Experiences of a PLC Course Taught to EE and EET Students in Tandem

William T Evans, PhD, PE
Nicole Winhoven-Kamm
Department of Engineering Technology
Electrical Engineering Technology
University of Toledo
Toledo, Ohio 43606
Email: william.evans@utoledo.edu
nicole.kamm@utoledo.edu

Abstract

When a fellow professor decides to leave town on a sabbatical, some adjustments are expected. In this case, the EE professor handed a course to the EET professor to teach while on sabbatical. The course is EECS 4420, Programmable Logic Controllers. The EET professor also teaches a similar course during the same semester, EET 4550, Programmable Controller Applications. It was decided to teach both courses together in tandem. This paper discusses the plusses and minuses of this arrangement.

The discussion will include areas to be covered, lab or project assignments and tests. Interaction between the various student groups will be noted. The observations and conclusions will be preliminary since the class will be on-going during the Spring semester, 2015.

As of the end of registration, each group contained 16 students with one graduate student enrolled. It is anticipated that the groups will work together and have similar results since the course is predominately a hands-on course. It will be interesting to share observations from the class near the end of the semester as well as keep an eye for other courses that potentially would benefit from a common lecture/lab environment.

Introduction

The need to provide a course to EE students became a request for the EET instructor to teach both students in the EET program and include EE students. The course was a course in PLCs (Programmable Logic Controllers). The EET students are required to take two courses in PLCs. EE students are offered their PLC course as an elective. The second of two courses in the EET program was chosen to incorporate both EE and EET students. Most students were senior level. The enrollment in the class is 16 students from EECS 4220 (Programmable Logic Controllers), 16 students from EET 4550 (Programmable Controller Applications) and three students from EECS 6980 – Special Topics in Electrical Engineering.

Most EECS students were somewhat familiar with PLCs since their co-op experiences had included programing PLCs on the job in an industrial environment. They are very motivated and had an interest in furthering their PLC experience. The EET students came from one of two backgrounds, either from having taken the required sophomore course EET 2410 or from a course taken elsewhere and transferred to the university. Since PLC courses vary widely from school to school, their experiences may vary greatly and not include the material from EET 2410.

Graduate students were the most interesting in that they seemed to have no or little experience with PLCs at all and were required by the nature of their course to have similar requirements to their undergraduate counterparts with some additional exercises to satisfy graduate school requirements.

A copy of the syllabus for all three groups is found in Appendix I. Not included are additions for the graduate course.

Course Development

The course is being taught in spring 2015 as a combined course (EET 4550, EECS 4220 and EECS 6980). The course includes a number of topics such as Siemens' function and function blocks, Human Machine Interface software familiarization, motion control and PID implementation. The text is a free web text developed by the instructor and found at:

cset.sp.utoledo.edu/~wevans (with login: wevans and password: Myeet12).

The course was developed from ideas formulated in a previous paper included in Appendix III.

Included in the course are review topics from the first course (EET 2410 – Programmable Controller Fundamentals) with students expected to review the topics quickly. The review is included in Appendix V and covers most topics found in the earlier course. The review was seen as necessary to give a common ground for the group. While the review was concentrated in the first two weeks of the course, time is given later to go back to selected topics to discuss areas not covered in the condensed review at the beginning.

The main topics for the course are the FB or FC block for Siemens, the HMI (Human Machine Interface), motion control and PID control. Each of these topics occupies approximately a month of time in the schedule of the course. Since the lab experiences are so closely tied to the course outline, the design of these labs is included next.

Lab Development

Labs included a lab to familiarize the student with the Siemens Function or Function Block. This lab was assigned during the first month of the course while those students not familiar with PLCs or specifically, the Siemens S7-1200, to become familiar with the Siemens hardware and software needed in succeeding labs. The need to teach the Siemens PLC is justified in Appendix III.

The second month included HMI labs featuring both the Siemens and Allen-Bradley software for HMI development. The lab assignments were to be completed with each brand and included a lab showing automation of the beer bottle down the conveyor and a cash register with buttons for a variety of menu items. Layout of the conveyor with the beer bottle is shown in the following figure 1.

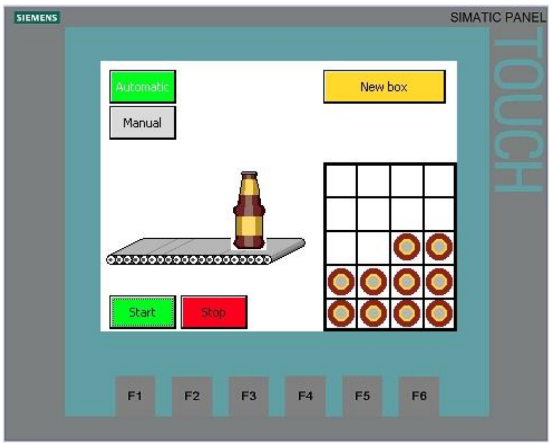


Fig. 1 Beer Bottle on Conveyor

Following the HMI lab experience, students are expected to become familiar with motion control equipment and be able to design a single axis servo and stepper using the template below for moving a single axis servo from Allen-Bradley and stepper from Siemens. The template serves as a model for this exercise.

While the students are participating in the motion labs, they must coordinate their lab time. There are four Allen-Bradley stations and four Siemens stations. Students are expected to fully program the application with each processor and the HMI for the PLC. They must switch at the mid-point from one system to the other in order to complete the entire requirement. Starter programs are given for each system and the systems are pre-wired for ease in implementing the project in the time allotted. The HMI for these two programs are to be modeled from the generic HMI shown below in Figure 2.

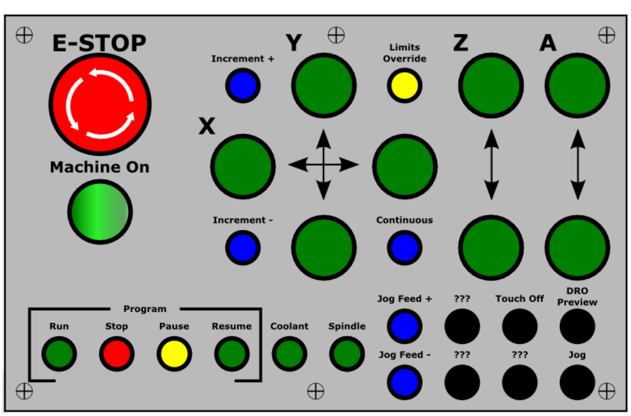


Fig. 2 Motion Control HMI Template

The two figures below show the A-B servo controller and the stepper controllers used with the Siemens processors.

