessary steps, and/or incorrect defense of any step will result in a failed attempt. Your only access to the faulted circuit will be via the test points, and only one unpowered test will be permitted.

If you must work remotely rather than in-person, the faulted system must be at the instructor's location while you request measurements and other diagnostic tests of the instructor via teleconferencing system (e.g. videoconference, telephone, text messaging).

Troubleshooting is *mastery-based*, meaning every one must be competently completed in order to pass the course, and you will be given multiple opportunities to re-try if you do not pass on the first attempt. Each re-try begins with another randomized fault on the same circuit. Scoring is based on the number of attempts necessary to successfully troubleshoot a circuit (e.g. 1 attempt = 100% ; 2 attempts = 80% ; 3 attempts = 60% ; 4 attempts = 40% ; 5 attempts = 20% ; 6 or more attempts = 0%).

file we\_1028

## 07 System Build

NAME: \_\_\_\_\_ DUE DATE: \_\_\_\_\_

Design and build an experimental system utilizing a PLC to monitor and/or control some system using counter and timer instructions. Examples include an electric motor control system, reaction timer game, traffic light controller, flame safety system with furnace purge timer, etc. Your system should be sufficiently complex to utilize the allotted time and to challenge your technical abilities. The PLC used in this system must be a different model from the one in your personal “PLC trainer” unit, to give you experience with different models of PLC. However, you may still find it useful to experiment with programming ideas in your trainer before implementing these in the system’s PLC.

*Instructor consultation* – Discuss design options and strategies, responsibilities and expectations (including conditions for receiving help), safety considerations, etc. Be sure to maintain all records in a single “Experiment Journal” digital document, including test data, design changes, and lessons learned.

**Stand-alone demonstrations** (may be completed in any order; –5% per re-try, each):

*Instructor certifies* – Demonstrate PLC programming software successfully communicating with the PLC.

*Instructor certifies* – Demonstrate a PLC program implementing at least one form of counter instruction, viewing the counter’s accumulated value within the ladder-diagram editing software.

*Instructor certifies* – Demonstrate a PLC program implementing at least one form of timer instruction, viewing the counter’s accumulated value within the ladder-diagram editing software.

*Instructor certifies* – Demonstrate an HMI communicating with the PLC and displaying both discrete and integer values used by the PLC’s program.

*Instructor certifies* – Demonstrate an HMI communicating with the PLC and controlling (setting) both discrete and integer values used by the PLC’s program.

*Instructor certifies* – Demonstrate an HMI providing redundant inputs to all physical switches, pushbuttons and other operator controls so that the program is able to be operated by physical controls or by the HMI.

**Complete system demonstration** (must be completed last, in a single session; –10% per re-try):

*Instructor certifies*

- System operates as expected with no surprising behaviors from the PLC
- HMI shows a descriptive representation of all relevant I/O states, count values, and elapsed times
- A single *Experiment Journal* document containing all information about the system
- Schematic diagram challenge

## 08 Assessment

NAME: \_\_\_\_\_

DUE DATE: \_\_\_\_\_

Design, build, and demonstrate a circuit where a programmable logic controller (PLC) controls the on/off status of a small electric motor according to the state of one or more pushbutton switches as well as a timer instruction within the PLC. Your circuit must use an “ice-cube” relay to interpose between the PLC’s output channel and the electric motor, and the electric motor must be powered from a different source than what is powering the PLC so that the PLC’s supply voltage does not “sag” or become “noisy” as the load energizes. You must connect a “commutating” diode in parallel with the relay’s coil to prevent the phenomenon known as “inductive kickback,” which may otherwise damage the transistor output on a PLC. Note that incorrectly connecting this diode will present a short-circuit to the PLC, so you *must* get it right!

This exercise tests your ability to properly interface devices with a PLC, to create a ladder-logic program for a PLC, to interpose between the PLC’s current-limited output channels and a load requiring much more current, to properly install a commutating diode.

The following components and materials will be available in addition to the PLC trainer you constructed for yourself: assorted “ice cube” **relays** with DC-rated coils and matching **sockets** ; **terminal blocks** ; 1N400X rectifying **diodes** ; assorted pushbutton switches ; lengths of **hook-up wire**. You will be expected to supply your own screwdrivers and multimeter for assembling and testing the circuit at your desk, as well as a copy of this page for your instructor to mark conditions. The instructor chooses specific I/O channels for you to connect switches and relay.

“Start” switch to input: \_\_\_\_\_ “Stop” switch to input: \_\_\_\_\_ Relay to output: \_\_\_\_\_

**PLC program** (instructor chooses one of these options):

- Motor starts after “start” switch held for \_\_\_\_\_ seconds, stops immediately when “stop” pressed
- Motor starts immediately, stops if “stop” switch held for \_\_\_\_\_ seconds
- Motor starts immediately and automatically stops after \_\_\_\_\_ seconds  
(Stop switch overrides this time delay to stop immediately if pressed)

Your program must be written such that every start/stop cycle behaves the same.

**SEQUENCE:** (1) Instructor chooses criteria and verifies no devices are wired to the PLC’s I/O terminals; (2) You write the PLC program (offline) and test it for syntax errors without connecting to a PLC; (3) Instructor inspects your program for completion prior to wiring; (4) You build and test the complete circuit (using a multimeter) without any power sources at all; (5) Instructor observes circuit energizing for the very first time; (6) You prove to the instructor that the circuit performs as predicted by your program, using test equipment as necessary.

Quantitative results must fall within the tolerance of your circuit’s components to be considered correct. For students working remotely rather than in-person, the entire exercise will take place via videoconference to permit instructor observation from start to finish.

