A PLC Achievement Test

Wm Ted Evans
University of Toledo, wevans@utnet.utoledo.edu

Abstract - **Recent actions by EET leadership signaled development of a nationally normed assessment exam for graduating EET students. This decision was made by the Electrical and Computer Engineering Technology Department Heads Association (ECETDHA) in 2007 and includes graduates of 2 and 4 year programs in EET. Although the PLC was not included in the basic test, indications are that it may be included in an achievement test or considered for the basic test in future years.**

The paper develops a list of topics that would possibly be covered in the PLC achievement test or incorporated into the basic test. Rational for inclusion of various topics will be discussed. Topic areas for consideration for the test will be given.

Introduction

The author has been an instructor of PLC (Programmable Logic Controller) coursework for 24 years and has had continuing involvement in programming the PLC for industry during this time. Learning that a test or series of tests for Electrical Engineering Technologists was being planned by SME [1] created a great deal of interest with the obvious question of whether a PLC section would be included. It was learned at a later date that the PLC test was not included in the initial offering. While not unexpected, it was a disappointment since the PLC course has been for many years a center of interest in regard to employment as well as a required course due to the recommendation of the industrial advisory board. PLC education has varied widely and tends to follow the existing texts with some variation left for the individual instructor. If the instructor is from industry the instruction is probably more timely and given more to examples from industry. Also, the state of the PLC in the US has developed over the years to favor Ladder programming almost exclusively. Programming in other parts of the world tends to favor other languages, specifically the IEC (International Electrotechnical Commission) standards Function Block Diagram and Structured Text. This difference must not be ignored or minimized. Inclusion of other languages, especially Structured Text should be a fundamental part of PLC coursework and any achievement test.

Fundamentals

A PLC test must include the fundamentals of combinational logic as well as inclusion of timers and counters in logic. Included in the fundamentals is a thorough understanding of the normally open and closed contacts, their relationship to Boolean logic, and various methods of building combinational and sequential logic circuits. The one-shot contact is a necessity and its inclusion is a must for logic circuits. How the one-shot is implemented should be followed by questions concerned with the issue of when the one-shot should be used. Timers and counters should be included in a test when looking at configuration of the timer or counter for specific functions. Also included in a test should be applications of timers. For example, cascading of timers or counters as well as race circuits in which a counter is in a race condition with a timer. An example of a race would include, “did the operator push the button three times in two seconds?” Such circuits prove use of timers and counters together in an application. Simple memory circuits should be required. An example is the fill logic for a tank in which a high and low level switch is installed.

Addressing

Also important is a good understanding of addressing. Various vendors’ addressing should be required including Allen-Bradley (A-B) and Siemens. Not including a specific vendor may lead to arguments but these two are the dominant vendors in the US and the world.

Allen-Bradley Addressing Structure:
(Table 1a)

REAL	Number in floating-point	32 bit
INT	Number in integer mode	16 bit
String	Ascii Characters	Variable
BOOL	Bit	1 bit
COUNT	Counter	32 bit
BOOL	Digital Input/Output	1/16/32 bit
REAL	Floating-point Number	32 bit
DINT	Integer (whole number)	32 bit
CONTROL	Sequencer/PID/etc	Variable
TIMER	Timer	32 bit

Siemens Addressing Structure: (Table 1b)

