Research Journal of Applied Sciences, Engineering and Technology 2(3): 283-288, 2010
ISSN: 2040-7467

Submitted Date: March 31, 2010 Accepted Date: April 13, 2010 Published Date: May 10, 2010

Design of a Virtual PLC Using Lab View

M.K. Abuzalata, M.A.K. Alia, Shebel Asad and Mazouz Salahat
Department of Mechatronics Engineering, Faculty of Engineering Technology,
Al-Balqa' Applied University, Amman, P.O. Box 15008, 11134, Jordan

Abstract: Using LabVIEW environment a trial has been made to create a set of programmable virtual instruments, which resemble the traditional PLC programmable functions and networks. The target of this work is to improve the programmability of PC-based control systems. Bringing the PLC to the industrial type computer it becomes possible to make use of the advantages of computer based DCSs. It will be possible to have unlimited number of programmable objects, and to run more than one program at the same time. By utilizing LabVIEW front panel it is an easy job to realize MMI as required, and by creating sub VIs the program becomes more compact and easier to debug.

Key word: Block diagram, front panel, LabVIEW, PLC, sub icon, virtual instrument

INTRODUCTION

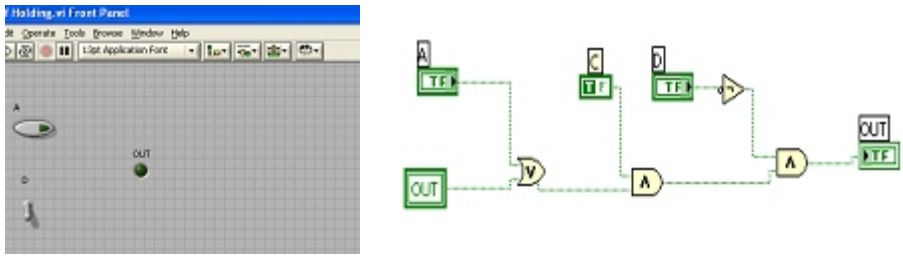
Today's manufacturing plants must operate faster, more efficiently, and with more flexibility than ever before. To do so, a growing number of manufacturers are implementing open architecture approaches such as PC-based DCS and SCADA control systems, there by bringing the speed and flexibility of PC technology to the plant floor. Today traditional PLCs are still in use at most plants, but windows-based PCs are increasingly becoming the preferred control mechanism for new installations. PLCs have become a favorite tool in the control industry because of their simplicity, robust I/O interface and reliable performance Webb and Reis (1999). Traditional PLC systems have proven to be information barriers to enterprise-wide data access. Originally the PLCs had no communications capability, but they began to be used in situations where communications was a desirable feature. At the time-being manufacturers of PLCs have devised many communications techniques and pseudo-standard protocols, which are utilized in industry. One may add that the inherent proprietary design of PLCs has limited data access for a number of reasons, such as the limited amount of memory, the nature of programming language (Relay Ladder Logic), and the data access Bryan and Bryan (1997), where data inside the PLC is stored in a data table and accessed by data table location. An important feature of PLCs, is that a standard PLC executes only a single program at a time, while an industrial computer is capable of executing several programs or tasks simultaneously in any order. Another two important PLC drawbacks may be noted also: The first one is that PLC register access is performed at a surprisingly low level on most PLCs Johnson (1994). The second one is that if the ladder logic and the host computer program both write to a PLC register, we have an obvious conflict. Instead, all registers should be one-way, that is, either the PLC writes to them or the host computer program does.

In contrast to PLCs, PCs have virtually unlimited memory, compared to traditional PLCs. In PC-based control systems, programming, MMI and data communication, access the same data through the same tag name. The end result is faster design cycles and less human error. In contrast with the PLCs, which normally implement an ON-OFF or PID modes of control the PC-based control systems provide very sophisticated analog control capabilities.

Concerning the programming languages, graphical flow chart programming lend themselves much better to the logical, sequential nature of communications interfaces required inside the control engine Parr (2001). PC-based control systems have pioneered the use of higher- level programming. Some suppliers of RTUs have created simple graphical user interface in order to configure the RTU easily. Using very powerful language LabVIEW it is possible to speed up programming considerably as it is designed to take measurements, analyze data and present data to the user. LabVIEW makes it easy to maintain good architecture in the applications because encapsulation and modularity are easy to implement through the use of sub. VIs.