Circuit design assessments are *mastery-based*, meaning every one must be competently completed in order to pass the course, and you will be given multiple opportunities to re-try if you do not pass on the first attempt. Each re-try requires new criteria (e.g. different circuit, different design parameters). Scoring is based on the number of attempts necessary to successfully design, build, and demonstrate a circuit (e.g. 1 attempt = 100% ; 2 attempts = 80% ; 3 attempts = 60% ; 4 attempts = 40% ; 5 attempts = 20% ; 6 or more attempts = 0%). Failing to follow instructions counts as an unsuccessful attempt and will require a re-try.

file we\_1133

## 09 Assessment

NAME: \_\_\_\_\_ DUE DATE: \_\_\_\_\_

Troubleshoot a fault within a *PLC-controlled* motor circuit. This circuit shall be constructed in such a manner that all circuit components and simulated faults must be hidden from view (e.g. covering it up with a box or towel) but test points will be available for contact with a multimeter's probes. A schematic diagram showing the circuit and its test points will be allowed for use during the troubleshooting exercise. You will also be allowed to examine a "live view" of the PLC's program to observe bit states and other relevant variables within the PLC's memory.

Possible faults include:

- Any cable failed open
- Any cable failed shorted
- Any component failed open
- Any component failed shorted
- Any component value altered
- Any bit forced in the PLC
- PLC run/stop mode incorrect
- Error introduced into the PLC program

First, you will demonstrate that the system functions properly. Then the instructor will either set up or supervise other students setting up a random fault in that system (hidden from view) while you are out of the room. You will then have a limited amount of time to independently perform measurements and other tests while under the continuous observation of the instructor. A successful troubleshooting exercise consists of both correctly identifying the location and nature of the fault, as well as logically defending the necessity of each diagnostic step. Incorrect fault identification, unnecessary steps, and/or incorrect defense of any step will result in a failed attempt. Your only access to the faulted circuit will be via the test points, and only one unpowered test will be permitted.

If you must work remotely rather than in-person, the faulted system must be at the instructor's location while you request measurements and other diagnostic tests of the instructor via teleconferencing system (e.g. videoconference, telephone, text messaging).

Troubleshooting is *mastery-based*, meaning every one must be competently completed in order to pass the course, and you will be given multiple opportunities to re-try if you do not pass on the first attempt. Each re-try begins with another randomized fault on the same circuit. Scoring is based on the number of attempts necessary to successfully troubleshoot a circuit (e.g. 1 attempt = 100% ; 2 attempts = 80% ; 3 attempts = 60% ; 4 attempts = 40% ; 5 attempts = 20% ; 6 or more attempts = 0%).

file we\_1028

## 10 System Build

NAME: \_\_\_\_\_ DUE DATE: \_\_\_\_\_

Design and build an experimental system utilizing a PLC to measure one or more analog variables via a sensor and activate discrete output(s) in response to the analog input(s). Examples include an air pressure monitoring system, voltage-sensing battery charge controller, water tank level monitor, thermostatic control system, etc. Your system should be sufficiently complex to utilize the allotted time and to challenge your technical abilities. The PLC used in this system must be a different model from the one in your personal “PLC trainer” unit, to give you experience with different models of PLC. However, you may still find it useful to experiment with programming ideas in your trainer before implementing these in the system’s PLC.

*Instructor consultation* – Discuss design options and strategies, responsibilities and expectations (including conditions for receiving help), safety considerations, etc. Be sure to maintain all records in a single “Experiment Journal” digital document, including test data, design changes, and lessons learned.

**Stand-alone demonstrations** (may be completed in any order; –5% per re-try, each):

*Instructor certifies* – Demonstrate PLC programming software successfully communicating with the PLC.

*Instructor certifies* – Demonstrate an HMI communicating with the PLC and displaying floating-point (real) values in the PLC’s memory.

*Instructor certifies* – Demonstrate how to properly interpret the I/O datasheet to determine analog-to-digital resolution and maximum signal limits, as well as how to properly connect the analog input channels to devices such as potentiometric sensors, potentiometers, etc.

*Instructor certifies* – Demonstrate the raw analog input signal’s digitized value as registered within the PLC’s memory across the full range of analog signal voltage or current presented to the input channel.

*Instructor certifies* – Demonstrate the HMI displaying the raw analog input signal’s digitized value.

*Instructor certifies* – Demonstrate the HMI displaying a scaled version of the analog input signal’s digitized value, the LRV and URV of that scaled range being arbitrarily programmable.

*Instructor certifies* – Demonstrate the PLC controlling a discrete output based on the analog input signal’s value.

**Complete system demonstration** (must be completed last, in a single session; –10% per re-try):

*Instructor certifies*

- Proper control action occurs
- HMI shows a descriptive representation of all relevant remote device states and values
- HMI shows an alarm if the analog signal value goes out of bounds due to a wiring failure
- A single *Experiment Journal* document containing all information about the system
- Schematic diagram challenge

## 11 Assessment

NAME: \_\_\_\_\_ DUE DATE: \_\_\_\_\_

Design, build, and demonstrate a circuit taking in a DC analog signal voltage from a potentiometer or DC voltage source (simulating an analog sensor) and scaling that signal to display numerical values on a Human-Machine Interface (HMI) networked to a PLC. The instructor will randomly assign the DC analog signal voltage range as well as the scaled numerical display range (e.g. 0 to +4 Volts DC must indicate as -50 kPa to +100 kPa, respectively). The PLC must also control the energization of a discrete (on/off) load based on the scaled value of this analog signal voltage.