IEC	Description	Data Type	Address Range
I	Input bit	BOOL	0.0 to 65535.7
IB	Input byte	BYTE, CHAR	0 to 65535
IW	Input word	WORD, INT, S5TIME,DATE	0 to 65534
ID	Input double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
Q	Output bit	BOOL	0.0 to 65535.7
QB	Output byte	BYTE, CHAR	0 to 65535
QW	Output word	WORD, INT, S5TIME, DATE	0 to 65534
QD	Output double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
M	Memory bit	BOOL	0.0 to 65535.7
MB	Memory byte	BYTE, CHAR	0 to 65535
MW	Memory word	WORD, INT, S5TIME, DATE	0 to 65534
MD	Memory double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
PIB	Peripheral input byte	BYTE, CHAR	0 to 65535
PQB	Peripheral output byte	BYTE, CHAR	0 to 65535
PIW	Peripheral input word	WORD, INT, S5TIME, DATE	0 to 65534
PQW	Peripheral output word	WORD, INT, S5TIME, DATE	0 to 65534
PID	Peripheral input double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
PQD	Peripheral output double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
T	Timer	TIMER	0 to 65535
C	Counter	COUNTER	0 to 65535
FB	Function Block	FB	0 to 65535
OB	Organization Block	OB	1 to 65535
DB	Data Block	DB, FB, SFB, UDT	1 to 65535
FC	Function	FC	0 to 65535
SFB	System function block	SFB	0 to 65535
SFC	System function	SFC	0 to 65535
VAT	Variable table		0 to 65535
UDT	User-defined data type	UDT	0 to 65535

Inclusion of Siemens may be difficult for some but I believe that their acceptance along with A-B should be a given. Siemens is the largest PLC vendor in the world[2], their machines are increasingly being used in the US and they have an advantage in the same types of markets that favor A-B, specifically automotive and the process industry.

An example of address usage should include not only direct address use but also indirect addressing or indexing. The Modicon instructions of Table->Register, Register->Table and Table->Table are examples of operations that the student must understand in order to program a sequential table problem. The test may use the indexing from Ladder or possibly require the creation of Structured Text code to implement the application.

State Diagrams

Development of a PLC program after a state diagram has been generated is of primary importance. The student should be able to take a simple state diagram and generate a program from the diagram to solve the task. The state diagram should include specific rules for proceeding from state to state.

PID Programming

The PLC has been used increasingly in the programming of process applications that were once dominated by process computers. Ladder or Function Block Diagram should be used. Control should be both vertical and horizontal. For example, a vertical example would require a program for one PID (proportional–integral–derivative) loop in either manual or auto. In auto, the choice is required between local and remote. An appropriate faceplate should be generated to complete the problem. Programming of a horizontal nature should include the programming of a number of PID loops connected together using a flow diagram or P&ID (Process and Instrumentation Diagram). Flow drawings or P&IDs can be found in a number of reference books from which the necessary PID blocks can be programmed.

Peer Communication

Since PLCs are rarely used in simple configurations but are connected to other PLCs and computers, it is important that the student know the general layout of some communications standards and programs using them. DeviceNet is most common in the US and ProfiBus is the most common in Europe. One or both of these should be required. An example may be the setting up of a list of objects in a scan list for DeviceNet and require the student to accurately address one or more of the devices.

The inclusion of Visual Basic in a computer and communicating via OPC should also be considered.

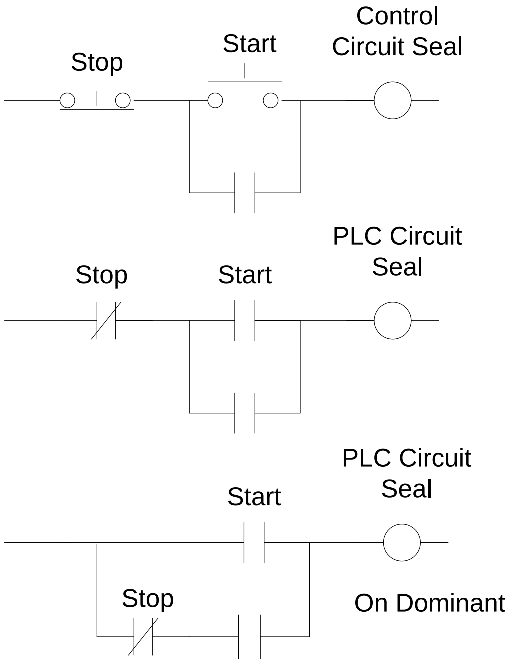
Setting up of networks including Ethernet should be required of the student. Either local area networks or wide area networks should be considered. Some use of security should be explored as the need for secure industrial networks is a high priority.

Use of and coordination of database information as it relates to the PLC should be explored as well.