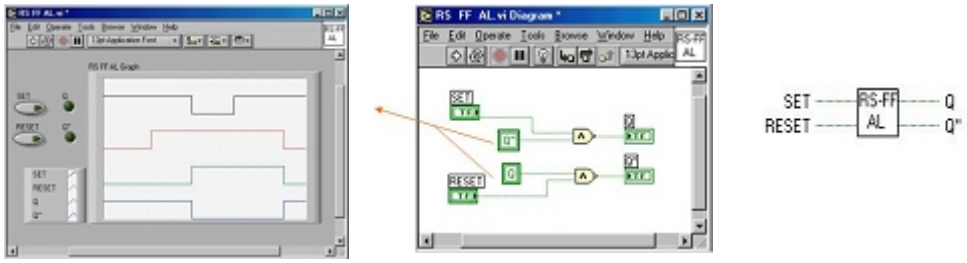
Remote Terminal Units (RTUs), Programmable Automatic Controllers (PACs) Baily and Mipenz (2003), differ from a PLC in that they are more suitable for wide geographical telemetry, often using wireless communications, while PLCs are more suitable for local area control. Modern RTUs are capable of executing

Corresponding Author: M.K. Abuzalata, Department of Mechatronics Engineering, Faculty of Engineering Technology, Al-Balqa'Applied University, Amman, P.O. Box 15008, 11134, Jordan



(a) The front panel (b) The block diagram

Fig. 1: The holding circuit



(a)The front panel (b) The block diagram (c) Sub VI

Fig. 2: R-S Flip-Flop active low

simple programs autonomously without involving the host computers. RTUs, PLCs and PACs are increasingly beginning to overlap in responsibilities and many vendors sell RTUs with PLC-like features Baily and Mipenz (2003), and vice versa. The Compact RIO is an example Saab (2008). RTUs have always been used in situations, where the communications are more difficult. But RTUs have poor programmability in comparison to PLCs.

Building on the above the target of this work is to help in that direction. i.e. to create a set of instructions (Sub VIs) which are LabVIEW equivalents for standard and common ladder logic networks, subroutines or functions. In PLCs, there is a library of programmable objects, special functions (subroutines), memory registers, memory bits (relays) and other elements, which are easily accessible and programmable. What we are planning is to create a similar library using Lab VIEW environment, bringing by that the PLC to the computer. This gives more flexibility and programmability to the industrial type computers or PACs.

MATERIALS AND METHODS

This study is conducted at Department of Mechatronics Engineering, Faculty of Engineering Technology, Al-Balqa'plied University Amman, Jordan, 2010.

LabVIEW Representation of the Basic Logic Elements:

The holding circuit:

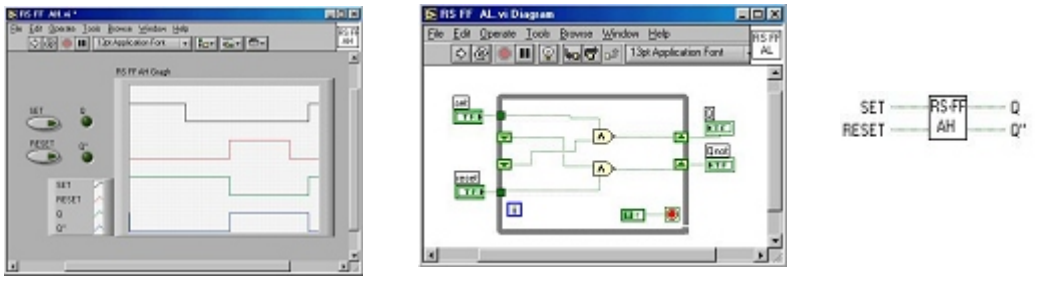
One of the most common PLC networks is the self-holding circuit. Fig. 1 shows a suggested front panel, block diagram VI and Sub VI of the LabVIEW equivalent VIs.