This exercise tests your ability to properly connect analog voltage signals to analog PLC input points, properly write ladder-diagram code for that PLC to interpret the analog signal and control an output based on its value, properly configure an HMI to communicate with the PLC to acquire the digitized values, and properly configure the graphical display of an HMI to show that scaled numerical value.

The following components and materials will be available in addition to the PLC trainer you constructed for yourself: an adjustable **DC voltage source** for the analog signal ; **terminal blocks** ; and lengths of **hook-up wire**.

**Analog input signal range** (instructor chooses) = from \_\_\_\_\_ V to \_\_\_\_\_ V

**Scaled display value range** (instructor chooses) = from \_\_\_\_\_ to \_\_\_\_\_

**Load turns on when** (instructor chooses) = display value is (*less than/greater than*) \_\_\_\_\_

The HMI's display must legibly show all digits of the scaled value, as well as the unit of measurement specified by the instructor.

**SEQUENCE:** (1) Instructor chooses criteria and verifies no devices are wired to the PLC's I/O terminals; (2) You write the PLC program (offline) and test it for syntax errors without connecting to a PLC; (3) Instructor inspects your program for completion prior to wiring; (4) You build and test the complete circuit (using a multimeter) without any power sources at all; (5) Instructor observes circuit energizing for the very first time; (6) You prove to the instructor that the circuit performs as predicted by your program, using test equipment as necessary.

Quantitative results must fall within the tolerance of your circuit's components to be considered correct. For students working remotely rather than in-person, the entire exercise will take place via videoconference to permit instructor observation from start to finish.

Circuit design assessments are *mastery-based*, meaning every one must be competently completed in order to pass the course, and you will be given multiple opportunities to re-try if you do not pass on the first attempt. Each re-try requires new criteria (e.g. different circuit, different design parameters). Scoring is based on the number of attempts necessary to successfully design, build, and demonstrate a circuit (e.g. 1 attempt = 100% ; 2 attempts = 80% ; 3 attempts = 60% ; 4 attempts = 40% ; 5 attempts = 20% ; 6 or more attempts = 0%). Failing to follow instructions counts as an unsuccessful attempt and will require a re-try.

file we\_1130

## 12 Assessment

NAME: \_\_\_\_\_

DUE DATE: \_\_\_\_\_

Troubleshoot a fault within a *PLC-based analog measurement* circuit. This circuit shall be constructed in such a manner that all circuit components and simulated faults must be hidden from view (e.g. covering it up with a box or towel) but test points will be available for contact with a multimeter's probes. A schematic diagram showing the circuit and its test points will be allowed for use during the troubleshooting exercise.

Possible faults include:

- Any cable failed open
- Any cable failed shorted
- Any component failed open
- Any component failed shorted
- Any component value altered
- Any bit forced in the PLC
- PLC run/stop mode incorrect
- Error introduced into the PLC program
- Misconfigured analog input channel/card on the PLC

First, you will demonstrate that the system functions properly. Then the instructor will either set up or supervise other students setting up a random fault in that system (hidden from view) while you are out of the room. You will then have a limited amount of time to independently perform measurements and other tests while under the continuous observation of the instructor. A successful troubleshooting exercise consists of both correctly identifying the location and nature of the fault, as well as logically defending the necessity of each diagnostic step. Incorrect fault identification, unnecessary steps, and/or incorrect defense of any step will result in a failed attempt. Your only access to the faulted circuit will be via the test points, and only one unpowered test will be permitted.

If you must work remotely rather than in-person, the faulted system must be at the instructor's location while you request measurements and other diagnostic tests of the instructor via teleconferencing system (e.g. videoconference, telephone, text messaging).

Troubleshooting is *mastery-based*, meaning every one must be competently completed in order to pass the course, and you will be given multiple opportunities to re-try if you do not pass on the first attempt. Each re-try begins with another randomized fault on the same circuit. Scoring is based on the number of attempts necessary to successfully troubleshoot a circuit (e.g. 1 attempt = 100% ; 2 attempts = 80% ; 3 attempts = 60% ; 4 attempts = 40% ; 5 attempts = 20% ; 6 or more attempts = 0%).

file we\_1030

## 13 System Build

NAME: \_\_\_\_\_ DUE DATE: \_\_\_\_\_

Design and build an experimental system utilizing both a PLC and HMI to implement control over a network. Examples include a PLC local to the HMI reading from inputs and/or driving outputs on a remote PLC, the PLC and HMI controlling a VFD or servo drive via a network, etc. Your system should be sufficiently complex to utilize the allotted time and to challenge your technical abilities. The PLC used in this system must be a different model from the one in your personal “PLC trainer” unit, to give you experience with different models of PLC. However, you may still find it useful to experiment with programming ideas in your trainer before implementing these in the system’s PLC.

*Instructor consultation* – Discuss design options and strategies, responsibilities and expectations (including conditions for receiving help), safety considerations, etc. Be sure to maintain all records in a single “Experiment Journal” digital document, including test data, design changes, and lessons learned.

**Stand-alone demonstrations** (may be completed in any order; –5% per re-try, each):

*Instructor certifies* – Demonstrate how to configure TCP/IP parameters to enable communication between two or more Ethernet devices.

*Instructor certifies* – Demonstrate the use of Wireshark or other “packet-sniffing” software to analyze the contents of messages conveyed over an Ethernet network.

*Instructor certifies* – Demonstrate the use of the `ping` command to test connectivity in an IP network.

*Instructor certifies* – Demonstrate the HMI displaying values read from a remote device.

