

1-1-2001

Factory Automation: A Controls Course for Every University

Kelvin T. Erickson

Missouri University of Science and Technology, kte@mst.edu

Follow this and additional works at: http://scholarsmine.mst.edu/ele_comeng_facwork



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

K. T. Erickson, "Factory Automation: A Controls Course for Every University," *Proceedings of the 2001 American Control Conference, 2001*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2001.

The definitive version is available at <https://doi.org/10.1109/ACC.2001.945879>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

Factory Automation: A Controls Course for Every University

Dr. Kelvin T. Erickson

Professor of Electrical & Computer Engineering
134 EECH, University of Missouri-Rolla, Rolla, MO 65409
kte@ece.umr.edu

Abstract

The majority of new control engineering positions is now in manufacturing and involves programmable logic controllers (PLCs). This paper describes two courses developed at the University of Missouri-Rolla to satisfy this demand. Both courses present the subject of programming PLCs with an emphasis on the design of the programs. Both courses contain an integral laboratory component that solidifies the concepts presented in the lectures. The philosophy and pedagogical features of the two courses are first described, followed by the structure of the lecture and the laboratory exercises.

Introduction

The field of automatic control has been undergoing a transformation over the past twenty years. Twenty years ago, the typical control engineering graduate (ChE, EE, or ME) had a course in feedback control theory and those interested in a career in control secured a position in the aerospace or chemical industries. In the last 20 years, the number of new control engineering positions in the aerospace industry has declined while there has been an increasing emphasis on manufacturing automation in order to cut the cost of production and/or increase product quality. Consequently, the number of control engineering positions in manufacturing has been dramatically increasing to the point that the majority of new control engineering positions is now in manufacturing and involves programmable logic controllers (PLCs). The typical college or university has been slow to recognize this trend. Many universities have devoted a portion of a course or laboratory to PLCs and a few universities have entire courses devoted to PLCs. In this author's opinion every university that teaches control system courses should have at least one course devoted to PLC programming. This paper describes two courses at the University of Missouri-Rolla. The first course covers the basics of PLC ladder logic programming, including PID control. The second course covers other PLC languages and factory communications and culminates in a class-wide project configured and run like an industrial project. Both of these courses have a prominent laboratory component. The content of both courses has evolved over the past 10 years based on the author's experience and alumni suggestions.

The particular courses described in this paper are taught in the Dept. of Electrical and Computer Engineering. However, enrollment is open to other disciplines. In addition, a short course version of the course appeals to students who do not have the time or the desire to have an entire course on PLC programming.

Philosophy

Both courses present the subject of programming a PLC with an emphasis on the design of the programs. Beyond teaching one how to program the PLC in its languages, the courses also cover

the more general problem: "Given a set of operational specifications, how does one develop the PLC program?" The first course addresses these topics but with the ladder logic programming language. The second course extends the material to cover other languages and more sophisticated applications. Both courses present the design process: the tasks involved, breaking the program into manageable pieces, standard code for the various parts, and handling the sequential parts of the problem.

Because of its current popularity, ladder logic is the language used for the majority of the courses. The industry trend is toward using the IEC 1131-3 (now IEC 61131-3) standard, and so it should be the primary language. However, IEC 1131-3 is only a voluntary standard and so individual manufacturers have some freedom in the implementation. The Allen-Bradley Logix5000, Modicon Quantum, and Siemens S7 PLCs all implement the 1131-3 standard. Because of their large installed base, Allen-Bradley PLC-5/SLC-500 PLC languages may also be covered.

Since a typical manufacturing plant may contain discrete, continuous, and batch processes, all of these applications are treated in both courses, although the emphasis tends to be on discrete and continuous processes.

Throughout, the courses contain example problems demonstrating good design practice. In addition, these problems are worked with each PLC covered in the class.

Pedagogical Features

Both courses have the following pedagogical features

- An emphasis on design and not just on teaching the programming language.
- Most of lecture time is spent working through example problems.
- The laboratory exercises are an integral part of the course.
- The laboratory exercises are small versions of real processes and involve real equipment, not just simulations.

