International Journal of Trend in Research and Development, Volume 2(5), ISSN 2394-9333
www.ijtrd.com

Development of A Novel Approach For Industrial Process Control

Ihedioha A.C, Eneh I.I. and Eke J.,
Department Of Electrical/Electronics Engineering, Enugu State University of Science and Technology, Enugu, Nigeria.

Abstract: This research is on development of a novel approach for industrial process control. Algorithmic State Machine (ASM) chart, State map and State assignment based upper tank control were implemented. The result shows that input multiplexing and output decoding can be applied to reduce the complexity of any sequential control system but the exact approach would differ depending upon the exact demand of each application. The approach is recommended because it is thorough both in the steps needed to realize the compact Reads Only Memory (ROM) architecture and in the control software development where each statement matches a row of the State Transition Table (STT).

Keywords*: Algorithmic State Machine (ASM), Approach, Industrial, Reads Only Memory (ROM)*

I. INTRODUCTION

A look at an online directory of 58 companies in Nigeria in the category of household appliances and services showed that 12 out of the 58 companies manufacture their products, while the remaining are into sales, marketing, distribution, importation and servicing of the appliances.

It was gathered also that in the category of electrical electronics, that the profile of most engineering companies in the directory is supplies, installation, consultancy, facility management, repair and maintenance as compared to other company profiles in countries like China which involves designing, building and repair [1]. It could be said that the language of engineering companies profile in China is manufacturing and wholesaling, while the language of their Nigerian counterparts is importing and wholesaling.

At the moment most of the automated machineries are of the imported variety. There is need to develop novel approaches that allow process control codes developed for one application to be re-used in a different process control application. This would enhance productivity and help to speed up industrial process control system development. The researcher sees a need to evolve a novel approach to solving these problems that could be applied to any industrial process control requirement.

Intelligent Agent (IA) technology can be used in the monitoring and control of various industrial processes to a very great advantage. When there are several locations in an industrial complex, it becomes difficult for a manager unaided with artificial intelligence to keep track of the quantities of raw materials used for each product line and the estimate of the volume of output [2]. It can be deployed as a tool to help management maintain a firm control of all processes in an industrial complex. With this approach, losses arising from pilfering of raw materials or finished products by operational hands can be determined even as they make returns at the end of their shifts [3]. Management information is captured by intelligent agents from each work site and used to provide detailed analysis via a graphical user interface (GUI). In this paper, a novel approach for industrial process control is presented.

II. DESIGN METHODOLOGY

The control algorithm for the 4 upper beverage tanks is shown in the Algorithmic State Machine ASM chart of Figure 1, while its state map and state assignment are shown in tables 1 and 2. It is desired that a signal STOPRUN be sent to the beverage blending positions whenever any of the four upper tanks is low on stock [4]. A STOPALARM is also issued when this happens. The workforce responsible for feeding the upper tanks with pure beverages is alerted when any of

the tanks is midway full to enable them process more fruits for that tank before it runs out of stock.