*Instructor certifies* – Demonstrate the HMI writing values to a remote device.

*Instructor certifies* – Demonstrate the effect of different network polling/update times on the local PLC’s ability to implement control.

**Complete system demonstration** (must be completed last, in a single session; –10% per re-try):

*Instructor certifies*

- Proper control action occurs
- Remote device fails in a safe state if the network becomes severed
- HMI shows a descriptive representation of all relevant remote device states and values
- A single *Experiment Journal* document containing all information about the system
- Schematic diagram challenge

## 14 Assessment

NAME: \_\_\_\_\_ DUE DATE: \_\_\_\_\_

Demonstrate various network diagnostic tools and techniques on a functioning PLC/HMI system. The instructor will randomly choose devices on the network to test, and to analyze data frames communicated between those devices. One of those chosen devices may be a personal computer connected to the network.

This exercise tests your ability to use diagnostic tools such as `ping`, `arp`, `netcat`, and/or `Wireshark`.

The following components and materials will be available to you: **functioning PLC/HMI network** in the lab room (you are welcome to use the network used by your PLC/HMI control system) ; **personal computer** with appropriate software installed ; and lengths of **Ethernet cable** with RJ-45 plug ends.

**Network device #1** (instructor chooses) = \_\_\_\_\_ (e.g. IP address)

**Network device #2** (instructor chooses) = \_\_\_\_\_ (e.g. IP address)

**Network device #3** (instructor chooses) = \_\_\_\_\_ (e.g. IP address)

Demonstrate the use of `ping` to test the connection of device \_\_\_, intentionally disconnecting that device during one test to show how `ping` responds when there is no connectivity.

Demonstrate how to identify the Ethernet MAC address of device \_\_\_ using network diagnostic tools.

Demonstrate how to filter only messages containing a certain IP address going to or from device \_\_\_ using packet-sniffing software.

Interpret the payload of a data frame sent from device \_\_\_ to device \_\_\_, using packet-sniffing software to do so. You are welcome to select which frame to interpret if there are many to choose from.

**SEQUENCE:** (1) Instructor identifies which devices you will perform your challenge on; (2) You configure and connect the necessary test equipment to the network; (3) Instructor observes you performing the tests and analyses.

Skill demonstration assessments are *mastery-based*, meaning every one must be competently completed in order to pass the course, and you will be given multiple opportunities to re-try if you do not pass on the first attempt. Each re-try requires new criteria (e.g. different circuit, different design parameters). Scoring is based on the number of attempts necessary to successfully design, build, and demonstrate a circuit (e.g. 1 attempt = 100% ; 2 attempts = 80% ; 3 attempts = 60% ; 4 attempts = 40% ; 5 attempts = 20% ; 6 or more attempts = 0%). Failing to follow instructions counts as an unsuccessful attempt and will require a re-try.

file we\_1131

## 15 Assessment

NAME: \_\_\_\_\_

DUE DATE: \_\_\_\_\_

Troubleshoot a fault within a *networked PLC/HMI control system*. This circuit shall be constructed in such a manner that all circuit components and simulated faults must be hidden from view (e.g. covering it up with a box or towel) but test points will be available for contact with a multimeter's probes. A schematic diagram showing the circuit and its test points will be allowed for use during the troubleshooting exercise.

Possible faults include:

- Any cable failed open
- Any cable failed shorted
- Any component failed open
- Any component failed shorted
- Any component value altered
- Any bit forced in the PLC
- PLC run/stop mode incorrect
- Error introduced into the PLC program
- Misconfigured channels/cards on the PLC
- Incorrect network device address

First, you will demonstrate that the system functions properly. Then the instructor will either set up or supervise other students setting up a random fault in that system (hidden from view) while you are out of the room. You will then have a limited amount of time to independently perform measurements and other tests while under the continuous observation of the instructor. A successful troubleshooting exercise consists of both correctly identifying the location and nature of the fault, as well as logically defending the necessity of each diagnostic step. Incorrect fault identification, unnecessary steps, and/or incorrect defense of any step will result in a failed attempt. Your only access to the faulted circuit will be via the test points, and only one unpowered test will be permitted.

If you must work remotely rather than in-person, the faulted system must be at the instructor's location while you request measurements and other diagnostic tests of the instructor via teleconferencing system (e.g. videoconference, telephone, text messaging).

Troubleshooting is *mastery-based*, meaning every one must be competently completed in order to pass the course, and you will be given multiple opportunities to re-try if you do not pass on the first attempt. Each re-try begins with another randomized fault on the same circuit. Scoring is based on the number of attempts necessary to successfully troubleshoot a circuit (e.g. 1 attempt = 100% ; 2 attempts = 80% ; 3 attempts = 60% ; 4 attempts = 40% ; 5 attempts = 20% ; 6 or more attempts = 0%).

file we\_1031

## 16 Lab clean-up

NAME: \_\_\_\_\_

This list represents all of the major work-items that must be done at every semester's end to prepare the lab space for the upcoming semester. Each student will have at least one task assigned to them.