First PLC Course

The objectives of the first course are to teach:

- PLC ladder logic programming language
- Approach to sequential problems
- Good program design practice
- Simple PID control tuning
- Introduction to sensors and actuators
- Introduction to factory communications

The topics in the first course are outlined in Table 1. This course has two hours of lecture and two hours of lab each week (total 3

semester-hours of credit). For the first course, it is important to get the student into the laboratory as soon as possible. By confining the material to the one (or two) PLCs in the laboratory, basic ladder logic and PLC wiring should be covered in four hours of lecture. To meet this goal, the laboratory does not meet during the first week of class and each lecture in the first week of class is extended by an additional half-hour.

Many examples are used throughout the course. After briefly presenting a new topic, it is illustrated by working through examples. Also, good design practice is emphasized through examples and showing examples of bad design practice.

Table 1. Outline of First PLC Course Lectures

Topic	Hours
Introduction to factory automation and PLCs	1
Basic ladder logic	2
PLC memory, discrete I/O and wiring	2
Timers, counters	2
Sequential applications	3.5
Troubleshooting	0.5
Exam 1	1
Tour of local industry	1
Analog I/O	0.5
Comparison, arithmetic instructions	4.5
Parallel operations	1
Standard project code, SFC language	1
Exam 2	1
PID control	6
Sensors and transducers	2
Exam 3	1
Introduction to factory communications	1

Laboratory Exercises

The first laboratory exercise should take about three weeks. By this time, the lectures should have covered sequential applications. For the remaining laboratory exercises, the student teams rotate among the other four laboratory exercises (two two-student groups per exercise), spending three weeks on each exercise. The students should be getting more efficient by the end of the semester, so there are only two weeks for the last laboratory exercise.

Currently, three PLCs are used in the laboratory exercises: Allen-Bradley Logix 5000, Allen-Bradley PLC-5, and Modicon Momentum. The Allen-Bradley PLCs are very similar in instruction set and programming, and are for the most part treated identically in the course, so the students really only have two PLCs in the laboratory. The exercises are scheduled so the students switch from the Modicon to the Allen-Bradley or vice versa only once in the semester. For the HMI exercise, the Allen-Bradley PLC-5 controls the equipment, but that is not very relevant to the students, since they are only constructing the HMI screens that interface to the PLC and its memory locations.

Following are descriptions of the current laboratory exercises and their educational aspects. Except for the first exercise, the exercises are periodically changed. Exercises that have been used in the past are described in Erickson (1993) and Erickson (1996).

First Laboratory Exercise

The first laboratory exercise is intended to familiarize the students with the basic ladder logic instructions, the PLC programming software, and simple sequential applications. The students wire a simple switch and light board to the particular PLC. Although most of them will not be doing wiring after graduation, they will often check wiring done by a technician, so exposure to wiring is important. The ladder logic exercises are:

1. Contact instructions in series (logical AND)
2. Contact instructions in parallel (logical OR)
3. Start/stop with momentary push buttons
4. Simple timer
5. Simple counter
6. Flashing lights
7. One-shot pulse of 5 seconds duration
8. Turn-signal (like Ford Thunderbirds)
9. Cereal box filler (solution worked in lecture)

Level Control

This exercise, shown in Fig. 1, is designed to provide a simple operator interface and simple analog level control application for a simple tank system shown. The outlet flow is the demand, over which the PLC has no control. The tank level is controlled by adjusting the position of a valve which is fed by a pump that draws water from a reservoir. You will need to program the PLC to maintain the desired level. The operator interface panel consists of start and stop pushbuttons, controller on/off lights, high and low level alarm lights, two LED displays each showing a two-digit number, a rotary switch that selects one of the alarms, two small pushbuttons to change the selected alarm, and a thumbwheel switch used to input the desired level. The rotary switch selects one of two level alarms: high and low. The selected value is then displayed on the lower LED display. The small pushbuttons are used to increase/decrease the selected value by 1 every time the pushbutton is pushed. If the pushbutton is held down, the selected alarm value should increase/decrease by 1 every second. The upper two-digit LED display shows the current level, in tenths of an inch. In order to save output channels, the two displays are multiplexed, that is they share the same 8 channels that define the value to be displayed and two select lines are used to "latch" the value into the appropriate display. The functions of the operator interface panel must be programmed in the PLC because the only connections to the operator interface are switch contacts and direct lines to the LED displays. This particular process was designed by the author and built at UMR.