Local variable R-S Flip-Flop: The Set-Reset instruction may be looked at as a development in the self-holding instruction. Wherever the enable input of a set-sequence is – ON; the point address of the set-instruction will turn ON and stay ON until an enabled reset sequence (RST) with the same point address is reached. The set sequence is self-holding (point address remains ON even if the enable input turns OFF). Lab VIEW equivalent circuit to the ladder logic instruction SET-RESET may be realized by utilizing on RS latch flip-flop. When referring to the output conditions of the flip-flop circuit, the definitions of SET and RESET are the following:

SET:	Q1 output is logical - 1
	Q' output is logical - 0
RESET:	Q1 output is logical - 0
	Q' output is logical - 1

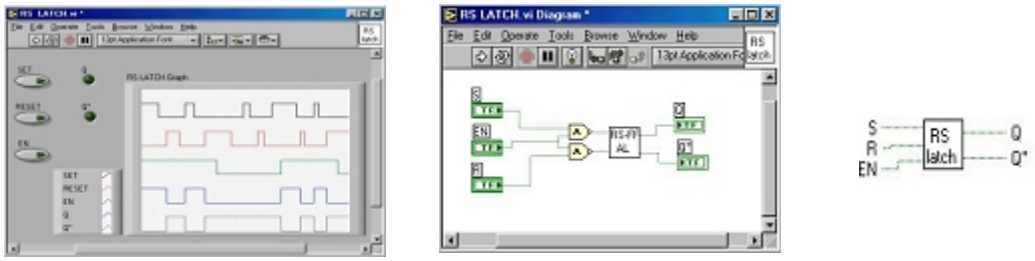
A basic RS latch flip-flop can be formed using a pair of logic gates-either NAND or NOR.In LabVIEW we cannot connect the output of the logic gate directly to the input of the other predecessor source gate. In order to do that local variables are used as suggested in Fig. 2,

Which resembles an active low (NAND type) LabVIEW equivalent VI and Sub VI for the SET-RESEST instruction. Considering the fact, that LabVIEW provides memory in the form of the shift register, another



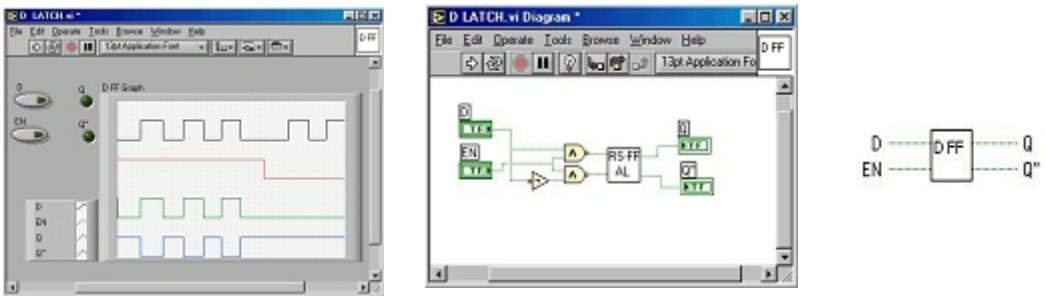
(a)The front panel (b)The block diagram (c)Sub VI

Fig. 3: RS Flip-Flop active high



(a) The front panel (b) The block diagram (c) Sub VI

Fig. 4: Clocked RS Flip-Flop



(a) The front panel (b) The block diagram (c) Sub VI

Fig. 5: D Flip-Flop

LabVIEW equivalent VI to the SET-RESET is given in Fig. 3. An RS-flip flop active high VI and sub VI are illustrated in Fig. 3.

Clocked S-R Flip-Flop: While the RS latch flip-flop is directly operated by the R and S input logic signals, the majority of flip-flops are clocked or triggered. An advantage of clocked RS flip-flop is that its output remains in its present state even if the R and S inputs don't remain the same, changing only when the clock pulse occurs. Figure 4 shows the front panel, block diagram VI and sub VI of this flip-flop.

D-Flip-Flop: One popular version of flip flops is a D-type, where the D input is the data or logical signal input, which determines the output after a clock pulse occurs. When the D input is (0), the Q output is (0) after the clock pulse, while (D) input of (1) results in a (Q) output of (1) after the clock pulse.The front panel timing diagram, the block diagram VI and SubVI of the D-type flip-flop are shown in Fig. 5.

