Journal of Advances and Scholarly Researches in Allied Education

Vol. X, Issue No. XX, Oct-2015, ISSN 2230-7540

DESIGN AND CHARACTERISTICS OF EMBEDDED PROGRAMMABLE LOGIC CONTROLLERS

AN INTERNATIONALLY INDEXED PEER REVIEWED & REFEREED JOURNAL

Journal of Advances and Scholarly Researches in Allied Education
Vol. X, Issue No. XX, October-2015, ISSN 2230-7540

Design and Characteristics of Embedded Programmable Logic Controllers

Nihar Ranjan Panda[1] Prof. Dr. Rajendra Singh[2]

[1]KEISIE International University

[2]Rohella-Hitech Institute of Technology- Bhubaneswar

Abstract – Design and Characteristics of Embedded systems are playing an increasingly important role in control engineering. Despite their popularity, embedded systems are generally subject to resource constraints and it is therefore difficult to build complex control systems on embedded platforms. The impact of automation is visible in all areas of industry as well as in everyday life. Automation makes the process control more efficient, increases productivity of work, manufacturing quality, decreases manufacturing costs. Automation is still in development so that it could succeed in filling all requirements of today's technical advance. For this reason we daily meet new questions about implementation of automation systems, their handling and expanding. One of these is the question of communication in industrial applications. In case of having more PLCs in one industrial network, it is necessary to solve their inter-communication. We should deal with this question in dependence on some facts, for example: used control system, used industrial network, transmission reliability requirements and so on. In this article we would like to present a solution for intercommunication between PLCs in one industrial network communication. A programmable logic controller (PLC) is a small embedded computer used for automation of real-world processes, such as control of machinery on factory assembly lines. The program controls the complex sequences required in modern manufacturing automation.

Keywords: Design, Characteristics, Embedded, Programmable, Logic, Controllers, Network, Communication, PLCs, Manufacturing, Automation, etc.

---------------------------------X---------------------------------

INTRODUCTION

The Embedded PLC executes all of the IEC61131-3 programming languages plus Flow Charts. The developer writes these programs via the Isa GRAF V3.50 Workbench (32 I/O package included). The Embedded PLC Application Kit gives you everything you need to get started programming a Rabbit Semiconductor SBC with the Isa GRAF PLC programming system.

The Embedded PLC Application Kit is powered by the BL2500, an advanced single-board computer that incorporates flash memory, SRAM, digital I/O ports, A/D converter inputs, RS-232/RS-485 ports and Ethernet interface (10/100 Mbps). The BL2500 single-board computer gives PLC designers extremely low-cost embedded control for high-volume applications such as product control, factory equipment control, access control, HVAC, and vending machines. The BL2500's compact board size of 100 x 100 mm is easily mountable in standard 100 mm DIN rail trays. External connections via polarized locking industry standard Molex® type connectors enable rapid assembly with wire harnesses. Originally, programmable logic controllers (PLCs) were specialized computers for the mapping of hardware relay-logic to software in order to save costs and reduce necessary efforts for modification and maintenance of hardware-logic. After this concept had gained broad acceptance PLCs became soon one of the most commonly used types of automation elements in industry (Strasser, *et. al., June* 2007). The main reason behind this was the high acceptance by control engineers who could continue to use their relay-logic designs using a graphic programming language called Ladder Diagram (LD). Figure 1 illustrates the mapping of a hard-wired relay-logic (a) to its LD-counterpart which is based on the related connections of switches (A.B.C) and a coil (D) to digital inputs and an output, using their physical representations %IX1. %IX2. %IX3 and %QX1. Respectively (b). In (c) the LD-diagram equivalent to the logic (a) is given with standardized symbols of IEC 61131-3. (Sadeghi, 2010)