Fig. 3 Servo Control for A-B

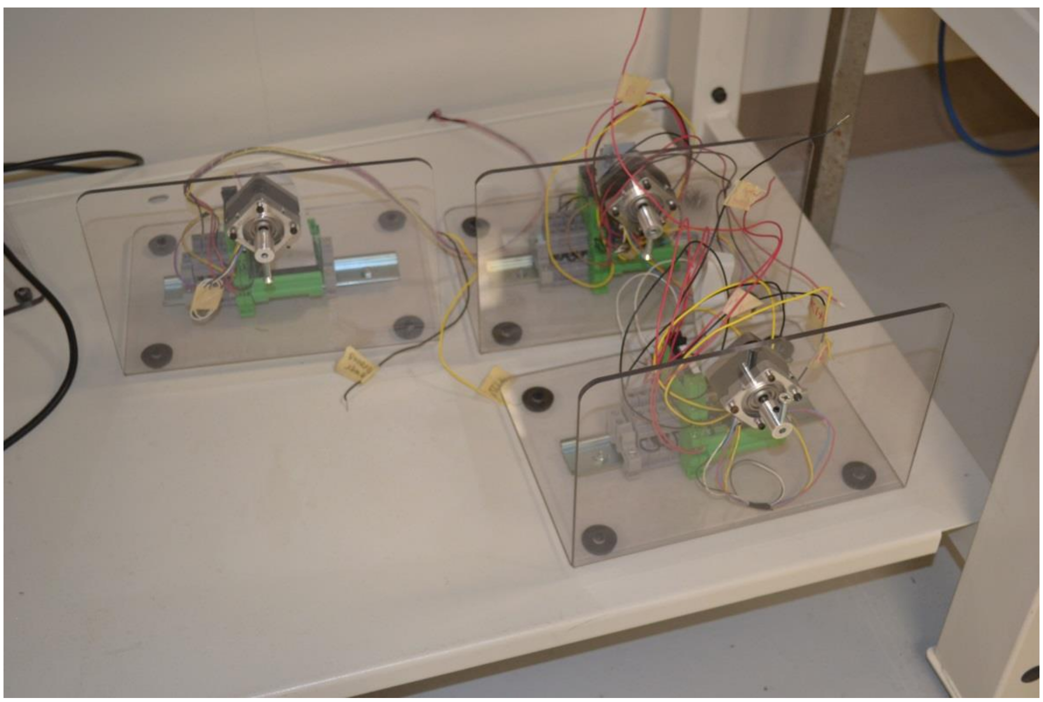


Fig. 4   Stepper Controller and Motor for Interface to Siemens S7-1200

The course also includes a segment concerning program planning. The various types of tasks and their implementation are discussed. There is a sample of this portion of the course in Appendix IV.

The final project requirement includes the PID algorithm and its implementation in some processes. The projects include a PLC coupled to a process. One of these is a water valve with flow transmitter feedback. The other is a ball levitated in a tube with fan control of air and laser height feedback. Both are to be programmed with the HMI package of the vendor and controlled in the various modes (auto as well as manual). While the purpose of the labs is not that of feedback control but rather the implementation of the PID controller using various control mechanisms, the activating of the loop as well as several tuning attempts gives a practical view on the subject of automatic control theory.

The three PID controllers shown below are not the only method the students may use to design their PID faceplates. Their purpose is to give an example of the practical steps needed in the design of a process using a HMI. The A-B and Siemens approach differ greatly in the treatment of the PID controller and its implementation. Having seen students program the two systems and achieve a successful implementation is a good indication that students can separate the underlying concepts from the application used by the various vendors' software.

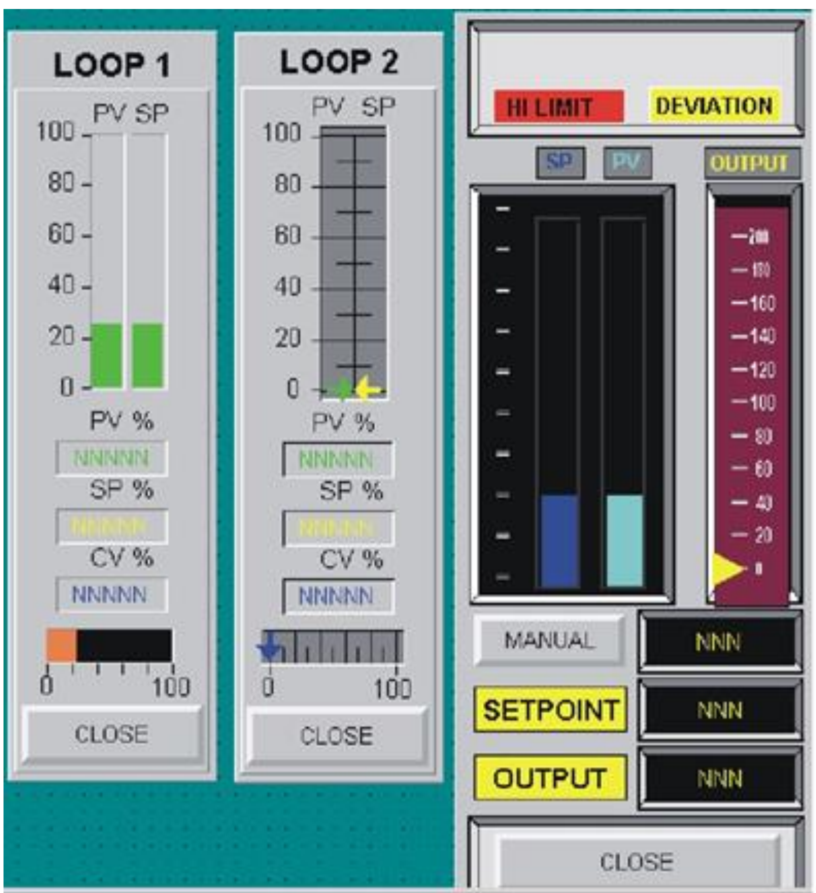


Fig. 5

Shown above in Fig. 5 are the HMI-sample PID faceplates used by Allen-Bradley. These are expected to be connected to the processes shown on the next page.

While the motion lab requires the students rotate between the two processors and equipment, the PID lab requires a similar rotation but with somewhat different requirements. There are four stations with the water valve. There are two water valves with a switch between the two stations at each valve. The two groups must share the development time and common equipment. But, when ready to try the program, they need to flip the switch to their trainer and begin the trial.

With the ball-in-tube PID experiment, there are eight trainers, enough for each group to try the PID process independently. Thus, there may be as many as 12 different groups involved simultaneously in the PID lab experience while only eight with the motion equipment.

Fig. 6 Water Valve Hardware

Fig. 7 Ball in Tube Hardware

Above are the two PID lab experiences. On the left is the water valve and flow transmitter with analog output. On the right is the ball in tube with the fan at the base and the laser position feedback at the top.

The final portion of the course is devoted to 'safety' PLC implementation. The use of safety PLCs in industry has burgeoned in the past few years and both A-B and Siemens have quality entries in this market. The discussion has no lab at present but the concepts of the various vendors' implementations of a safe PLC from both a hardware and software perspective are introduced. While the basic concept of programming has already been accomplished, the implementation of a safe PLC requires several additional steps. Functions using safe PLC hardware are introduced and the need to use these functions and share data with other PLC code is discussed.

Results

The result of combining this course is not yet known since the course is still actively being taught and in the early phase. If early results are an indicator, the course will be a success. Rarely do students enjoy this course since so many details are required for the various PLC programs. By its nature, the PLC project is highly stressful since the project is not seen to succeed until very late in the programming process. There is no guarantee that the results will be a success in a given period of time. With each lab/project experience, the instructor will help and eventually suggest possible programming segments to complete the project but only after the student has sufficiently tried to accomplish the task first.

Also, several program segments are given in each of the labs, especially with the motion labs. These labs require a great learning curve and the program given will give students confidence that the program will indeed allow the motor to turn.