Using a while loop and shift register another LabVIEW block diagram VI is shown in Fig. 6. The input terminal of the shift register represents the (D) input, the left terminal represents the (Q) output and the loop iteration index [i] represents the clock input. An interesting characteristic of this VI is that it realizes an ON-delay and an OFF-delay of the output (Q) relative to the input (D). When the period of the (D) input is longer than the period of iteration, the time delays will be a multiple of [i] period, and when the period of [i] is higher than that of the (D) input, the time delays will be a part of

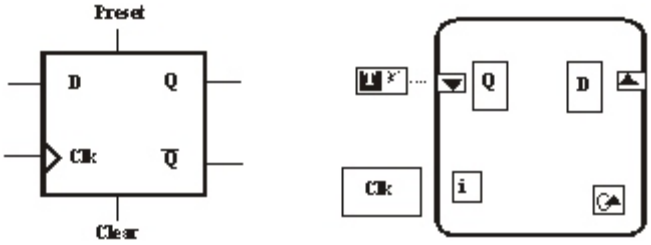
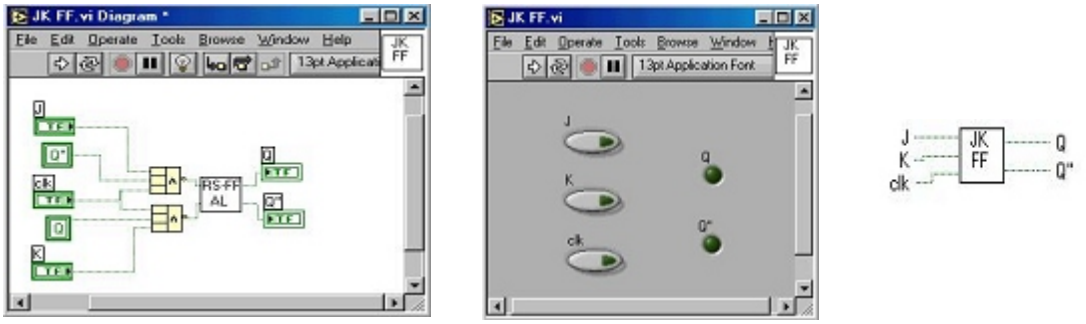
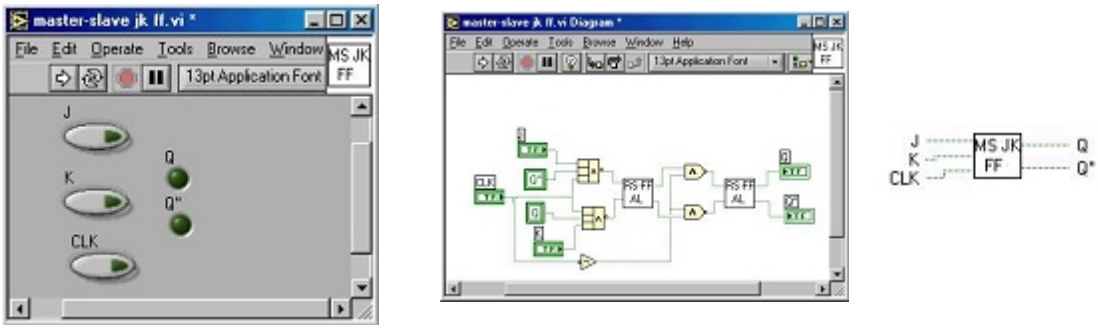


Fig. 6: Shift register as virtual unit



(a)The front panel (b)The block diagram (c)Sub VI

Fig. 7: JK Flip -Flop (Single Pulse)



(a)The Front panel (b)The block diagram (c)Sub Icon

Fig. 8: Master Slave JK Flip -Flop(single pulse)