Conveyor Assembly and Inspection

The conveyor assembly process (Fig. 2) is the most complicated sequential process in the laboratory. This exercise is designed to provide a conveyor operation that consists of sorting, assembly, and inspection of the assembled part. At first glance, the problem seems overwhelming (given the time constraints), but this process provides the students with experience in decomposing a large problem into smaller subproblems. The particular unit is a commercially-built unit by Bytronic, Inc.

The process is described in Erickson (1996) and so the educational aspects of the problem are discussed here. The students are guided to break the problem into manageable subproblems and tackle each one separately:

1. Sorting area – direct the plastic rings down the chute into the assembly area.
2. Managing the assembly chute – The number of rings in the chute must be tracked. If four rings are in the assembly chute, no more must be ejected down the chute. If the assembly area is empty, and there is a ring in the chute, then a ring must be allowed to drop in the assembly area. Various delays must be used to account for the initial ring to slide into position, etc.
3. Check diameter of assembled component – When a part is detected at this station, a clamp is extended to hold the part in position and gauge the diameter. There is a slight problem with the IR sensor that detects the presence of a part. When the part is clamped, the sensor output turns off. The students must account for this problem in their ladder logic.
4. Defect tracking – Various sensors check for the presence of the metal peg and the plastic ring. Reject lamps track the progress of a defective part down the conveyor. When a defective part reaches the reject solenoid, it is pushed off the conveyor. The use of a shift register to track defects does not work because the spacing between assembled components is not uniform.

The process of breaking up the larger problem into smaller pieces is difficult for many students. Most of them have little experience in this part of the design process. With some guidance by the instructor, they soon understand this part of the design process.

Pick-and-Place

This exercise is designed to provide a simple sequence control application for PLCs that uses servo and stepper motors with commercial motor drivers and position encoders. This exercise also provides experience with the Allen-Bradley Control Logix motion module, which is tightly integrated with the PLC processor. The tri-axis pick-and-pace robotic cell is shown in Fig. 3. The robot transfers a part from one place on the table to another. The position of the gripper is determined by counting pulses from optical position encoders on each axis. The home position for each axis is determined with limit switches. Before each process cycle begins, the system waits on an externally generated start signal indicating that a part is in fact ready to be moved. On receiving this signal, the robot begins its cycle at the home position, moves to the pick-up position, picks up the part, moves it to the drop-off location, and moves back to the home position.

The robot system is built using a commercial XYZ table and gripper from Arrick Robotics and relies on commercial drives to control the four motors. The X and Y axes are controlled by the M02AE 2-axis motion module which controls two Allen-Bradley drives/motors. These two axes are controlled with special motion instructions which largely hide the details of the drive/motor/encoder system. The Z axis and the gripper are controlled with commercial stepper motor drives. For these two drives, there is not a motion module and so the details of the control of the motor are handled by the ladder logic.

The application is a straightforward application of sequential control. Each move is treated as one step. For example, the first move starts from the home position and moves along the +Y axis until the instruction that moves the Y axis motor 10.5 revolutions is complete. A move along the Z axis is defined by a certain number of pulses being counted. So, a transition between steps occurs at

the completion of a motion instruction, a certain number of pulses, or on a limit switch (when returning to the home position).

This application involves commercial servo control, which interfaces to the PLC processor using special motion instructions. In order to make the process easier, the students are led step-by-step through the configuration and auto-tuning of the X and Y servos and run a short sample program. This program serves as the basis of the program needed by the full exercise.

Human/Machine Interface

In this exercise, a simple human-machine interface (HMI) for an automated storage/retrieval system (AS/RS), shown in Fig. 4, is programmed. The HMI includes animation, buttons and numeric fields. The RSView package by Rockwell Software is used in this exercise, but the concepts apply to any commercial HMI package. For the AS/RS, the student constructs a window displaying information to the operator and allowing the operator to control the AS/RS, shown in Fig. 5. The RSView software package is used to build the window by constructing and defining the various objects on the screen and linking them to a variable/memory location in the PLC.

Second PLC Course

The objectives of the second course are to teach:

- Other PLC programming languages (sequential function chart, function block diagram, structured text)
- Large control system projects
- PID cascade and feedforward control
- Factory communication networks

The topics in the second course are outlined in Table 2. As for the first course, this course has two hours of lecture and two hours of lab each week. Since the students already have programming experience from the first course, the laboratory exercises start immediately.