Success is measured more by the end result of whether the project works or not, not from the process used to achieve that success. The use of multiple training stations is of high importance in that there is not just one person or group programming the application simultaneously but rather eight or more groups working toward the goal of a project that runs.

Results of tests are yet to be evaluated but will be included at the time of the presentation. There are a total of four tests in the course.

Are there many interactions between the groups in the course or do there seem to be three separate classes that co-exist? Early results tend to suggest that the three groups will remain three separate groups. This may change as the semester continues. It is hoped that some interaction between the groups will occur. Already, interaction between grad students and others is noted as lab assignments are due and the lack of any programming experience by this group is more evident. The lab is available via card-swipe day and night and on week-ends so there is ample opportunity for those with a general lack of experience to spend the time necessary to become acclimated to the PLC and its various programming techniques.

As the semester continues, the three groups have seemed to intermingle more. The separateness of the groups is not as noticeable.

The EECS group definitely is the most motivated. This is as it should be. They were the most motivated students through their academic career to this date and this trend continues in this course. Their respect for EET students may have been enhanced however. As a result of this course, they have competed, shared project time, shared programming tips for about 10 weeks and seem to be closer to their counterparts than before. This will serve both groups well in the future as they work with each other in the ‘real world’. The grad student population continues to be frustrated by the heavy work load demanded of them in the class. The projects are very demanding taking large portions of time. Since the result is not guaranteed until very near the end, pressure to succeed is very high.

The number of students is more than the course was designed to accommodate as well so everyone has learned to accommodate and share.

Over all, the experience has been good. Students are learning a skill of programming the PLC in a way that they never expected while sharing a space with those unlike themselves. This has been a growing experience for all including the instructor.

Summary

Whether EET students and EECS students can exist in the same classroom has been satisfied. They can be in the same room (at the same time in the same course with the same instructor). Can they both learn the concepts outlined in a course if the course is of the type that the experiential is emphasized and the theoretical is not? Of course they can – especially in one similar to the one outlined here! Does one want to label it a success and replicate it elsewhere? That is a question this author does not want to answer. However, with care and proper selection of material, there may be other courses that will have a genuine appeal to both student segments and be able to serve both groups simultaneously. Whether or not the course is taught in this way again is not as important as that when it was taught based on the needs of a group (the instructor went on sabbatical), the course was taught to both groups and was a success.

Bibliography

1 Programmable Logic Controllers: An Emphasis on Design and Application, Dogwood Valley Press, Kelvin T. Erickson
2 Programmable Logic Controllers: Fundamentals and Applications, Stipes Publishing, Wm T Evans
3 WinCC Basic V12.0 SP1 Systems Manual, Siemens
4 Kinetix 350 Start Up Instructions, Allen-Bradley
5 FactoryTalk Gateway, Allen-Bradley

AUTHOR INFORMATION

William Ted Evans, Professor, PhD, PE
U. of Toledo, Engineering Technology Dept.
University of Toledo, Toledo, Ohio 43606, wevans@utnet.utoledo.edu

BSEE, 1971, U. of Illinois, U-C
MSEE, 1975, U. of Toledo,
PhD, IE, 2005, U. of Toledo

Nicole Winhoven-Kamm, Instructor
U. of Toledo, Engineering Technology Dept.
University of Toledo, Toledo, Ohio 43606, nicole.kamm@utoledo.edu

BSEET, 2001, U. of Toledo

Appendix I

EET 4550	Programmable Controller Applications	Spr 2015
EECS 4220	Programmable Logic Controllers	Spr 2015

When and where	Lecture NE 2390 4:00 pm - 5:15 pm M,W	Lab NE 2390 3:00 pm - 3:50 pm M,W (and also NE 2350 as necessary)
Instructor	Prof. Wm Ted Evans, PhD, PE (Ohio)-Office: NE 1621, Phone 419-530-3349, cell 419-343-3681 Email: william.evans@utoledo.edu, web: www.cset.sp.utoledo.edu/~wevans Signin with: wevans password: Myeet12	
Office Hours	12:30-3:00 M,W	
Prerequisite	**Prerequisites:** EET 2410 and CSET 2200,	
Textbook	Online text at above website – Hybrid Text	
Useful References	Various vendor texts at their websites or at the above	
Grading	**Labs 30 %** **Test 1 15 %, Test 2 15 %, Test 3 20 %, Final Exam 20 %** **(A >= 90, B >= 80, C > = 70, D > = 60)**	
Class rules and regulations	1. No eating, drinking, or smoking in classrooms. 2. There are no make-up exams for this course. If you have a problem or conflict and cannot attend an exam, let me know beforehand and we will try to work something out. No credit will be given for a missed exam that we haven't made arrangements about beforehand unless you have a ***really excusable*** emergency. Cell phone use will not be allowed. If you do not have a calculator, buy one and bring it to class. ***Cheating is not allowed and will be punished by rules of U of Toledo Student Handbook.***	
Catalog descriptions	Use of programmable controllers and computers in factory automation. Topics include process control, supervisory software, PLC networking, PLC/CNC integration, device configuration, use of programming software and PLC language standards.	
Catalog descriptions	Use of programmable controllers and computers in factory automation. Topics include process control, supervisory software, PLC networking, PLC/CNC integration, device configuration, use of programming software and PLC language standards.	
Topics and reading assignments (subject to change, any changes will be notified in the class beforehand)	• Review of A-B, Siemens PLC programming • Addressing Review • Introduction to programming - RS-Logic Software • Introduction to HMI concepts • Introduction to PLC-CNC programming • Tuning of loops, PID algorithms • Process programming • PLC networking concepts • Discrete and analog I/O concepts	
Class dates (Exam dates are subject to change.)	1-12, 1-14, 1-21, 1-26, 1-28, 2-2, **2-4**, 2-9, 2-11, 2-16, 2-18, 2-23, 2-25, 3-2, **3-4** 3-16, 3-18, 3-23, 3-25, 3-30, 4-1, 4-6, **4-8**, 4-13, 4-15, 4-20, 4-22, 4-27, 4-29 **Final 2:45-4:45 Thursday May 7**	
	Homework assignments are listed on the website and are accepted only before or on the assigned day. Labs are to be printed from the website and brought to lab. Labs to be graded only if submitted at end of assigned class period.	