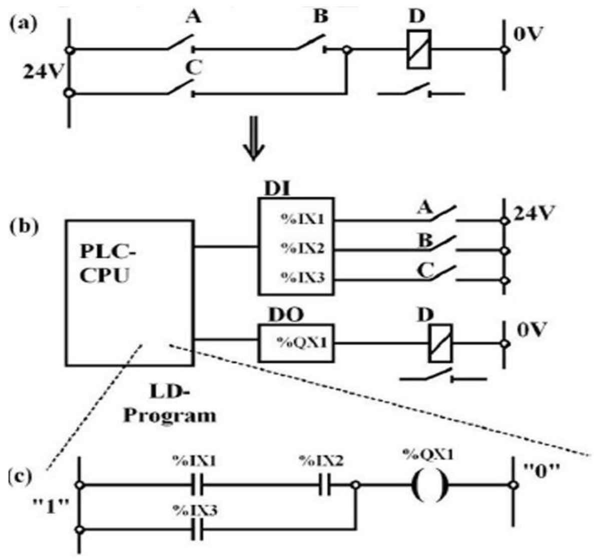

Figure 1- Ladder Diagram equivalent (c) to a given hard-wired relay-logic (a) using a PLC and its digital I/O (b)

The only drawback was that the inputs and outputs of the software-logic could no longer be simultaneously checked and adjusted, but only sequentially "*as fast as possible"', because of the finite computing time of the PLC (Sadeghi, 2010). The consistency of PLC digital input signals during a computation cycle had to be guaranteed by freezing their values in a process-map and by simultaneously updating the PLC output-signals, at the beginning and at the end of each new cycle, respectively.

REVIEW OF LITERATURE:

This basic permanent cyclic principle of digital signal processing (see Figure 2) has been used in PLCs up to now, and the increasing efficiency of microprocessors resulted for a long time mainly in the ability of computing more logic in the same cycle time. It is only since the advent of the Fust international standard IEC 1131-3 for programmable controller languages in 1993 that the introduction of modem real-time programming principles has been discussed with growing interest also with respect to PLC-software. Although a first step towards industrial application was made in this direction by adding parallel functionality to PLCs for feedback-control, the concurrent task execution of feedback and sequential control tasks, which is common practice with embedded micro-computers and is a forced trend with embedded PCs (Thramboulidis, 2012). did not however have any influence on PLC software design. Therefore. PLC efficiency improvements were achieved essentially only via the hardware. (Alois, *et. al.,* 2015)

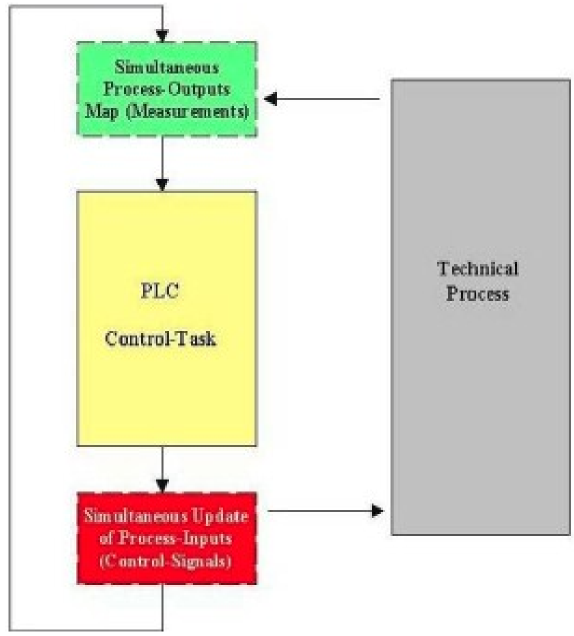

Figure 2- Permanent cyclic execution of a PLC controller task.

On the other hand, the range of problems solvable by PLCs increased significantly by extending the PLC software to include additional graphic languages, e.g.. Sequential Function Chart (SFC) and Function Block Diagram (FBD) for sequential and feedback control, respectively (Trejo, *et al.,* 2013). The graphic languages once again play a decisive role in the high acceptance of PLCs; however the textual languages ST and EL are also often used. The introduction of the IEC 1131-3 standard resulted once again in an expected "acceptance jump" by PLC engineers and programmers due the significant savings in training which were achieved by unifying the existing proprietary industrial language dialects of different PLC manufacturers.