### Non-technical tasks

- Thoroughly clean whiteboard(s) and all table/workbench surfaces
- Vacuum-clean the floor of all debris after picking any and all larger pieces
- Organize all wires, cables, power cords, and test leads neatly into their storage locations
- Vacuum-clean the exterior and interior of all electrical enclosures and cabinets
- Move test equipment to their proper places and clean their outer surfaces
- Note any depleted bins (electronic components, fasteners, terminals, cables, etc.)
  - Check boxes and spools of components to see if depleted bins may be re-filled from these
  - Report to instructor for re-ordering in preparation for next semester
- Note any depleted PCBs (e.g. fuseholder PCBs, proto PCBs, etc.)
- Power up each lab PC to check for routine software updates, then power down for the break

### Technical tasks

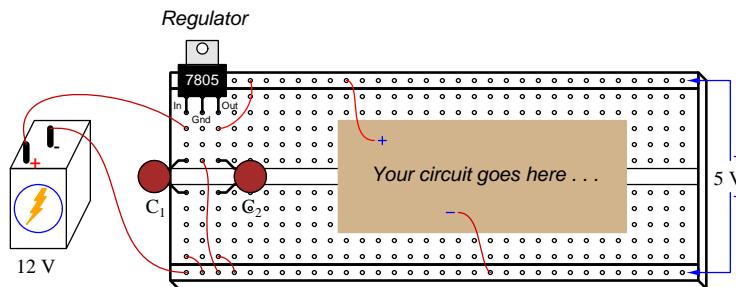
- Check fastener storage bins to ensure none are misplaced
- Check resistor storage bins to ensure none are misplaced
- Check fuse storage bins to ensure none are misplaced
- Check inductor/transformer storage bins to ensure none are misplaced
- Check capacitor storage bins to ensure none are misplaced
- Check semiconductor device storage bins to ensure none are misplaced
- Test power supplies and benchtop multimeters for basic functionality (e.g. all *power supply* voltage adjustments functional, current limits functional, voltage/current meters functional ; all *multimeter* all voltage ranges functional, all current ranges functional, overcurrent fuse good)
- Test oscilloscopes and signal generators for basic functionality (e.g. all *oscilloscope* channels functional, all vertical sensitivity settings functional, all timebase settings functional, triggering functions properly ; all *signal generator* waveshapes functional, magnitude and frequency adjustments functional)
- Test and clean all soldering stations (e.g. clean/replace porous and brass-wool cleaning pads, check tips for wear, ensure correct temperature settings and that the tip solders well)
- Test all lab power system (e.g. solar power grid, hand-crank generator) circuit breakers and fuses
- Test all special lab systems for basic functionality ; ensure schematic diagrams are in place
  - Digital clock, window crane system, window sensors, precision voltage reference
  - Ping all installed Ethernet addresses to ensure devices respond (e.g. PLC processors)

## General Circuit Design Tips

When designing and constructing circuits for experimental and prototyping purposes, the following tips are recommended for success:

- **Sketch a schematic diagram before constructing anything.** You need to have a clear understanding of what it is you intend to build before you begin building, in order to avoid major errors and hazards, and planning your build in schematic form is an excellent way to do that. Having a clear diagram in hand also aids others who you might wish to help you if things don't work as planned.
- **Build and test in stages.** If you try to build the entire system before testing it, you will very likely encounter multiple errors which will be more time-consuming to diagnose than if you took the time to build and test each portion of your circuit before building and testing the next portion.
- **Choose resistor values between 1,000 and 100,000 Ohms** unless there is some compelling design rationale for using a smaller or larger values. Reactance values within AC circuits should also fall within these same limits. Circuits built with low-value resistors tend to dissipate a lot of power when energized by constant-voltage sources, while circuits built with high-value resistors tend to exhibit "signal sag" when connected to loads and/or test equipment.
- **Use decoupling capacitors** connected in parallel with the DC power pins of every integrated circuit, to stabilize DC voltage for reliable operation. This is especially critical for high-speed digital circuits and sensitive analog circuits, where variations in DC supply voltage may compromise signal integrity. 1  $\mu$ F ceramic capacitors work well for this purpose, and should be located as close to each IC's power supply terminals as possible.
- **Diversify your learning experience** by using different types of test equipment (e.g. DMMs, VOMs, oscilloscopes), different types of construction techniques (e.g. solderless breadboards, soldered connections, terminal blocks), and different types of power sources. Remember, the reason you are in this course is to *learn*, not just to complete assignments!

When using sources of energy other than laboratory-quality power supplies, you may need to stabilize source voltage to ensure reliable circuit function. This is especially true when using chemical batteries, solar panels, and other electrical sources known for varying voltage output. A simple integrated circuit called a *three-terminal fixed voltage regulator* takes in power at some voltage larger than what your circuit needs, and dissipatively reduces the voltage level to a fixed value determined by the part number of the regulator IC. The popular LM78xx series of voltage regulators is recommended, where the last digits represent the fixed output voltage (e.g. 7805 = 5 Volts, 7812 = 12 Volts, etc.). An illustration showing how such a regulator might be installed in a solderless breadboard for general experiment/prototype use is shown here:



Decoupling capacitors  $C_1$  and  $C_2$  help stabilize the regulated voltage if your circuit's load current happens to pulse rather than be steady over time. Consult the regulator IC's datasheet for pin designations, recommended capacitor sizes, and also for general maximum voltage and current ratings.

## General Troubleshooting Advice

All electronic circuit faults fall into at least one of these categories:

- **Connection fault** – the components are not properly connected together.
- **Design flaw** – the circuit cannot work because something about it is incorrectly designed.
- **Lack of power/signal or poor quality** – the power and/or signal source is “dead” or “noisy”
- **Component fault** – one or more components is faulty.
- **Test equipment** – either the test equipment itself is faulty, or is not being used appropriately.

Of these categories, the one causing more problems for students initially learning about circuits than all the others is the first: *connection fault*. This is because the ability to translate an idea and/or a schematic diagram into a physical circuit is a skill requiring time to develop. Many such problems may be avoided by (1) drawing a complete schematic of what you intend to build before you build it, (2) marking that schematic to show which connections have been made and which are left to make as you are wiring it, and (3) using an ohmmeter (not your eyes!) to verify that every pair of points which should be connected are connected and that no points which should be electrically distinct from each other are in fact electrically common.