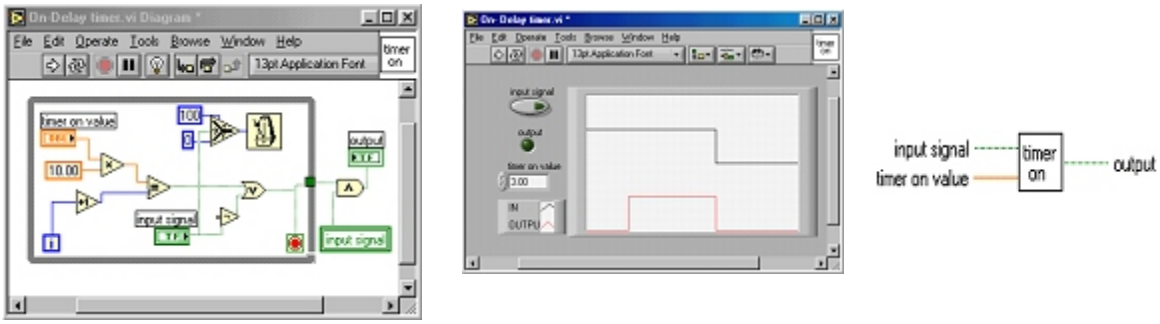
the period of [i]. So, as with all clocked flip-flops, it is important to know how or when during the clock cycle (iteration cycle) the output may change state.

J-K Flip-Flop: Concerning (JK) flip-flops, they are the most useful and versatile flip-flops available. The most important characteristic is that they have no invalid combinations of inputs, in that the disallowed combinations associated with RS flip-flops is a toggle condition with (JK) flip-flops. One way to represent a (JK) flip-flop is to add a pair of three-input AND gates at the inputs of an (RS) flip-flop, as shown in Fig. 7.

When J=1 and K=0 the flip-flop will set (Q) high on the first clock pulse. Subsequent pulses will have no affect.When J=0 and K=1 the flip-flop will reset (Q) low on the first clock pulse, and subsequent pulses will have no affect.

JK Master –Slave Flip-Flop: The workhorse of flip-flops is the (JK) master-slave flip-flop. This versatile device can be used in virtually every application where the flip-flops already discussed are used without the race problems or intermediate states associated with other flip-flops. Its front panel, block diagram and Sub VI are given in Fig. 8.

On delay timer: The block diagram, front panel and Sub VI of the ON-delay timer are given in Fig.9 Delay time interval is measured in seconds. When the preset value is reached the timer gives an output (high). If the timer is disabled the output instantaneously switches to low logic. The iteration terminal contains the current number of completed iterations, 0 during the first iteration, 1 during the second, and because of that the increment function is



(a) The front panel (b) The block diagram (c)Sub VI

Fig. 9: On Delay Timer

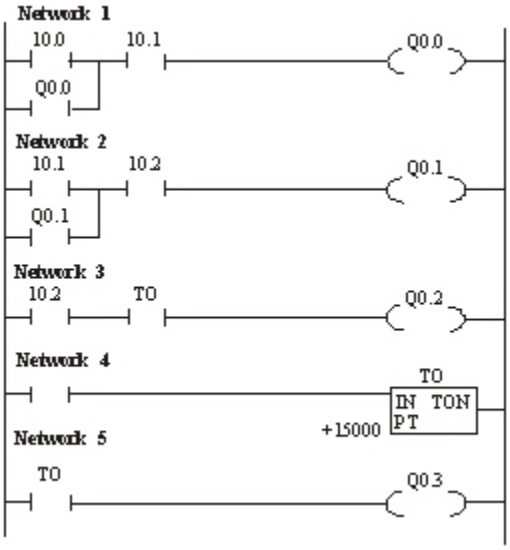


Fig. 10: Siemens ladder diagram of the 7-day Tea Maker

Table 1: Input/Output of the 7-day Tea Maker

Device		Letter	Identification No.		
				PLC1	LabVIEW
Inputs:	*Time switch*	TS	1	10.0	TS
	Float switch	FS	2	10.1	FS
	Thermostat	TH	3	10.2	TH
Outputs:	*Valve*	V1	1	Q0.0	V1
	Heating	E	2	Q0.1	E
	Element		*3*	Q0.2	V2
	Valve	V2	4	Q0.3	B
	Alarm bell	B			

used. Because the Wait Until ms Multiple has a constant measured in ms (100 ms), factor 10 is multiplied by the constant in order to have a delay in seconds.