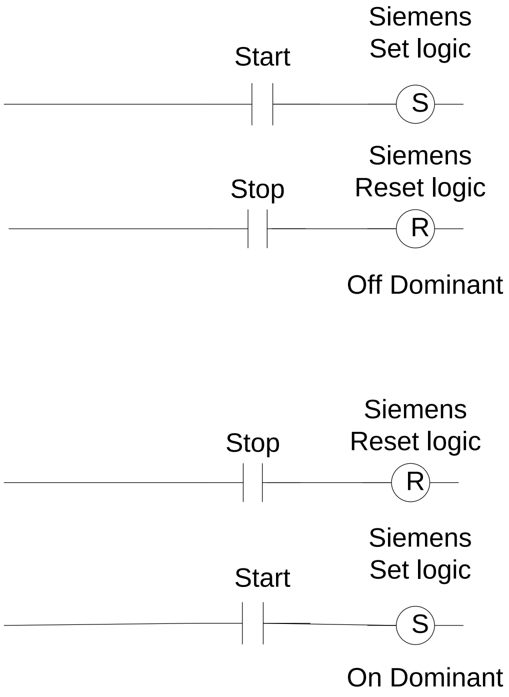
European Standards and Practice

It is believed that international standards should be used as much as possible when deciding on a set of standards. Common practice examples should use the US standard but examples of European common practice should be included.

(Fig. 1)



(Fig. 1, cont.)



The example of Figure 1 demonstrates differences between common practice of Allen-Bradley (US controls) and Siemens (European controls).

Whether dealing with standards or common practice, the US standard should not be the only standard tested. Both need to be understood. It may be advantageous to teach the international standard as well as the American and test to both.

Safety

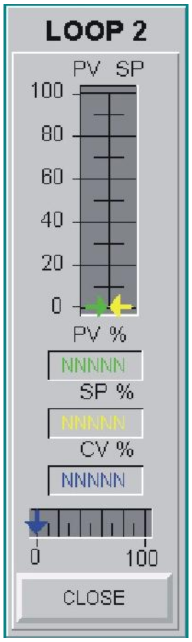
The basic design of safety circuitry has become a major design concern in industry. Design of a safety circuit as well as the various standards relevant to the safety design should be included in the PLC test.

Examples include study of various safety standards and building of safety circuits using PLCs. [3] [4]

Human Machine Interface

Design of the Human Machine Interface (HMI) should be included in the test. Required in the test should be the design of a button, numeric indicators, bar graphs and animation. Generation of historical data displays should be required. Alarm usage is also important and should also be required. An example question might include the design of a screen including a button that turns on a memory function, turns off the same memory function and reports the present status of the memory bit in the PLC. This type of program control is required in design of the PID block as well as a number of other types of applications. Overall layout of a screen for control of a process that includes PID should be included.

A faceplate common to most PID blocks is shown as an example of animation with the HMI (Fig. 2).



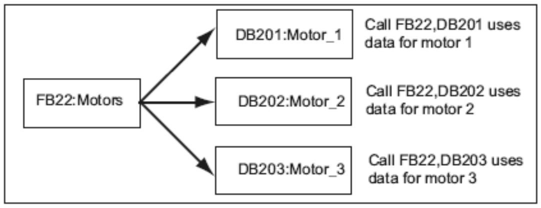
Faceplate (Fig. 2)

Also, layout of screens using various modes should be explored. An example would be the design of the operator interface for a lathe or mill.

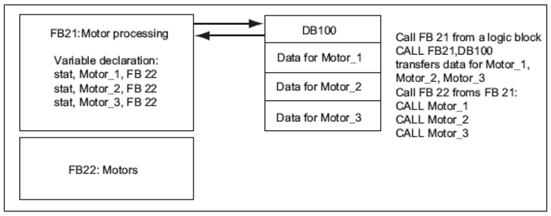
Program Organization

The design of software should include questions about the organization of that software. Included in modern PLCs is the idea of “classes” or encapsulation of an idea from the computer language C. The function block is the outgrowth of this effort. Both A-B and Siemens provides function blocks as building blocks for action as well as data storage. This organizational approach should be required in a PLC test.