### *Troubleshooting strategies*

- **Verify the symptom(s)** – Always check to see that the symptom(s) match what you’ve been told by others. Even if the symptoms were correctly reported, you may notice additional (unreported) symptoms helpful in identifying the fault.
- **Use test equipment rather than your eyes to find problems** – your multimeter (and other tools) will show you things your eyes cannot see, and this will build habits and skills invaluable in situations where visual inspection is impossible.
- **Verify good power quality** – Is the source voltage within specifications, and relatively free of “ripple” and other noise?
- **Check signals at component terminals** – Use an oscilloscope or multimeter to check for proper signals at each of the component pins, to see if each one matches your expectations. An important check, especially for integrated circuits, is whether the measured output signal(s) are appropriate for the measured input signal(s).
- **Simplify the system** – If possible, re-configure the circuit to be as simple as possible, because complexity makes faults harder to find.
- **Take more and different measurements to get un-stuck** – if the source of trouble is still not apparent, take measurements in places you haven’t tried yet to obtain new data. Also, use different test instruments that will show you other facets of the signals, for example try using an oscilloscope to test a signal if you’ve only been using a multimeter since an oscilloscope will reveal things a multimeter cannot.
- **Swap identical components** – If particular a component is suspected of being faulty, and you are able to swap another (identical) component for it, do so to see whether or not the problem moves with the old component. If so, that component is to blame; if not, the problem lies elsewhere.
- **Always look for Root Cause(s)** – don’t declare success simply by finding the proximate (i.e. the most direct) cause, but continue your search to find what design flaw, circumstance, or other distal cause led to it.

## Functional and Conformance Testing

Engineering is the process of *designing to specification*. As such, every new design must begin with identifying those specifications and determining how to prove the design will meet or exceed each specification. Two different categories of tests apply to any design, *Functional Tests* and *Conformance Tests* (also known as *Compliance Tests* or *Type Tests*).

**Functional Tests** check whether the system fulfills its intended function(s). These tests focus on features, examples of which are listed here:

- A radio communication system's ability to both transmit and receive certain types of information, to provide the user with relevant data on the system's performance, etc.
- An engine's ability to output a certain minimum amount of horsepower.
- A computer's ability to process certain types of mathematical operations at certain minimum speeds, to execute functions unique to that design.
- A security system's ability to identify authorized versus unauthorized personnel, to log access data.
- An electric motor's ability to achieve an advertised energy conversion efficiency.
- A switch's ability to open and close an electric circuit on demand.
- The display(s) and controls for an electronic system operate consistently and predictably.

**Conformance Tests** check whether the system complies with third-party regulations, safety standards, reliability requirements, etc.

- A radio communication system's ability to transmit only the intended frequency(ies) and to not exceed FCC-regulated power output.
- An engine's ability to operate while outputting no more pollutants than allowed by the EPA.
- A computer's ability to process industry-standardized data types, to not radiate or conduct high-frequency signals that could interfere with other electronic devices.
- A security system's reliability as measured over a specified range in time.
- An electric motor's resistance to electrical ground faults, its ability to operate while not exceeding a maximum specified amount of acoustic noise.
- A switch's physical dimensions agree with NEMA standards, can safely interrupt rated current, has a certain minimum dielectric strength, etc.
- The display(s) and controls for an installed system being operable by someone restricted to a wheelchair (i.e. Americans with Disabilities Act "Standards for Accessible Design").

When selecting and designing your own project, you will be asked to generate a list of testable criteria which will become the foundation of your project's Functional and Conformance Tests. After selecting your project concept, you will decide what it must do (Functional) and identify relevant safety, quality, reliability, and/or interference concerns (Conformance). Functional criteria are more or less arbitrary, but Conformance criteria are objectively-based.

The following pages list industry best-practice standards and recommendations with source references to external documents. If you are at a loss for Conformance criteria in your project, this is an excellent reference. A well-designed Conformance Test reliably measures its criterion, whatever that may be.

## Construction Standards

The following list contains best-practice standards applicable to most electrical and electronic projects:

### General layout

- All components shall be mounted securely where applicable.
- All electrical components shall be located to avoid accidental exposure to liquids.
- All manual controls (e.g. buttons, handles, knobs) shall be accessible, clearly labeled, and function without undue effort.
- Fragile components (e.g. heat-sensitive semiconductors) shall be easily accessed for replacement.

### Fastening

- All threaded fasteners shall be properly engaged and tightened.
- A minimum of 1-1/2 threads shall extend beyond the threaded hardware (e.g., nut), unless specified otherwise.
- All cable ties shall be trimmed off, flush with the back end of the strap head. (*NASA-STD-8739.4, NFPA 79 2007 edition (13.1.5.6)*)

### Thermal considerations

- Power-handling components shall have adequate cooling capacity, usually in the form of a heat sink.
- All components expected to run hot shall be located in such a way that their heat does not affect the function or longevity of any other components.
- Components dissipating heat in quantities of 1 Watt or greater, or in quantities sufficient to damage a PCB shall be mounted with sufficient standoff [ $> 1.5$  mm (0.060 in)] and shall be mechanically restrained.