Appendix II

EET-4550-001 Test 1 Name____
100 pts total, 10 pts/ea

1. Name the three parameter types used to interface to a Siemens function:

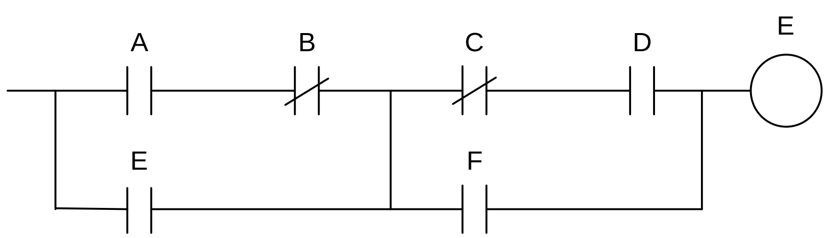
   a.____

   b. ____

   c. ____

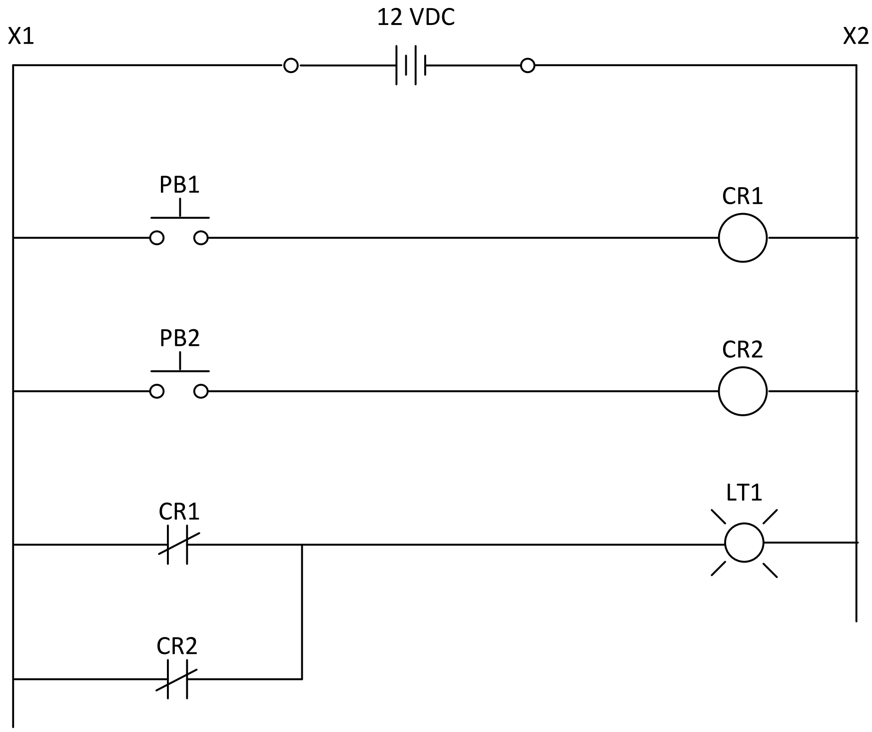
2. Data blocks are either single _______________ or multi _______________. What is the deciding factor which to use?

3. Convert the following seal circuit to a S/R circuit.



Proceedings of the 2015 ASEE North Central Section Conference

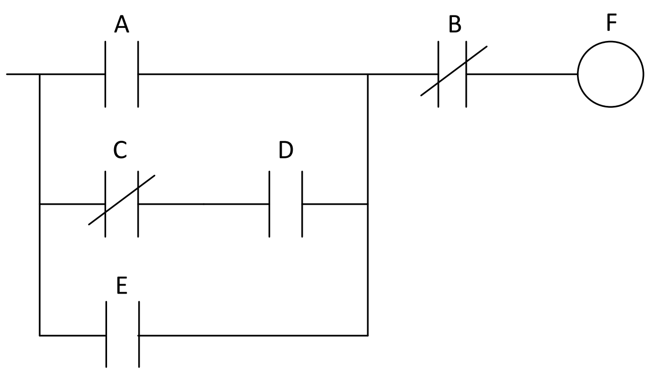
4. Observe what happens when the buttons are pushed:



Fill in the table with either 'ON' or 'OFF':

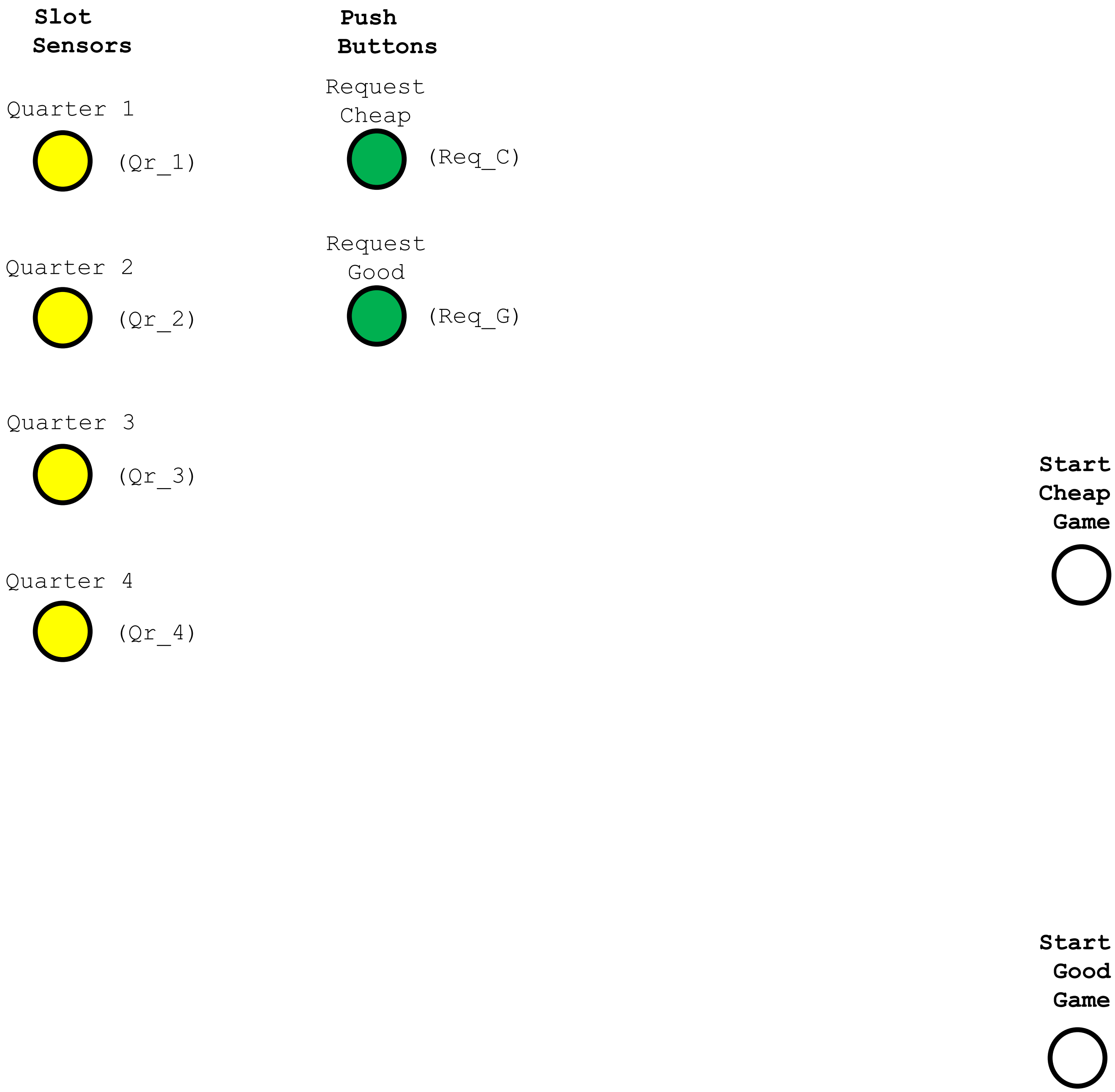
LT1	PB 1 Off	PB1 On
PB2 Off		
PB2 On		

5. Find the DeMorgan inverse of the following in Ladder Logic:

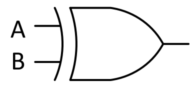


Proceedings of the 2015 ASEE North Central Section Conference

6. The attached buttons and coin slot sensors are part of an arcade game. Two games are in the same arcade box. One is a cheap game and one is a good game. If the player inserts quarters in any three of the four slots marked quarter 1 through quarter 4, and pushes the **Request Cheap** button, the cheap game starts. If the player puts quarters in all four of the quarter slots and pushes the **Request Good** button, the good game starts. Program rungs to energize a coil for starting the cheap game and a coil for starting the good game. The cheap game does not start if all four quarter slots are filled. Assume all state assignments for the slot sensors and buttons are equal to 1.