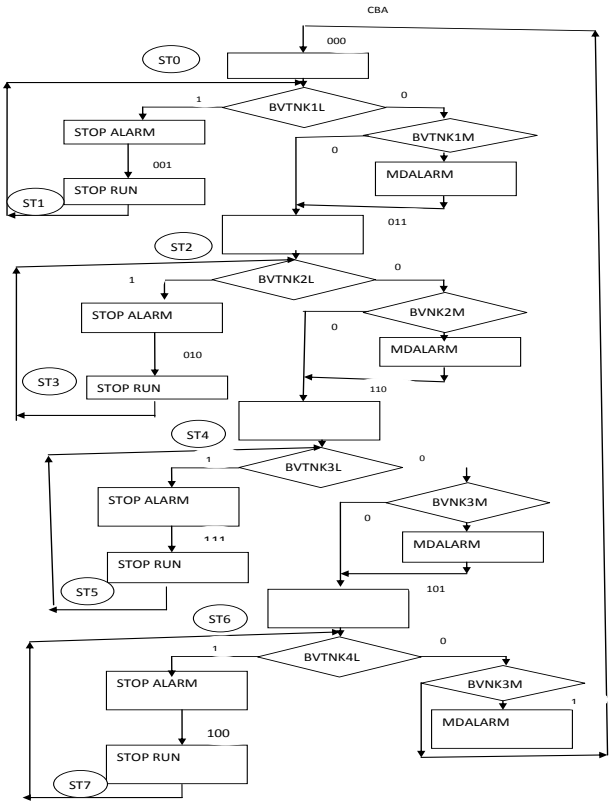

Figure 1: Algorithmic State Machine ASM chart

Table 1: State map

CB \ A	0	1
00	ST0	ST1
01	ST3	ST2
11	ST4	ST5
10	ST7	ST6

Table 2: State Assignment

State Name	State Code
ST0	000
ST1	001
ST2	011
ST3	010
ST4	110
ST5	111
ST6	101
ST7	100

III. THE DESCRIPTION THE ASM CHART

The description of the ASM chart is as follows:

1. At state ST0, the control system checks if beverage tank 1 is low (BVTNKIL). If so, it triggers a STOPALARM and generates a STOPRUN signal at state ST1. It remains in that loops containing ST1 until the situation is remedied.
Again if at ST0, the beverage tank 1 is not low, the system checks if the stock has run down to half tank (BVTNK1M), If BVTNK1M is not yet at the middle position the control system transits to state ST2. If however the beverage tank 1 is at the middle position, the control signal triggers an alarm to alert those processing the fruit so that they know that the tank would run out of stock if more fruit juice is not put there soon enough. The control system then transits to state ST2.

2. At state ST2, the control system checks if beverage tank 2 is low on stock (BVTNK2L) if so, it triggers a STOPALARM and generates a STOPRUN at state ST3. It remains in that loop containing ST3 until the situation is remedied.

3. Again if at ST2 the beverage tank 2 is not low, the system checks if the stock has rundown to half/middle tank (BVTNK2M). If BVTNK2M is not yet at the middle position, the control system transits to state ST3. If however the beverage tank 2 is at the middle position, the control signal triggers an alarm to alert those processing the fruit so that they know that the tank would run out of stock. The control system then transits to the state ST4.
At ST4 the control system checks if beverage tank 3 is low on stock (BVTNK3L). If so, it triggers a STOPALARM and generates a STOPRUN at state ST5. It remains in that loop containing ST5 until that situation is remedied.
Again, if at ST4, the beverage tank 3 is not low, the system checks if the stock has rundown to middle of tank (BVTNK3M). If BVTNK3M is not yet at the middle position, the control system transits to state ST6. If however the beverage tank 3 is at the middle position, the control signal triggers an alarm to alert those processing the fruit so they know the tank would soon run out of

stock. The control system then transits to state ST6.

4. At state ST6, the control system checks if beverage tank 4 is low on stock (BVTNK4L) if so, it triggers a STOPALARM and generates a STOPRUN at state ST7. It remains in that loop containing ST7 until the situation is remedied.
If at ST6, the beverage tank 4 is not low, the system checks if the stock has rundown to middle of tank (BVTNK4M). If BVTNK4M is not yet at the middle position, the control system transits to state ST0. If however the beverage tank 4 is at the middle position, the control signal triggers an alarm to alert those processing the fruit so that they know the tank would run out of stock soon. The control system then transits to state ST0 and the entire cycle is repeated.
The state transition table corresponding to the ASM chart of figure 1 is shown in table 3. Here there are 11 input lines comprised of eight qualifiers (BVTNK1L, BVTNK1M, BVTNK2L, BVTNK2M, BVTNK3L, BVTNK3M, BVTNK4L, BVTNK4M), and three present state codes bits (C, B, A). For a fully expanded state transition table there would be 2^{11}=2048 address lines to decode.