Also, the pace of the lecture is generally faster than the first course. Since all of the students have prior programming experience, there is very little instruction. Most of the lectures are spent working examples. The students are heavily exposed to standard industry practices. For example, simulating a process using the available PLC without any physical I/O is covered early and used throughout the course. This practice of testing a program before installing it in the real process is common in industry. Also, the students practice programming and troubleshooting their programs on a remote PLC. Many of the homework exercises involve programming and testing the solution on one of the lab PLCs, including simulating the physical process being controlled. These exercises are not done as part of the lab, so the students access the lab PLCs over the department Ethernet network. The students thus experience the process of remotely programming/testing a PLC, another common industry practice. It is not uncommon for an engineer in his/her office 200-km from the actual process to modify the PLC program while the process is in operation.

A class-wide project is a significant part of the course. This exercise is modeled after the author's experience in industry and involves multiple student teams. The process involves multiple

interconnected units, typically chemical. For example, the process consists of units such as raw material storage/handling, blending, reacting, ion exchange, product storage, and product loadout. Each team is assigned one unit. To do this exercise effectively, there are multiple PLCs that can communicate with each other and multiple computers for the HMI screens. Each team has its own PLC and HMI computer. The project involves some coordination between PLCs so that PLC-to-PLC communication is required. The students are also required to simulate the actual process in the PLC. The students are given a narrative that describes the operation and control equipment of each unit. The narrative includes a piping & instrument diagram (P&ID) for each unit. In addition, the students are required to follow a programming guideline (like a corporate standard) that dictates how sequential operations, alarms, motors, valves, and PID loops are coded in the PLC and in the HMI device. Each team must also program an HMI screen for their unit.

Factory communication networks are presented in a different manner. The students are each assigned one factory communication network protocol to research and present their findings to the class. The network protocols researched by the students are: Interbus-S, Seriplex, Controller Area Network (CAN), DeviceNet, Foundation Fieldbus, Profibus DP, LonWorks, Data Highway +, Modbus +, TIWAY, ControlNet, Ethernet. Their presentation should address the physical characteristics of the network (topology, medium, distance, baud rate, etc.), transport mechanism (master/slave, peer to peer, arbitration, message size, etc.), history, applications and open standard or vendor-specific. For each of these networks, information is available from textbooks, vendor information, and the World Wide Web.

Table 2. Outline of Second PLC Course Lectures

Topic	Hours
Assembling a system, review of communications	2
Process simulation with PLC	2
Other IEC languages (SFC, ST, FBD)	8
Advanced PLC instr. (sequencer, shift reg., FIFO, LIFO)	5
Exam 1	1
Working with large projects (partially concurrent with lab exercise)	3
Safety considerations	2
Factory communication networks	4
Cascade, feedforward PID control	2
Exam 2	1

Laboratory Exercises

As with the lecture part of the course, the laboratory exercises cover more advanced topics. One of the exercises is a reworking of a laboratory exercise from the first course, but using one or more of the other PLC languages. Another exercise involves assembling and initially configuring all of the PLCs in the laboratory (currently more than 30), including setting up multiple communication networks. Other exercises involve troubleshooting and correcting an existing program. One of the exercises involves communication between multiple PLCs.

The laboratory for the second course uses the same three PLCs as in the first course, plus the Allen-Bradley SLC-500. The students are expected to be reasonably proficient on all of the processors, and so the students switch between PLC processors multiple times during the semester. In addition, the students use both RSView and InTouch for the HMI exercises.

Set up laboratory

The students set up the PLCs in the laboratory using a network drawing (Fig. 7), station layout, and PLC setup information. The students are literally presented with the PLC parts (chassis, power supplies, processors, input/output modules) and expected to assemble and configure the PLCs according the specifications. This is a group project, so each student is not required to do everything, only that each one be an active participant in the process. The lab is set up progressively so that each lab session builds on the previous one.

The students use the documentation provided (installation instructions, "Getting Started" manuals, etc.) to set up and configure each piece of equipment and to set up the network. The individual modules are provided with their factory default settings, which is generally different than desired. Therefore, the students must deal with DIP switch and/or software configuration of the hardware.