Considering the above, creation of other VIs which may be utilized in virtual PLCs is straightforward. This includes other different timers, counters, shift register, serial to parallel and parallel to serial data converters and others. Because of the limited size of this paper we cannot illustrate these VIs nowand we are planning to include them in another paper. Instead, we shall demonstrate the validity of the suggested technique in the following 7-day tea maker example Clements-Jewery and Jeffcoat, (1996).

RESULTS AND DISCUSSION

Sequence of operation of the Tea Maker: The operation of the tea maker required is as follows:

- The time switch closes at the appropriate time in the morning and initiates the cycle
- Valve V1 is opened and water fills the kettle K until the float switch FS operates
- This switches off valve V1 and switches on heating element E
- Water in the kettle boils and operates thermostat TH
- This switches the off the element E and switches on valve V2
- Hot water flows into the teapot and V2 must shut off when the teapot is full
- An alarm bell rings to inform the user that the tea is made
- The system relies upon the user to replace the teapot, complete with tea, every day and to fill up the tank each week. The sequence listed is a 'program specification'

Identification of inputs and outputs (Table 1):
The ladder diagram of the 7-day tea maker program using Siemens PLC software (S7-200) (Fig. 10).

The ladder diagram equivalent of the 7-day tea maker program, using LabVIEW (Fig. 11).

From Fig. 11, the above block diagram VI was run and tested. It gave the same operation sequence as the PLC ladder diagram.

CONCLUSION

Making use of LabVIEW Controls and functions palettes, it is possible to create any virtual instrument,

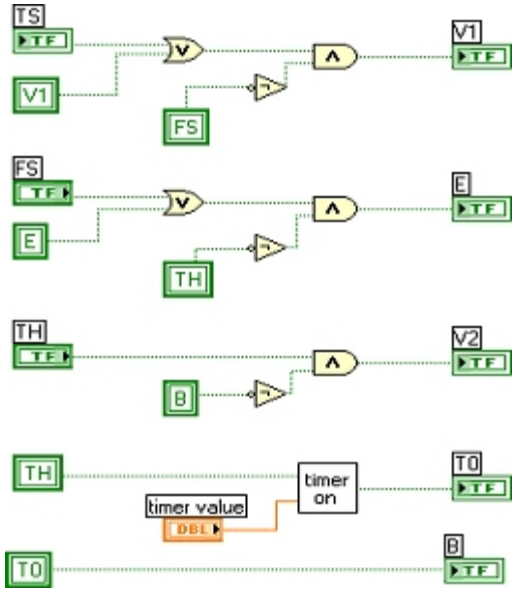


Fig. 11: LabVIEW ladder diagram of the 7-day Tea Maker

which resembles any PLC function, subroutine or network. Thus it is possible to create a PC-based virtual PLC without limitation to the number of virtual PLC components (timers, counters, relays, and shift registers).

Realizing a Virtual PLC improves the programmability of PC-based control systems and makes it possible to gain all the advantages of PC-based control. This includes the unlimited PC memory, running more than one program simultaneously, program connectivity and communication interfacing, and utilizing the front panel as MMI.

REFERENCES

Baily, D. and E.W. Mipenz, 2003. Practical SCADA for Industry (IDC Technology) (Paperback). Newnes Publishers, Oxford UK. ISBN: 07506 58053.

Bryan, L.A. and E.A. Bryan, 1997. Programmable Controllers, Theory and Implementation Industrial Text Company. 2nd Edn., Marietta, USA.

Clements-Jewery, K. and W. Jeffcoat, 1996. The PLC Workbook, Prentice Hall Europe, UK. ISBN: 0-13-489840-0.

Johnson, G.W., 1994. LabVIEW Graphical Programming. McGraw-Hill, Inc., New York, USA. ISBN: 0-07-032692-4.

Parr, E.A., 2001. Programmable Controllers: An Engineer's Guide Newnes. 2nd Edn., Oxford, UK. ISBN: 0-7506-39350.

Saab, R., 2008. National Instruments PACs. Compac RIO a Distributed Intelligent Platform for Implementing DCS-PACs Features. N.I., South WA. USA.

Webb, J.W. and R.A. Reis, 1999. Programmable Logic Controllers Principles and Applications. 4th Edn., Prentice Hall, New Jersey, USA. ISBN: 0-13-679408-4.