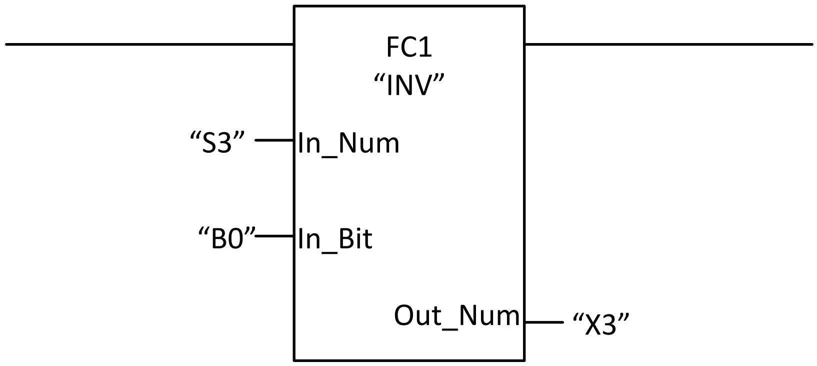
7. Convert the following Digital Logic gate to ladder logic. The truth table is included:

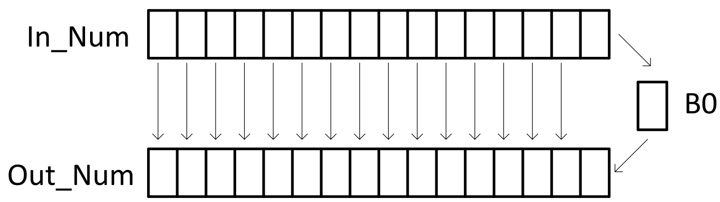


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

8. Write a rung to turn on a coil when the value in int_1 < 20 or float_1 > 8.

9-10 In OB1, there is a FC1 accessed that does the following:





Copy In_Num into Out_Num except for the bit 0 which is the exclusive OR of Bit 0 and the variable ‘B0’.

Build the function block “INV” to complete the operation. Show all tables and logic:

Appendix III

Using PLCs to Train Engineers and Engineering Technologists

By Wm Ted Evans, PhD, PE
Engineering Technology
U. of Toledo
Toledo, Ohio

Abstract

A course in Programmable Controller Applications needs to help the student find good employment. The course should also support subjects taught in the digital sequence as well as networking and programming. While the major emphasis has been to stress a particular manufacturer, specifically Allen-Bradley since AB has become the dominant force in the US, a second look should be made to better train the student using Siemens at the basic and advanced level. While change is not easily accepted, especially when large capital outlay is involved, the change should happen and happen soon.

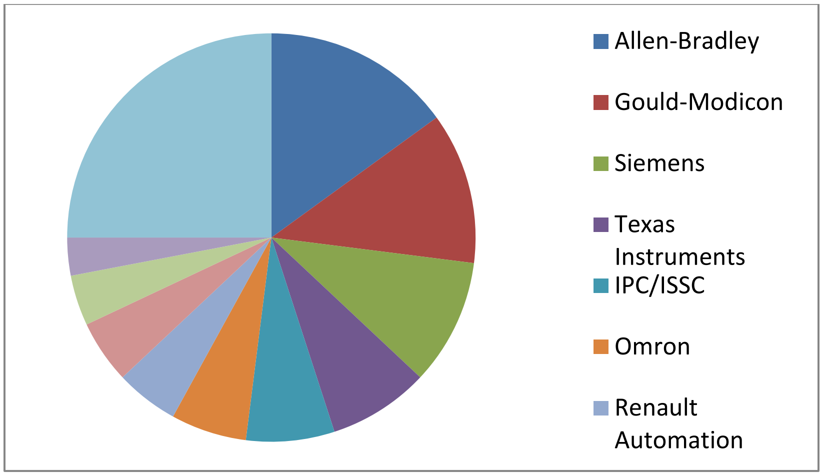
Introduction

Manufacturers of PLCs have been many and varied in the past with a stiffening competition over the last ten years. The effect has been a thinning of the ranks of PLC vendors. It costs much more to bring products to market than it did a few years ago. Foreign competition has caught up and in many ways surpassed domestic PLC manufacturers' technology. A number of buy-outs, consolidations and joint operating agreements have thinned the number of PLC manufacturers to a few. Allen-Bradley is the mainstay American company producing PLCs. Also in the US are General Electric and the combined Schneider Electric’s Modicon and Sq D organization. In Europe, the leading PLC manufacturer is Siemens and in the Far East, Japan's Omron and Mitsubishi. These companies are considered among the strongest automation systems companies in the world.

PLC Products

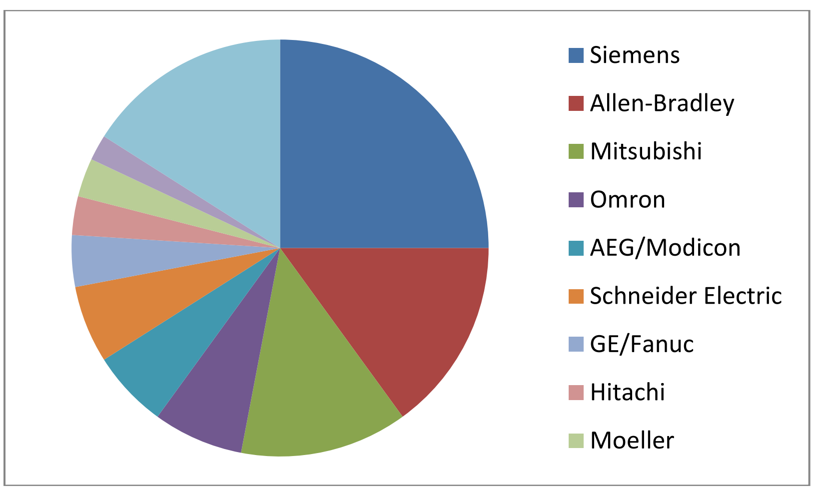
PLCs vary in size and type in a way similar to other manufactured products. Common to most manufacturers are the full size, compact, mini, micro and nano versions. Not surprisingly, the Japanese tend to dominate the mini, micro and nano end of the product while the Americans and Europeans tend to dominate the larger models. Siemens has the highest market share worldwide with its strong performance in Europe. Its purchase of TI's PLC division in the USA and its leadership in worldwide distribution and overall automation thrust has increased its lead.
The following four graphs show global PLC market share for various periods of time:

Global PLC Market share in 1982/83 [1]