### Power wiring

- All electrical sources greater than 24 Volts shall be guarded against accidental contact (e.g. use recessed terminals with no exposed metal).
- All electrical sources capable of generating currents exceeding conductor ampacity ratings shall be overcurrent-protected, regardless of voltage.
- Overcurrent protection shall be on the ungrounded (“hot”) conductor(s) only (*NFPA 70 2017 edition (240.15(A))*). No grounded conductor shall be overcurrent-protected or switched (*NFPA 70 2017 edition (240.22)*).
- Small power transformers shall be overcurrent-protected on their primary windings, and optionally on their secondary windings. For primary currents less than 2 Amperes, exclusive primary overcurrent protection may be as high as 300% of full-load rating, 250% if secondary overcurrent protection also included. For secondary currents less than 9 Amperes, overcurrent protection may be as high as 167% of full-load rating. (*NFPA 70 2017 edition (450.3(B))*)
- All metallic panels and electrical enclosures receiving power from the AC line shall be bonded to earth ground for safety, and this bonding verified by electrical resistance measurement. Resistance between nearest facility ground point (e.g. plug ground prong) and chassis shall be  $0.1 \Omega$  or less. (*NASA-STD-4003A, NFPA 79 2007 edition (18.1)*)

- All power conductors shall be strain-relieved so that tension applied to them will not stress the electrical connections themselves (*NFPA 79 2007 edition (13.4.3.1.1)*). Permanent conductors not in a raceway shall be securely fastened at least every 6 inches using cable ties or other appropriate means.
- All conductors shall be prevented from chafing against any sharp edges (*NFPA 79 2007 edition (13.5.1.2)*), and this includes installing bushings on all electrical fittings.
- All power conductors shall be properly colored according to American wiring conventions (e.g. red and black for DC + and -, black and white for AC “hot” and “neutral”, green for earth ground) (*NFPA 79 2007 edition (13.2)*). Colored tape is permissible at both ends of a wire whose insulation is not of the correct color.

### Other wiring and connections

- All other conductors shall have sufficient ampacity and insulation voltage ratings for their application (*NFPA 79 2007 edition (12.5)*).
- Compression terminals – crimping of solid wire, component leads, or stranded wire that has been solder-tinned, is prohibited. The conductor shall extend a minimum of flush with, and a maximum of one (1) wire diameter beyond the conductor crimp edge. No protruding wire strands outside the terminal barrel. (*NASA-STD-8739.4*)
- Compression-style wire splices are prohibited between terminals (*NFPA 79 2007 edition (13.1.2.1)*).
- Solid wire is prohibited where wire motion is possible, to prevent metal fatigue from vibration and other mechanical stresses. Only stranded wire shall be used for unrestrained wire runs, wires running between components not rigidly mounted to each other or to the same rigid frame, etc. (*NFPA 79 2007 edition (12.2.4)*)
- Only solid wires shall be wrapped around screw terminals, and this direction shall be clockwise. Wrap distance shall be between  $180^\circ$  and  $270^\circ$  (between  $\frac{1}{2}$  and  $\frac{3}{4}$  turn).
- Attached wires shall withstand being lightly pulled with fingers.
- All wire insulation shall be intact (i.e. no bare wires anywhere).
- After insulation removal, the remaining conductor insulation shall not exhibit any damage such as nicks, cuts, or charring. Conductors with damaged insulation shall not be used. Scuffing from mechanical stripping or slight discoloration from thermal stripping is acceptable. (*NASA-STD-8739.3*)
- All exposed wiring shall be generally neat and not messy in appearance.
- Multiple conductors extending beyond an enclosure or panel shall be bundled together as a multi-conductor cable wherever possible, unless separation is necessary to avoid undesired signal coupling between conductors.
- Solderless breadboards are prohibited for permanent assemblies; acceptable only for prototyping.
- Tape is prohibited as electrical insulation; heat-shrink tubing shall be used instead.
- Cables shall not be bent below the minimum recommended inside bend radius (6 diameters for flexible coaxial cable, 3 diameters for multi-wire harnesses 10 AWG and smaller). (*NASA-STD-8739.4*)
- Cables must contain enough extra length (*service loops*) to permit future re-termination, maintenance, and other physical adjustments made to the system.