As part of the process, the students install at least three networks in the laboratory for PLC-to-PLC communication: Data Highway+ (Allen-Bradley), ControlNet, and Ethernet. They use these networks in the next exercise.

This laboratory exercise was developed at the specific request of companies who hire our students.

Factory Communications

This exercise is designed to provide experience in programming communications between Allen-Bradley Logix5550, PLC-5, and SLC-500 PLCs. Communication between Modicon Quantum and Momentum PLCs is also programmed. Also, the communication network performance will be measured.

Initially, two PLCs are programmed to exchange 100-word blocks of memory on one of the networks. Each PLC reads the other PLC's memory every 0.5 seconds. Then, the programs are modified to construct a heartbeat that is monitored by both of them. If the heartbeat is lost for 3 seconds, both PLCs should turn on an alarm indication. The alarm indication should turn off as soon as the heartbeat is detected.

Next, the students are instructed to use the PLC to determine the average delay to get a message from one PLC to another on the same network segment. As part of the exercise, the students must determine how to make this measurement. After that is accomplished, the students measure the delay when the message must travel across a gateway or router to a PLC on another network segment.

Troubleshooting

In this exercise, the students are given a PLC connected to a process. They are given the problem description and a program

that does not work. The particular problems and processes are generally from laboratory exercises not used previously by the students. The supplied programs are back-up versions of a program that was saved before the program was completely debugged. Sometimes, the instructor starts with a working program and then purposely introduces multiple errors in the code so the program functions improperly or not at all. The students are required to find and correct the errors.

Class-wide Project

The general structure of this project is described above. One unique aspect of this project is that the students each work on their part of the problem, and then the entire class meets to commission the entire system within a fixed time (2 hours). One discovers which groups have thoroughly debugged their code!

PID Control

The pH neutralization setup is indicative of some of the processes found in chemical processing industries and is used to provide an application in split-range and cascade control. The experimental setup is self-contained on a roll-around cart (Fig. 6). There are three inlet tanks: mild acid (vinegar), mild base (dilute baking soda), and an unknown chemical (mild acid or base). All three streams are pumped (through a valve, to regulate the flow) into a mixing tank. The unknown chemical stream is assumed to come from a waste process and the PLC does not control the flow or the pH of this stream.

A number of operations must be done automatically to start up the process, described in Erickson (1996). Under normal operation the control objectives are: 1) use the mild acid and mild base to neutralize the pH in the mixing tank; and 2) control the level of the liquid in the mixing tank to 10 cm by adjusting the outlet valve. The pH process is nonlinear, but rather than implement an adaptive gain algorithm, the controller gain will be constant since, the system will be operating in a narrow range. The pH controller output signal manipulates the acid and base flow controllers (split range control). Both flows should not be on simultaneously, since that would represent a waste of material. The system should control the pH regardless of the changing flow and pH of the unknown inlet stream (system disturbances). A hand valve manipulates the flow of the unknown stream and the pH is manipulated by pouring a different chemical (mild acid/mild base) into the tank of unknown chemical.

The control of the pH and level is accomplished with PID controllers. The output of the pH controller is converted to an acid flow loop setpoint and a base flow loop setpoint. The students must program the PID controllers in the PLC and tune the controllers.

Short Course Version

Some students do not have the time or the desire to have an entire course on PLC programming. In addition, non-EE students, most notably chemical engineering students, realize that the knowledge helps when looking for a job. Therefore, a short-course version of the first course is taught once or twice per year. This course has about 10 hours of lecture and 14 hours of laboratory time and covers about 1/3 of the first PLC course topics. It is taught during

two successive weekends so there is minimal disruption to student schedules.

Acknowledgments

Financial support from Rockwell Automation and the National Science Foundation (EID-9050822 and DUE-9750737) is gratefully acknowledged.

References

- Erickson, K. T. (1993). Experiments for an Undergraduate Automation Laboratory. *Proc. 1993 Amer. Control Conf.*, vol. 2, pp. 1731-1736, Omnipress.
- Erickson, K. T. (1996). Innovative Experiments for Undergraduate Factory Automation. *Proc. IFAC World Congress 1996*, vol. G, pp. 99-104, (paper 4c-03 6), San Francisco, USA, June 30 - July 5, 1996.