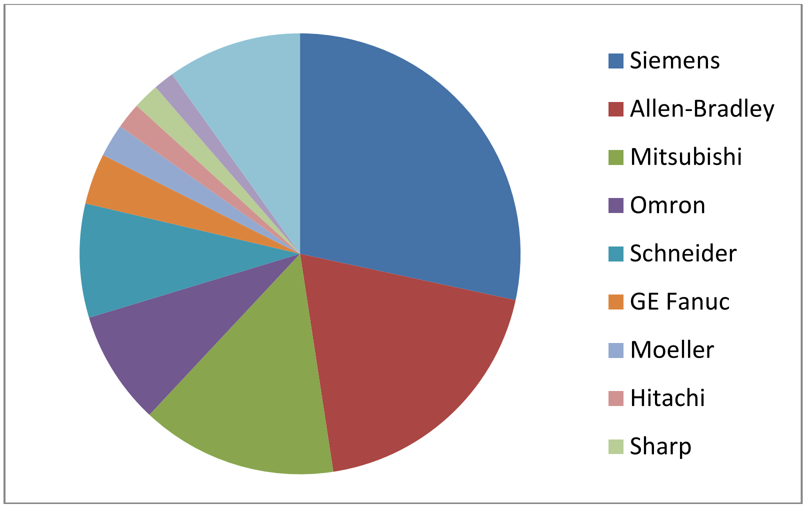


In the graph above from 1982/83, Allen-Bradley and Gould-Modicon hold the overall lead. This market was very diverse with market share not dominated by any one or two companies. "Other" was the largest overall group. In the next three graphs, Siemens is seen to overtake Allen-Bradley. Many strong players were left behind and eventually dropped off or were absorbed by other competitors. Today, Schneider with its Modicon product line may be the next to "drop off" the graph.

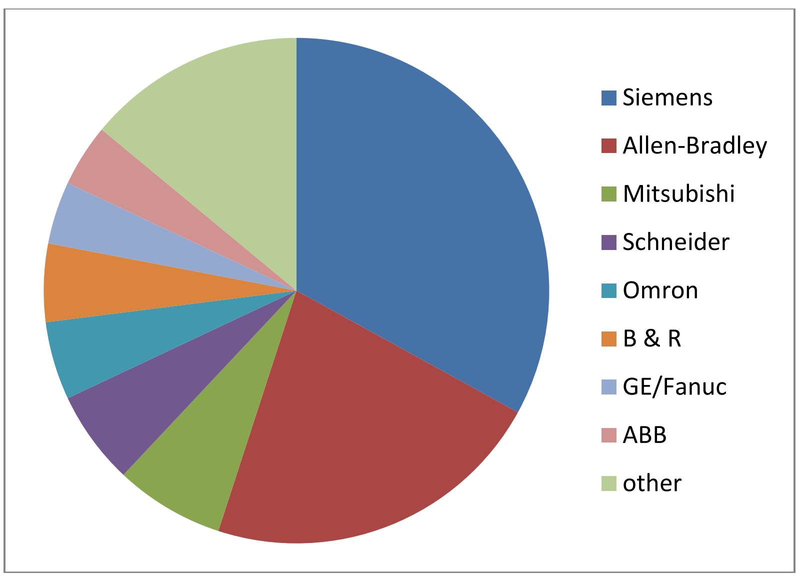
Global PLC Market share in 1992/93 [1]



Global PLC Market share in 2003 [2]

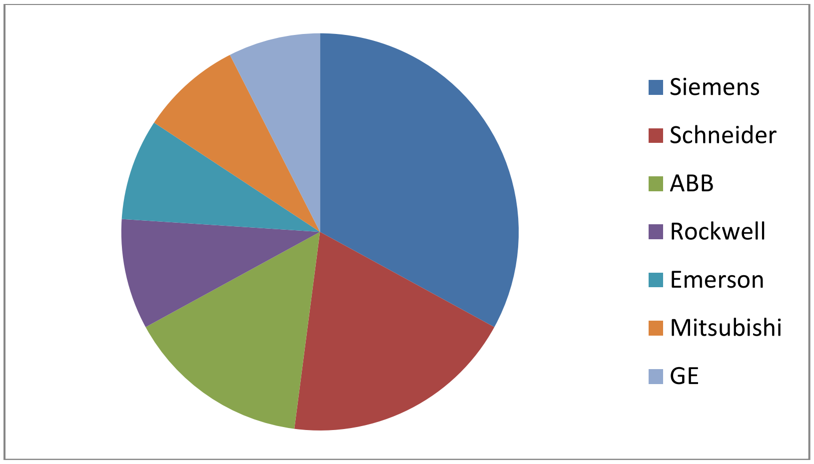


Global PLC Market share in 2007/2008



As can be seen, Siemens has made steady gains in each graph from its early status as a small player to its near-dominant status today.

The following graph is from the 2004 graph for total automation equipment dollars with Siemens leading with 8.8 billion euros for the 2004 year. [3]

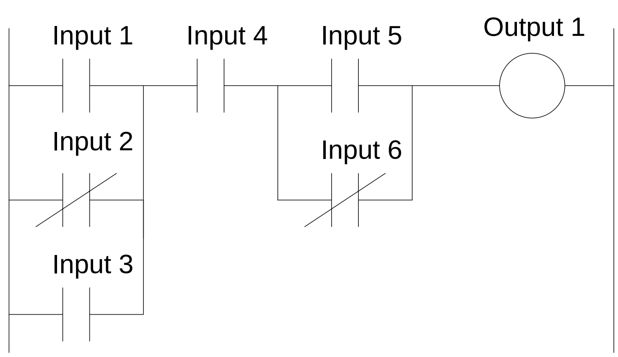


Programming Siemens PLCs

Programming is next reviewed with an eye to trends that have supported the surge in its popularity.

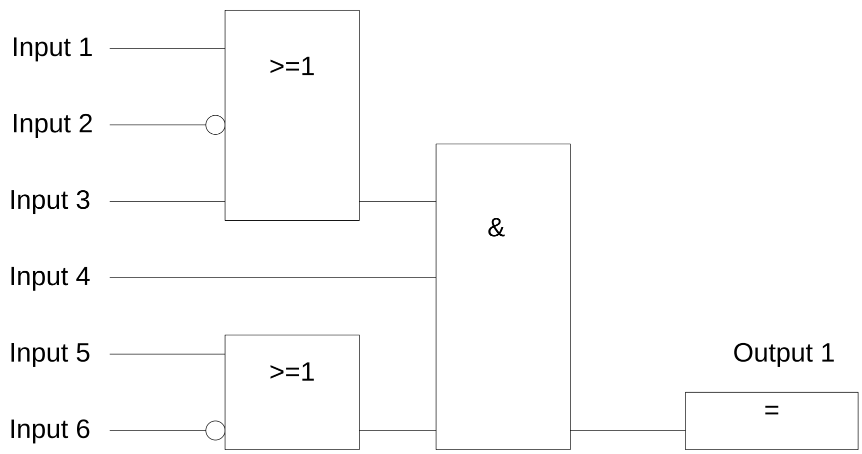
LAD

The ladder diagram (LAD) is based on the circuit diagram and is therefore especially advantageous for the representation of logic controls. LAD is most widely used in the discrete manufacturing industries including automotive.



FBD

The function block diagram (FBD) uses standardized and additionally vendor-specific function symbols such as AND and OR functions. FBD is preferred by some in the process industries.