## Soldered connections

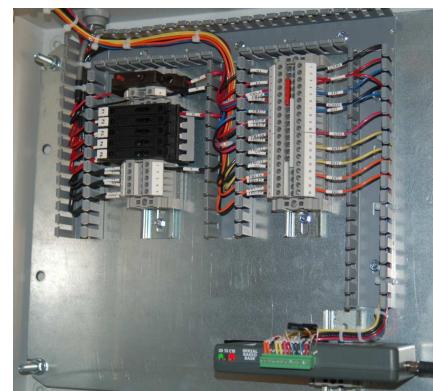
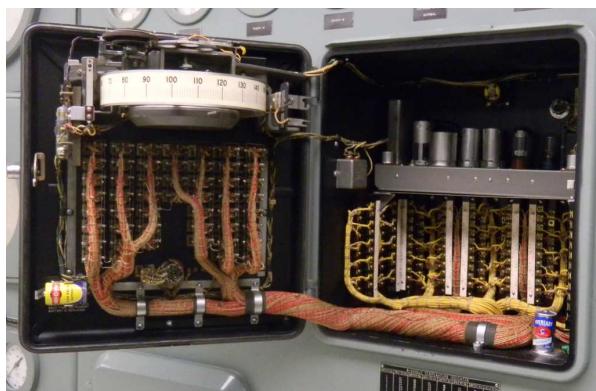
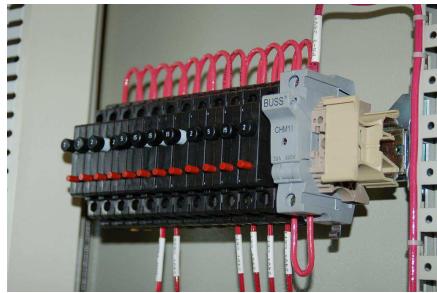
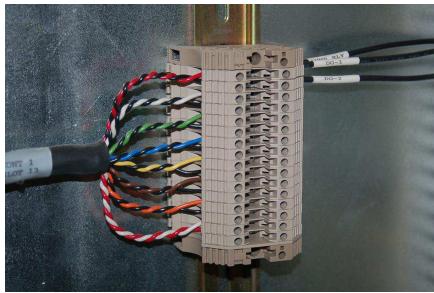
- Visual Appearance – the appearance of the solder joint surface shall be smooth, nonporous, undisturbed and shall have a finish that may vary from satin to bright depending on the type of solder used (*NASA-STD-8739.3*). Overheated solder has a dull, gray, frosty and/or crystallized appearance (*NASA-STD-8739.2*).
- Solder Coverage – the molten solder shall flow around the conductor and over the termination areas. (*NASA-STD-8739.3*)
- Tinning – tinned surfaces, which are to become part of the solder termination, shall exhibit 100% coverage. When tinning stranded wires, the solder shall completely wet the conductor, penetrate to the inner strands, and exhibit 100% coverage. Wire strands shall remain distinguishable. Wicking of flux or solder shall be minimized. (*NASA-STD-8739.3* and *NASA-STD-8739.4*)
- Minimum Insulation Clearance – the insulation shall not be embedded in the solder joint. The contour of the conductor shall not be obscured at the termination end of the insulation. (*NASA-STD-8739.3*)
- Maximum Insulation Clearance – the insulation clearance shall be less than two wire diameters, including insulation, but in no case shall permit shorting between adjacent conductors. Insulation clearance shall be referenced from the first point of contact of the conductor to the terminal. (*NASA-STD-8739.3*)
- Mechanical Support and Strain Relief – wire bundles shall be supported so that the solder connections are not subjected to mechanical loads. Conductors shall be provided with sufficient slack to preclude tension on the solder termination or conductor. (*NASA-STD-8739.3*)
- Through-hole component leads and conductors terminated straight through a PCB shall extend 0.5 mm (0.020 in) to 2.29 mm (0.0900 in.) beyond the pad surface. Leads may be bend up to 30° from the vertical plane to retain the part during soldering. (*NASA-STD-8739.3*)
- Component bodies shall not be in contact with soldered terminations. (*NASA-STD-8739.3*)
- The radius of a bend in the lead of a component shall not be less than the lead diameter or lead thickness. (*NASA-STD-8739.3*)

## Circuit design

- Decoupling capacitors – every integrated circuit should have a decoupling capacitor of at least 0.1  $\mu\text{F}$  connected closely to its DC power terminals, for the purpose of stabilizing DC voltage at each IC. The same is true for loads generating noise, such as DC electric motors.
- Separate power sources – when possible, separate power sources should be used to energize high-current loads versus low-current control circuitry, in order to avoid placing noise caused by the high-current loads onto the control circuit's power rails.
- Star-point power distribution – circuits that might interfere with one another through a shared power source should have separate power conductors routing to that source.
- Keep high-current loops small – conductors carrying high currents should run closely parallel to each other in order to minimize the “loop area” and thus minimize magnetic field creation.
- Twisted-pair, shielded cable – sensitive signals should be protected from crosstalk by electrostatic shielding (grounded at one end only) and by twisting conductors (to minimize magnetic coupling).
- Conductor fault tolerance – circuits linked by plug-in conductors should be designed to assume safe states if disconnection occurs.

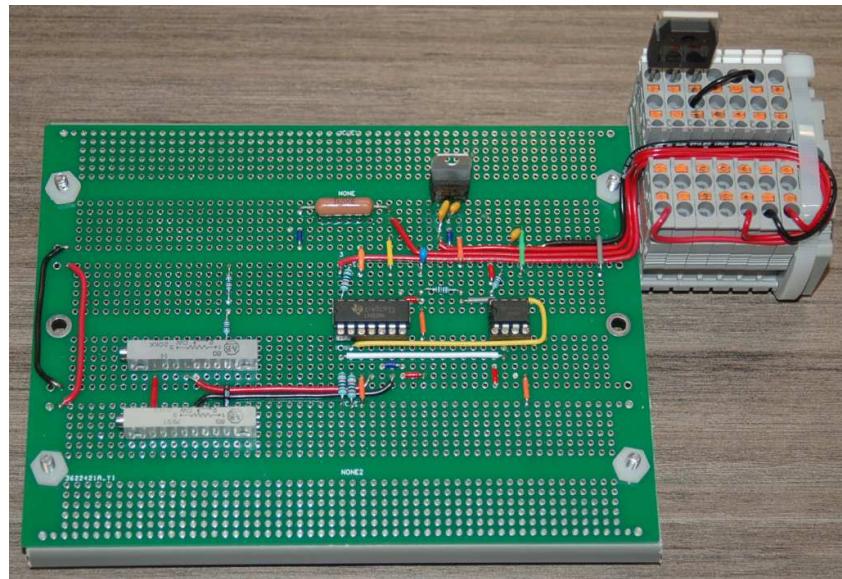
- Consistent control directions – clockwise knob rotation and upwards switch/slider should result in “more” or “on” actions.

### Exemplar wiring (professional)



### Exemplar wiring (student)

PWM pulse generator circuit constructed by Joe Archer in Fall 2023:



file wp\_standards