A Cost-Effective Approach to the Design and Implementation of Microcontroller-based Universal Process Control Trainer

[1]Udeze Chidiebele. C, [3] Uzedeh Godwin,
[1,3] R & D Department, Electronics Development Institute (FMST-NASENI), Awka, Nigeria.

[2]Prof. H. C Inyiama,[4] Dr C. C. Okezie,
[2,4] Electronics and Computer Engineering Department, Nnamdi Azikiwe University, Awka, Nigeria.

Abstract—**This paper presents a novel approach to the design and implementation of a low-cost universal digital process control trainer. The need to equip undergraduates studying Electronic Engineering and other related courses in higher institutions with the fundamental knowledge of digital process control practical was the main objective of the work. Microcontroller-based design and implementation was the approach used where only one AT59C81 with few flip-flops were used for the whole eight processes covered by the trainer thereby justifying its low-cost and versatility.**

Keywords- process control; control algorithm; Algorithmic State machine (ASM) chart; State Transition Table (STT); Fully-expanded STT; Control software.

I. Introduction

The basic objective in process control is to regulate the value of some quantity which means to maintain that quantity at some desired value regardless of external influences. The desired value is called the reference value or set point [2]. Inyiama H. C and Okezie C.C (2007) stated that a process control is typically a sequential logic system whose control algorithm can be represented in the form of a flow chart called an algorithmic State machine (ASM) chart [3] or in the form of State Transition Diagram (STD) [4]. The ASM chart is a diagrammatic description of the output function and the next-state function of a state machine to implement an algorithm and becomes part of the design documentation when completed. The symbols covered are the STATE BOX, THE DECISION BOX, THE CONDITIONAL OUTPUT BOX and ASM BLOCK.

According to Clare C.R [1973], the ASM chart has three basic elements: the state, the qualifier and the conditional output [3]. The need to equip undergraduates studying Electronic Engineering and other related courses in higher institutions with the fundamental knowledge of digital process control practical was the main objective of this work. The design approach used was the use of Algorithmic State machine (ASM) charts State Transition Diagrams (STD), State Transition Tables (STT), Fully Expanded STT, and Link Path Addressable ROM Structure. The significance of the work extends to the fact that it presented a standard approach to the design and implementation of the process control systems that can be applied to any process controls in the industries and it also bridges the gap created in the lives of these students due to their lack of exposure to electronics practical.

The following processes were covered by the trainer:

- Temperature Level Control systems
- Automatic Liquid Dispenser System
- Automatic Water Pump control system
- Traffic Light Control System
- Upper Tank Water level Control systems
- Lower Tank Water level Control systems
- Automatic Gate control system
- Automatic Street Light Control system

II. Methodology & Design Analysis

The design of the digital process control system can be achieved through various methods which include: Gate-Oriented Design which involves equation-to-gate conversions, map simplifications, output function synthesis, and next-state function synthesis, flip flops, state assignment and hazards. When such discrete logic gates (such as AND, NAND, OR, NOR, EX-OR, INVERTERS etc) and memory flip flops are used, what is termed a RANDOM LOGIC system results. Since such a system involves mostly small scale integration components, the component count is usually high for a fairly complex circuit. This implies several interconnections and many potential sources of error. Modifications and maintenance are also difficult to achieve when either re-designing or fault-finding becomes necessary. A random logic system is therefore very inflexible.

Fortunately however, logic systems can be implemented in forms more structured than random logic. Such systems employ structured logic devices such as Multiplexers (or Data Selectors) (5), and Read-Only-Memories (ROMs). A multiplexer-based controller requires as many multiplexers as there are columns in the bit-pattern sequence to be generated and each of these multiplexers must have at least as many data input lines as there are rows in the bit-pattern sequence. Thus, if an 8-bit pattern is to be generated by means of multiplexers, a total of 64 data input lines would be involved. This is in addition to other control inputs and outputs necessary in such a system. The rapid proliferation of data input lines (and hence potential sources of error) in multiplexer based complex sequential logic systems is its main disadvantage in such applications. Multiplexers being structured logic devices do, however have a considerable advantage over random logic in that errors in the design can be corrected simply by altering the logic levels applied to their data inputs. For compact, reliable and easily maintainable implementation of a complex sequential logic ROMs have an edge over multiplexers and are usually preferred.

The microcontroller-based implementation was chosen for this work due to its numerous advantage over the others which include its simplicity, and the fact that a simple control software can be developed which can be used without any modification in any control system, no matter how complex or how simple, and even when the numbers of input qualifiers, state code, size and number of output lines differ from one control system to the other, provided one input port is sufficient for Address inputs and one output port for the control pattern output. The system is made up of the hardware subsystem and software subsystem. The hardware subsystem is comprised of the input interface, the control systems and the output interface.