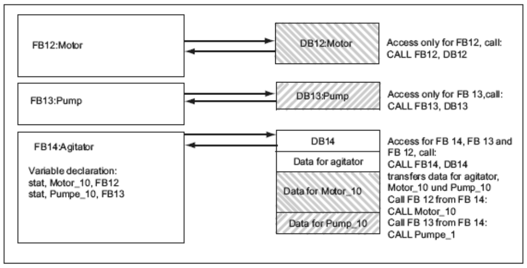
Examples of Siemens’ data block used in function block design are given below. See Fig. 3a, b, c)


FB22:Motors
DB201:Motor_1
DB202:Motor_2
DB203:Motor_3
Call FB22,DB201 uses data for motor 1
Call FB22,DB202 uses data for motor 2
Call FB22,DB203 uses data for motor 3

Example of Instance Data Blocks (Fig. 3a)



Example of Multiple Instance Data Blocks (Fig. 3b)



Example of Multiple Instance Data Blocks (Fig. 3c)

My Wish List

At some point, the list may grow to the point of exhaustion and the desire is to stop and say enough. Before a conclusion, there is a short list of other ideas that are wished for by this author.

They include understanding the fault recovery and event scheduling of the specific PLC as well as optimizing the program for time and efficiency.

The list also includes an understanding of the design of motion modules for various non-coordinated and coordinated axis machines (servo motor control).

Also included in the list is the desire to see the student tested on which language is best for a specific application. For example, a shift register may be useful for tracking of parts on a conveyor but the user may also want to add more sophistication which would imply the move to Structured Text. In other words, which language best fits the application. Always leading with Ladder is not advisable and the student needs to understand the power and efficiency of each language available and use the best one in each application.

Other Issues

The inclusion of so many different areas of study may be difficult for some. This is not a “one semester and test” solution being outlined. Multiple semesters are required to accomplish even a majority of these items. And concerning a text, most texts today do a great job of teaching the fundamentals but what about the more specific items mentioned above? The best text found to date in this author’s opinion is the free on-line text from Dr. Hugh Jack at Grand Valley State University in Grand Rapids, Michigan[5]. There are other acceptable texts but this one is very thorough and very free. If a test is built with good standards, the texts will follow that teach these standards. The Berger texts on programming are recommended for the Siemens PLC[6].

Conclusion

The design of a test for PLCs should be inclusive of a number of areas of study that may stretch the present PLC course and instructor. Equipment and texts may need to be enhanced to provide a valid teaching experience in preparation of the test created. The test should not compromise and only require a few of these areas but require elements from most or (hopefully) all of the topics described. Especially important is the importance of using a number of different languages or a best language to solve a particular problem. The result will be worth the effort.

REFERENCES

[1] Society of Manufacturing Engineers, www.sme.org – see achievement tests, eet

[2] Evans, Wm. T., “Using PLCs to Train Engineers and Engineering Technologists”, ASEE NCS Conf. Apr. 1, 2009

[3] Duta, Michael J., “ISA 2009 Maumee Event Risk Assessment”, mjduta@ra.rockwell.com

[4]Customer Advanced SIMATIC Safety Integrated Workshop - Developed by the Systems Engineering Support Center (SESC), Johnson City, TN

[5]Engineer on a Disk, Dr. Hugh Jack, P. E, Professor and Chair of Product Design and Manufacturing, Engineering at Grand Valley State University, Grand Rapids, Mi.

[6] Berger, Hans, “Automating with PROFINET”
Berger, Hans, “Automating with SIMATIC”
Berger, Hans, “Automating with STEP 7 in LAD and FBD”
Berger, Hans, “Automating with STEP 7 in STL and SCL”
Berger, Hans, “Controlling with SIATIC”

AUTHOR INFORMATION

William Ted Evans, Professor, PhD, PE
U. of Toledo, Engineering Technology Dept.

Univesity 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