1- HISTORICAL PERSPECTIVES:

The PLC has its origins in the motor manufacturing industries. Manufacturing processes were partially automated by the use of rigid control circuits, electrical, hydraulic and pneumatic. It was found that whenever a change had to be made, the system had to be rewired or reconfigured. The use of wiring boards on which connections could be changed by unplugging them and changing them around followed. With the development of micro-computers it was realised that if the computer could switch things on or off and respond to a pattern of inputs, then the changes could be made by simply reprogramming the computer and so the PLC was born. There are still many applications of automated systems with permanent connections to perform a single control action (ONDROVIČOVÁ, 2010). Often the system uses logic components to produce the correct action (electronic and pneumatic). The PLC" mimics this process by performing the logical operations with the programme rather than with real components, hi this way cost savings are produced as fewer components are needed and more flexibility is introduced as programmes can be changed more easily than reconfiguring a hard ware system.

2- PLC ARCHITECTURE:

The PLC activates its output terminals in order to switch tiling's on or off. The decision to activate an output is based on the status of the system's feed-back sensors and these are connected to the input terminals of the PLC. The decisions are based on logic programmes stored in the RAM and or ROM memory. They have a central processing unit (CPU), data bus and address bus. A typical unitary PLC is shown below.

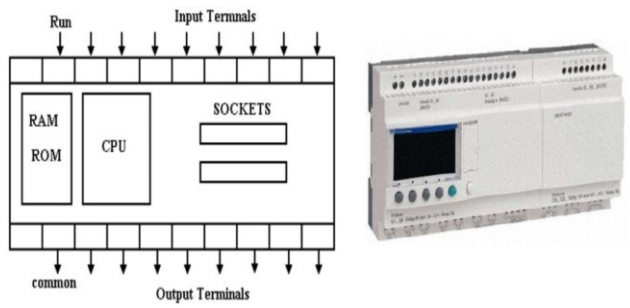

Figure 3: A typical unitary PLC

The next diagram shows a very oversimplified diagram of the structure. The Central processing Unit controls everything according to a programme stored in the memory (RAM or ROM). Everything is interconnected by two busses, the address bus and the data bus (shown as a single red line). The system must be able to communicate with external devices such as programmers, display monitors and Analogue Digital converters.

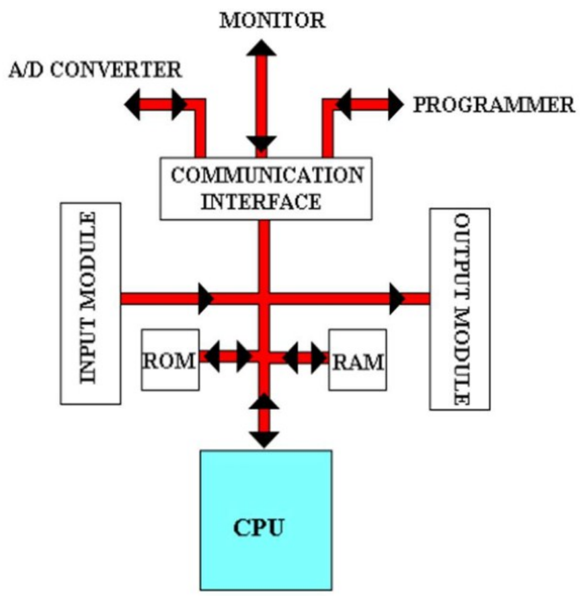

Figure 4 Structural Processing of PLC systems

3- THE PLC IN AUTOMATION ENGINEERING:

The first Programmable Logic Controller (PLC) was developed by a group of engineers at General Motors in 1968, when the companies were looking for an alternative to replace complex relay control systems.