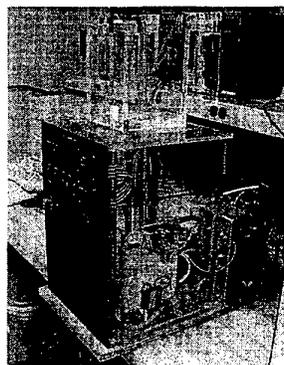


Fig. 1. Level Process

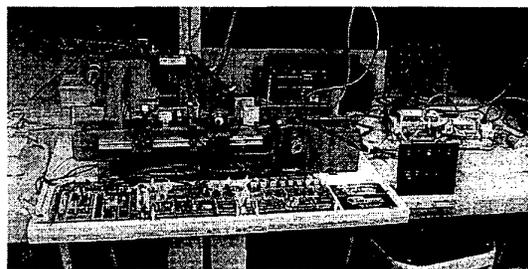


Fig. 2. Conveyor System

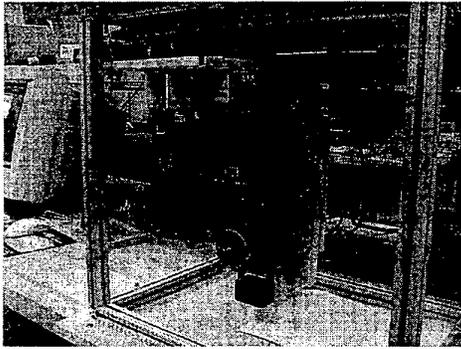


Fig. 3. Pick and Place.

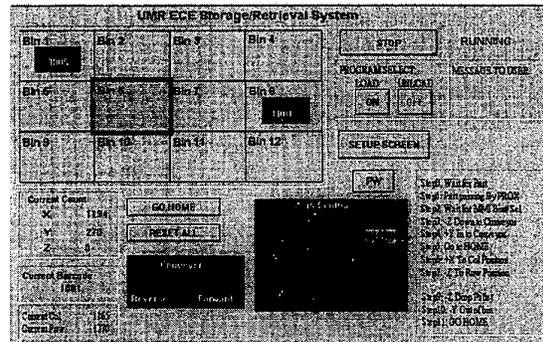


Fig. 5. AS/RS Screen.

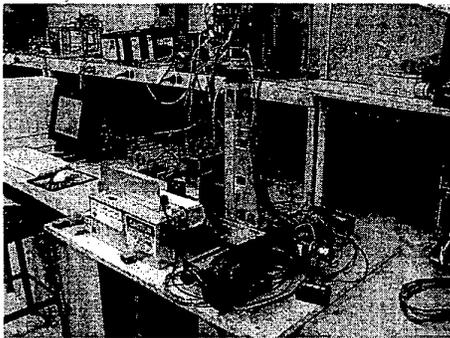


Fig. 4. Automated Storage/Retrieval System



Fig. 7. pH Process.

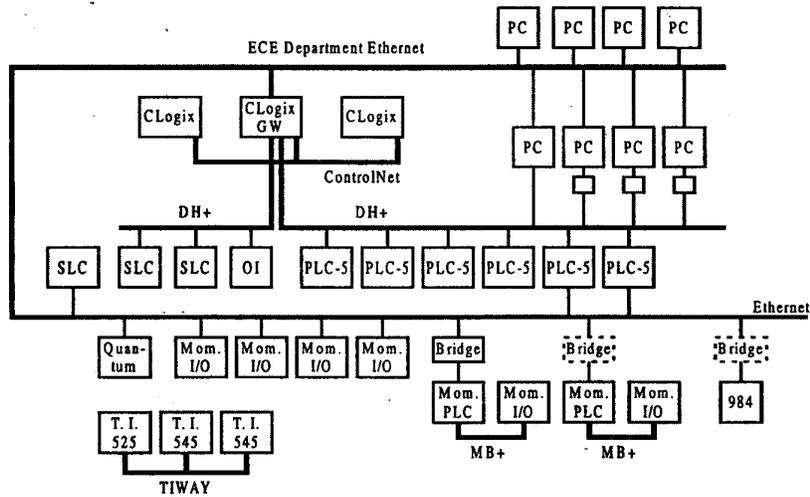


Fig. 7. Factory Automation Laboratory Network.