Table 3: STT for Upper Tank Control

Link Path	**Qualifiers**								**Present State**		**Next State**		**State Output**	**Conditional Output**	
	BVTNK1L	BVTNK1M	BVTNK2L	BVTNK2M	BVTNK3L	BVTNK3M	BVTNK4L	BVTNK4M	Name	Code	Name	Code	STOPRUN	STOPALARM	MIDALAR
L1	0	0	-	-	-	-	-	-	ST0	000	ST2	011	0	0	0
L2	0	1	-	-	-	-	-	-	ST0	000	ST2	011	0	0	1
L3	1	-	-	-	-	-	-	-	ST0	000	ST1	001	0	1	0
L4	0	0	-	-	-	-	-	-	ST1	001	ST2	011	1	0	0
L5	0	1	-	-	-	-	-	-	ST1	001	ST2	011	1	0	1
L6	1	-	-	-	-	-	-	-	ST1	001	ST1	001	1	1	0
L7	-	-	0	0	-	-	-	-	ST2	011	ST4	110	0	0	0
L8	-	-	0	1	-	-	-	-	ST2	011	ST4	110	0	0	1
L9	-	-	1	-	-	-	-	-	ST2	011	ST3	010	0	1	0
L10	-	-	0	0	-	-	-	-	ST3	010	ST4	110	1	0	0
L11	-	-	0	1	-	-	-	-	ST3	010	ST4	110	1	0	1
L12	-	-	1	-	-	-	-	-	ST3	010	ST3	010	1	1	0
L13	-	-	-	-	0	0	-	-	ST4	110	ST6	101	0	0	0
L14	-	-	-	-	0	1	-	-	ST4	110	ST6	101	0	0	1
L15	-	-	-	-	1	-	-	-	ST4	110	ST5	111	0	1	0

L16	-	-	-	-	0	0	-	-	ST5	111	ST6	101	1	0	0
L17	-	-	-	-	0	1	-	-	ST5	111	ST6	101	1	0	1
L18	-	-	-	-	1	-	-	-	ST5	111	ST5	111	1	1	0
L19	-	-	-	-	-	-	0	0	ST6	101	ST0	000	0	0	0
L20	-	-	-	-	-	-	0	1	ST6	101	ST0	000	0	0	1
L21	-	-	-	-	-	-	1	-	ST6	101	ST7	100	0	1	0
L22	-	-	-	-	-	-	0	0	ST7	100	ST0	000	1	0	0
L23	-	-	-	-	-	-	0	1	ST7	100	ST0	000	1	0	1
L24	-	-	-	-	-	-	1	-	ST7	100	ST7	100	1	1	0

Fig 2 shows how input multiplexing can be used to reduce the direct qualifier output to two.

In this implementation, the four low level sensor signals are multiplexed such that one is selected at a time.

This permits the presentation to the control system of one low level sensor signal and corresponding midpoint sensor signal at one and the time is required by many of the link paths of figure 1 (the ASM chart). Another advantage of fig 2 is that the number of address lines is now 5 instead of 11 and the control pattern will now compose 2^5=32 rows. Thus, the clever application of multiplexing has significantly reduced the ROM memory demand of the application. The fully expanded state transition table corresponding to figure 2 is shown in table 4.

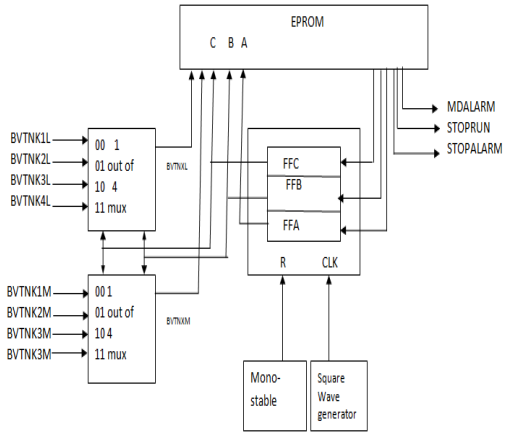

Figure 2: ROM Based Upper Tank Control Using More Than One Input Multiplexer

The foregoing shows that input multiplexing and output decoding can be applied to reduce the complexity of any sequential control system but the exact approach would differ depending upon the exact demand of each application.

This makes the approach quite universal as it can be adapted to match the needs of any particular control application [5].