STL

Statement list provides functions programmed with mnemonic abbreviations similar to assembler programming. STL is the most unrestrictive form of programming.

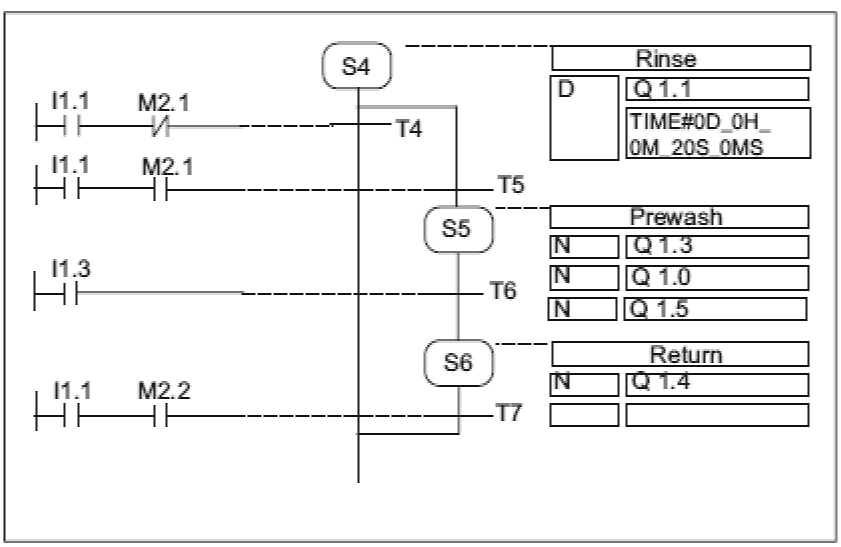
```
A(
O      Input 1     //OR function
ON     Input 2
P      Input 3
)
A      Input 4     //AND function
A(
O      Input 5
ON     Input 6
)
=      Output 1    //Output assign
```

Siemens had at first used STL programming exclusively and had been very successful with it. STL was a natural language for engineers who had been taught to program in an assembler-based language in college. It was also supported by the technology schools in Germany and Europe. While ladder diagrams have been preferred in the US, it may be good for the US educational effort to start with STL and then proceed to ladder diagram as a second language.

IEC 61131-3 was intended to achieve the long-term aim of creating user software largely vendor-independent and being able to port it to devices of difference to system integrators who want to use different target systems. While the IEC 61131-3 standard has been supported by all the major PLC vendors, hardware differences and overall product sophistication make the ultimate goal hard to reach that software is totally portable between vendors. The table below shows the Siemens language compatible with each of the IEC 61131-3 languages: [4]

	Simatic	IEC 61131-3	
Graphical representation	LAD Ladder Diagram	LD Ladder Diagram	Based on circuit diagram
	FBD Function Block Diagram	FBD Function Block Diagram	Based on switching circuit systems
	S7-Graph for sequencers	SFC Sequential Function Chart	For sequential control
	S7-HiGraph State-transition Diagrams		For asynchronous processes
	CFC Continuous Function Chart		In the form of technology oriented diagrams
Textual form	STL Statement List	IL Instruction List	Similar to Assembler
	S7-SCL Structured Control Language	ST Structured Text	Pascal-like High-level language

S7-HiGraph State language is shown in the figure below. It is useful for "asynchronous, non-sequential processes." The process is defined in terms of a number of different state diagrams that may run asynchronously to each other.



Blocks

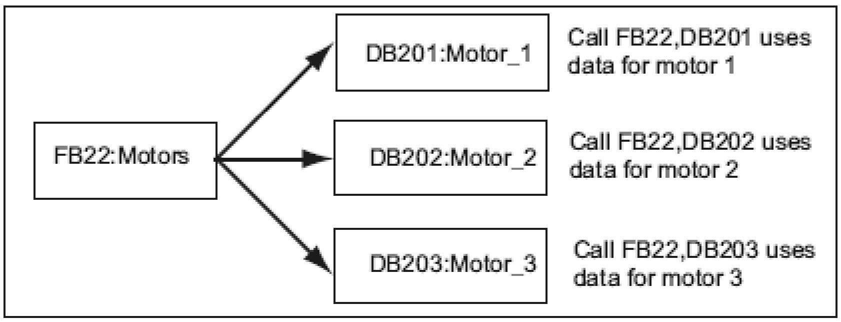
Blocks are the interface between the Siemens operating system for the S7 processors and the user program. OB1 is used to store the main program that is continuously scanned in the background. If other blocks are not being executed, OB1 is active. Interrupts and error handling programs may stop OB1 but should be constructed to quickly complete their task and give time for execution of OB1.

Organization block programming determines the execution sequence of the program and overall execution of the PLC.

Instance Data Blocks

An instance data block is built to store the data for a function block call. Each call requires a separate instance data block. Variables declared in the function block (FB) are saved in the instance data block (DB).

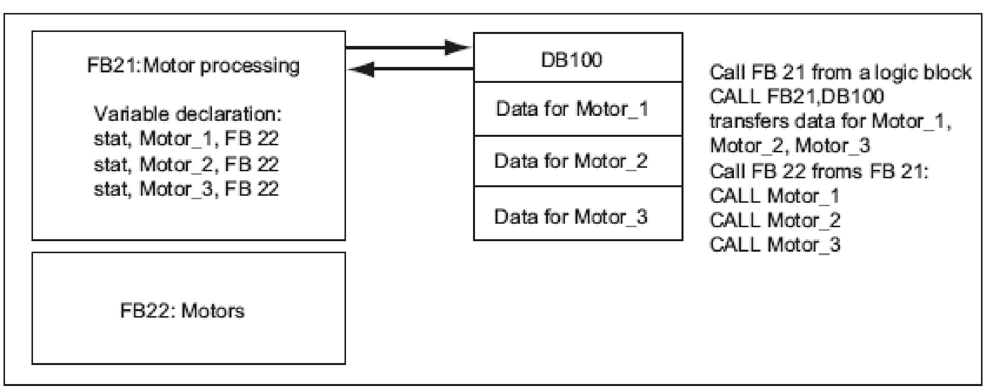
The figure from the manual *Programming with Step 7V5.4,* Ch. 4 shows a graphical representation of the FB/DB relationship. Each call statement of the FB requires a separate data block (DB) to execute.



This example shows a function block calling the function FB22 three times. First, data from DB201 is used. This data is comprised of data for "motor 1". The second call of FB22 uses data for "motor 2". The data is stored in DB202. Finally, a third call uses data from DB203 for "motor 3".