The new control system had to meet the following requirements:

- Simple programming
- Program changes without system intervention (no internal rewiring)
- Smaller, cheaper and more reliable than corresponding relay control systems
- Simple, low cost maintenance

4- INCREASE FLOW MEASUREMENT USING EMBEDDED PLC:

This study explores the use and benefits of embedding hydrocarbon flow computer technology into programmable logic controllers (PLCs) for integrated measurement, control, and communications. Accurate measurement of hydrocarbon gases and liquids using industry measurement standards is vital for process control systems, leak detection, and the transfer of product between two parties, more commonly known as custody transfer. As the complexity and scale of energy systems increase, automation technologies become more important for production efficiency and controlling costs. Process control engineers now have new innovative automation technologies for flow measurement, control and data management (Vyatkin, *et. al.,* 2005). As technology advanced, flow measurement methods expanded to include more accurate and reliable techniques such as magnetic field, positive displacement, and ultrasound. However even with this improved technology, true flow calculation needed to take into account other measurements and conditions. To calculate true flow, temperature, pressure, pipe diameter and material purity all need to be taken into account. Because these attributes are dynamic (changing frequently overtime), a way to continuously measure these variables and then calculate flow was required (Function blocks, 2005). The flow computer was invented precisely for this purpose.

CONCLUSION:

The study suggests one potential technique and particular tools for rapid prototyping of embedded systems and design space exploration. The tools include the developed core (kernel) prototyping board and two types of extensions for standard peripheral devices and special-purpose systems. This enables the users to assemble complex systems from the supplied blocks supported by the proposed libraries and IP cores, which significantly simplifies the design process and shortens the design lead time. The technique gives an easy opportunity to extend and to customize the functionality of available hardware allowing constructing circuits that are optimized for particular applications. This enables the users to combine advantages of block-based and application-specific design. Design And Characteristics of Embedded Design has advantages to deal with dynamics. However, its performance is hard to predict and can be inferior to a centralized decision methodology under stable conditions. The agent based design adds configurability to control logic in control systems. A stepwise decision process with details of agent types and their decision logic has been designed for agent-based decision process. Some widely neglected problems in the agent-based design such as communication burden, local interest conflicts have also been studied in this research.

REFERENCES:

T. Strasser, C. Sunder, A. Zoitl, M. Rooker, and J. Brunnenkreef, (June 2007). “Enhanced IEC 61499 Device Management Execution and Usage for Downtimeless Reconfiguration,” in Industrial Informatics, 2007 5th IEEE International Conference on, vol. 2, pp. 1163–1168.

M. Sadeghi (2010). —Developing IEC61499 in Industrial Processes, Measurement and Control Systems (IPMCS), vol 5 issue 4,pp 238-241, wseas transaction on systems and control.

K.Thramboulidis, (2012). —IEC 61499 vs. 61131: A Comparison Based on Misperceptions, IEEE.

M. Sadeghi (2010). —Developing IEC61499 in Industrial Processes, Measurement and Control Systems (IPMCS), vol 5 issue 4, pp 238-241, wseas transaction on systems and control.

Alois ZOITL1 Gunnar GRABMAIR2,Franz AUINGER3 and Christoph (2015). SUNDER4 ,—Executing real-time constrained Control Applications modeled in IEC 61499 with respect to Dynamic Reconfiguration pp 62-67,IEEE.

M. Trejo-Herbabdez,E. Lopez-Mellado (2013). —Specification of Manufacuring Systems Controllers Using the Standard IEC 61499,pp-179-184, IEEE.

FRANEKOVÁ, M.; KÁLLAY, F.; PENIAK, P.; VESTENICKÝ, P. (2007). Komunikačná bezpečnosť priemyselných sietí. Žilina. ISBN 9078-80-8070-715-6.

ONDROVIČOVÁ, M. (2010). Control system SIMATIC S7-300 [online]. 20. 1. Available at WWW: <http://www.kirp.chtf.stuba.sk/~ondrovic/simatic.htm>.

V. Vyatkin, J. Christensen, and J.L. Martinez Lastra, (2005). "OOONEIDA: An Open, Object-Oriented Knowledge Economy for Intelligent Distributed Automation", IEEE Transactions on Industrial Informatics, 1, (1), p. 4-17.

Function blocks (2005).: International Standard IEC61499, International Electro technical Commission.