Table 4: State Transition Table (STT) for the Optimized ROM Solution of Fig. 2

Link Path	Qualifiers		Present state		Next state		State Output	Conditional Output		
	BVTNKXL	BVTNKXM	Name	Code	Name	Code	STOPRUN	STOPALARM	MIDALARM	HEX OUTPUT
L1	0	0	ST0	000	ST2	011	0	0	0	18
L2	0	1	ST0	000	ST2	011	0	0	1	19
L3	1	0	ST0	000	ST1	001	0	1	0	0A
L3	1	1	ST0	000	ST1	001	0	1	0	0A
L4	0	0	ST1	001	ST2	011	1	0	0	1C
L5	0	1	ST1	001	ST2	011	1	0	1	1D
L6	1	0	ST1	001	ST1	001	1	1	0	0E
L6	1	1	ST1	001	ST1	001	1	1	0	0E
L7	0	0	ST2	011	ST4	110	0	0	0	30
L8	0	1	ST2	011	ST4	110	0	0	1	31
L9	1	0	ST2	011	ST3	010	0	1	0	12
L9	1	1	ST2	011	ST3	010	0	1	0	12
L10	0	0	ST3	010	ST4	110	1	0	0	34
L11	0	1	ST3	010	ST4	110	1	0	1	35
L12	1	0	ST3	010	ST3	010	1	1	0	16
L12	1	1	ST3	010	ST3	010	1	1	0	16
L13	0	0	ST4	110	ST6	101	0	0	0	28
L14	0	1	ST4	110	ST6	101	0	0	1	29
L15	1	0	ST4	110	ST5	111	0	1	0	3A
L15	1	1	ST4	110	ST5	111	0	1	0	3A
L16	0	0	ST5	111	ST6	101	1	0	0	2C
L17	0	1	ST5	111	ST6	101	1	0	1	2D
L18	1	0	ST5	111	ST5	111	1	1	0	3E
L18	1	1	ST5	111	ST5	111	1	1	0	3E

L19	0	0	ST6	101	ST0	000	0	0	0	00
L20	0	1	ST6	101	ST0	000	0	0	1	01
L21	1	0	ST6	101	ST7	100	0	1	0	82
L21	1	1	ST6	101	ST7	100	0	1	0	82
L22	0	0	ST7	100	ST0	000	1	0	0	04
L23	0	1	ST7	100	ST0	000	1	0	1	05
L24	1	0	ST7	100	ST7	100	1	1	0	26
L24	1	1	ST7	100	ST7	100	1	1	0	26

IV. STATE TRANSITION TABLE BASED CONTROL SOFTWARE DESIGN PROCEDURE

The following steps where developed towards achieving a STT based control software design for micros [6]. They include:

1. Put down the specifications for the envisaged system.
2. Develop the corresponding Algorithmic State Machine (ASM) chart.
3. Derive the STT corresponding to the ASM chart.
4. Reduce input/output line demand of the application using input multiplexing and output decoding if need be.
5. Modify STT by placing the State Code column before the Qualifiers' column
6. Write Case Construct based implementation of the STT in software.
7. Fit software into the microcontroller.

CONCLUSION

This research has presented a novel approach for industrial process control. We saw that one way of implementing a complex control system is to use the compact ROM design in the external architecture and state transition table based control software in the internal architecture. This approach is recommended because it is thorough both in the steps needed to realize the compact ROM architecture and in the control software development where each statement matches a row of the STT.

References

[1] M. Mercian, V. Korclic, A. Zoitl, A. Lazinica. Knowledge based multi-agent architecture. Computational Intelligence for modelling, Control and Automation. Pp 55, 2006.

[2] D. Camacho, R. Aler, D. Borrajo and J. M. Molina. A multi-agent architecture for intelligent gathering systems. Al communications, The European Journal on Artificial Intelligence, IOS Press, Vol. 18, Issue 1, pp. 15-32, 2005.

[3] S. Ahmead and M. U. Bokhari. A new approach to multi-agent architecture for secure and effective E-learning. International journal of computer applications. Vol 46, no 22, 2012.

[4] F. P. Maturana and Douglas H. Norric. Multi-agent mediator architecture for distributed manufacturing. Journal of intelligent manufacturing. Vol 7, issue 4, pp 257-270, 1996

[5] A. A. Betanzos, B. G. Berdinas and J. A. Suarez Romero. A multi-agent architecture for intrusion detection. 6th international conference on knowledge-based intelligent information and engineering services (KES 2002) Crema, Italy. 2002.

[6] M. Kolp, P. Giorgini, and J. Mylopoulos. Organizational multi-agent architectures: a mobile robot example. AAMAS' 02 proceedings of the first international conference on autonomous agents and multi-agent systems. Pp 94-95, 2002.