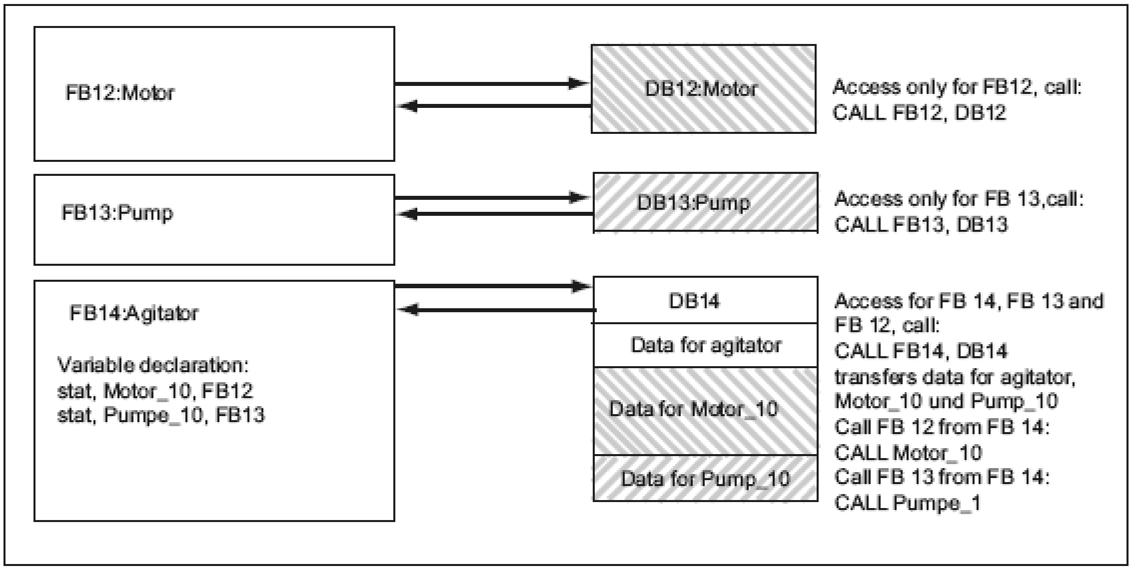
An alternate approach to one data block for each function call requires a programmed multiple instance data block. To accomplish this requires an additional FB be programmed. In the example below, the function block FB22 is created. In the data block DB100, static variables with the FB data type must be declared. The single data block has areas of data for each individual parameter set.

The figure below shows a graphical representation of this FB/DB relationship.



With the call of function block FB21, the function block FB22 is called which executes the same function as that shown in the earlier example. The data for each individual call statement is found in the area for that particular "motor".

Data blocks can include several instances of different function blocks as shown in the figure below. Again, the figure below from the manual *Programming with Step 7V5.4,* Ch. 4 shows a graphical representation of this FB/DB relationship.



Siemens approaches each area of programming in an organized and methodical way. The use of blocks has increased their acceptability as the number of standard blocks has grown and allowed the systems engineer to proceed with a project in a more organized manner.

Decision to Change PLC Training

The overall decision to use Allen-Bradley was always an easy decision to make. Many small and medium users have exercised their votes and have been A-B's most vocal supporters over the years. Their influence on the Industrial Advisory Boards at various universities and community colleges has had a lasting effect on the decision to use A-B.

The rise of Siemens may surprise many including this engineer and educator. What has fueled the change and will it continue until Siemens is the dominant PLC vendor?

First is my belief that the STL language has many inherent advantages not taught in most current courses involving the PLC in the United States. It is similar to an assembler language with some additional instructions to make it easier to use for the controls engineer. It is a very flexible language and must be considered when programming complex applications. The function block language or FBD is similar between A-B and Siemens and should not be considered when deciding to teach from either A-B or Siemens training material. FBDs popularity is somewhat limited and may never be the language of choice for controls engineers.

Organization of the data and function blocks in Siemens shows an advance beyond traditional ladder diagrams and gives the programmer additional organizational tools for advanced program development.

Next is a strong opinion that we must be open to adapt as educators to the changing climate of PLC languages and make changes when justified. The American control standards must be taught but world standards must be included in any course. Since many students will be working for multi-national companies with diverse requirements, the most dominant PLC must be understood and taught as well as the favorite of the US market.

Finally, it is my opinion that the present PLC market in the US is changing dramatically toward Siemens. Our students must be trained on the best of all types of PLC equipment in order to best compete. To not have a good grasp of the Siemens PLC in planning a PLC course may lead to further deterioration in the quality of job opportunities for our students when they graduate.

Summary

In the Milestones book from Siemens, the following quote was found:

"Siemens had at first used STL programming exclusively and been very successful with it. It seemed reasonable to program something in the way people think of it and describe it verbally. High education standards in Germany and Europe also supported this approach.

In the USA, where training for skilled workers was generally less intensive than in other countries, the ladder diagram, derived from the circuit diagram, dominated from the start." [5]

While this quote may be difficult for the American educator to assimilate and accept, especially for those who prefer the Allen-Bradley programming methods, it may be time for a re-evaluation and movement toward Siemens and the STL programming approach.

Bibliography

[1] Milestones in Automation from the Transistor to the Digital Factory, Arnold Zankl, Siemens, 2006, p. 89

[2] Milestones in Automation from the Transistor to the Digital Factory, Arnold Zankl, Siemens, 2006, p. 211

[3] Milestones in Automation from the Transistor to the Digital Factory, Arnold Zankl, Siemens, 2006, p. 208

[4] Milestones in Automation from the Transistor to the Digital Factory, Arnold Zankl, Siemens, 2006, p. 150

[5] Milestones in Automation from the Transistor to the Digital Factory, Arnold Zankl, Siemens, 2006, p. 54

[6] Programming with Step 7V5.4, Siemens Automation, Ch. 4

Appendix IV

PLC Program Organization Topics

If you want to execute a section of your logic	**Then use this type of task**	**Description**
All the time	Continuous Task	The continuous task runs in the background. Any CPU time not allocated to other operations (such as motion, communication, and periodic or event tasks) is used to execute the programs within the continuous task. • The continuous task runs all the time. When the continuous task completes a full scan, it restarts immediately • A project does not require a continuous task. If used, there can be only one continuous task
• At a constant period (example, every 100 ms) • Multiple times within the scan of your other logic	Periodic Task	A periodic task performs a function at a specific period. Whenever the time for the periodic task expires, the periodic task: • interrupts any lower priority tasks • executes one time • returns control to where the previous task left off You can configure the time period from 0.1 ms to 2000 s
Immediately when an event occurs	Event Task	An event task performs a function only when a specific event (trigger) occurs. Whenever the trigger for the event task occurs, the event task: • interrupts any lower priority tasks • executes one time • returns control to where the previous task left off The trigger can be a: • change of a digital input • new sample of analog data • certain motion operations • consumed tag • EVENT instruction

Appendix V

Review Topics Emphasized from First EET Course (EET 2410)

Ch. 1 Richard Morley
How PLC solves logic
PLCs in world
IEC 61131-3

Ch. 2 Boolean logic to Ladder translation
Parts of a motor starter and Ladder Logic implementation

Ch. 3 Allen-Bradley and Siemens software familiarization including connecting to a PLC

Ch. 4 Allen-Bradley and Siemens software familiarization including download and checkout of a simple given program.

Ch. 5 Signal Assignment in the I/O list
Siemens Addressing
Allen-Bradley Addressing
DeMorgan's Theorem reviewed in Ladder Logic

Ch. 6 Retentive memory and being able to convert from type to type
When to use seal/memory circuits
How one-shot works
Examples of one-shots being used

Ch. 7 Counters and Timers reviewed

Ch. 8 Numerical calculation and comparison statements reviewed

Ch. 9 Consideration of elements inside the panel

Ch. 10 Consideration of the elements of a control system outside the panel

Ch. 11 State Diagrams and their inclusion in the PLC

Ch. 12 Special Use Instructions and their implementation

Ch. 13 Batching/Table implementation and data use