The Structure of the Microcontroller-based Digital Process Control is shown in Fig. 2 below. It comprise of the microcontroller with its input and output ports, the input interface subsystem connected to the input port of the microcontroller, the output interface subsystem connected to output port of the microcontroller. The input interface system comprise of all the sensors that will be used for a particular process. For example in the temperature level control system, LM 35 is the sensor, which senses the temperature of the water container. Also the output interface comprises all the transducers such as light emitting diodes, LCDs and so on. Note that buttons and keypads are part of the input subsystems since they are used for selecting the particular process to be controlled.

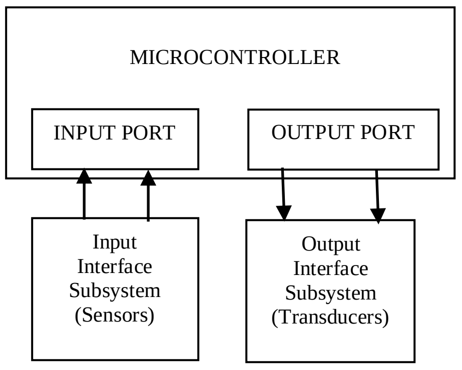
MICROCONTROLLER
INPUT PORT
OUTPUT PORT
Input Interface Subsystem (Sensors)
Output Interface Subsystem (Transducers)

Figure 4. The Structure of the Microcontroller-based Digital Process Control.

III. THE GENERATION OF CONTROL BIT- PATTERN SEQUENCE

The procedure that is followed for the generation of the control bit-patterns include: the drawing of ASM chart for each of the processes, transforming the ASM charts into a state transition table, and expanding the state transition table fully so that the location address and the content address for the process are generated which is then burned into the flash drive of the microcontroller.

This procedure was used for all the processes to generate the control bit-patterns in Table 3 below but the design process that led to that was illustrated in this paper by one of the processes which is the temperature level control system of water or other liquids . The ASM chart of the temperature control system is shown in Fig. 2 below. These ASM chart is then transformed into a state transition table of Table1 and then to fully expanded state transition table of Table2.

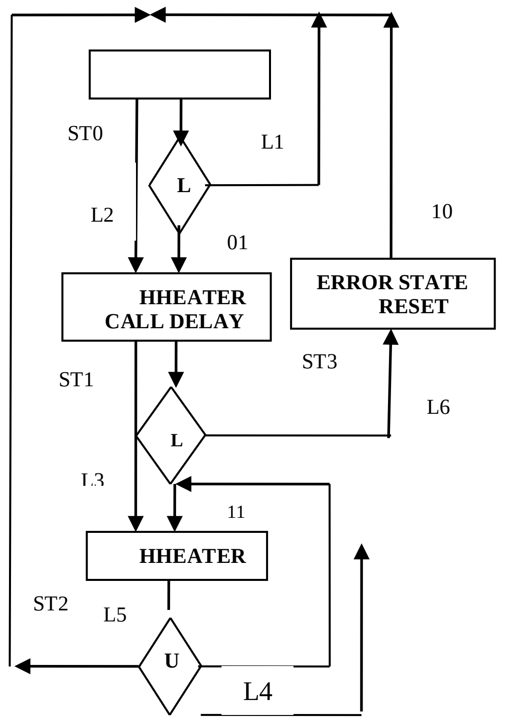

Figure 5. (i)The ASM chart of the temperature control system

2 (iii) STATE ASSIGNMENT 2 (ii) STATE MAP

STATE NAME	STATE CODE
ST0	0 0
ST1	0 1
ST2	1 1
ST3	1 0

B \ A	0	1	
	ST0	ST1	0
	ST3	ST2	1

The labels or names inside some of the state boxes are the state outputs. The ASM charts of the Fig. 2 above have only one state output namely: HHEATER. The logic level of the output signal is high or active when the control system is in that state. The bit pattern at the top right end of the state box is its state code. The letters B, A above the state code signify that two flip flops B and A are used to represents the various states of the machine.

The state codes are the logical levels at the Q outputs of these two flip flops respectively. Each rectangular box in the

ASM chart is a state box. The word (e.g ST0) enclosed in a circle at the bottom left hand corner of a state box is the state name. Here ST0 stands for state 0, and similar interpretations apply to the other states, hence ST1 Means state 1, and so on. Each decision box has one entry path and two exit paths. The exact value of an input qualifier determines which exit path is followed out of a decision box. In the ASM chart of Fig. 2 Uth and Lth are the input qualifiers.

With the help of a K_map in which the state names are inserted serially in an adjacent cells (Fig. 2ii) and which is called a state map, the state codes are chosen such that only one bit changes level as one moves from one state of the control system to another. This is clearly brought out in the state assignment of Fig. 2iii. This method of state assignment facilitates complexity reduction when hardwired logic is the preferred technique of control system implementation and helps to prevent race hazards [3].

Every ASM chart has an equivalent tabular representation known as a State Transition Table (STT) [3]. An ASM chart can be fully described in terms of the link paths comprising it. A State machine such as is represented by an ASM chart attempts to change state (i.e. transits from the present state to the next) when a clock pulse occurs. A link path is a path followed from the present state back to itself or to another state, when the clock pulse arrives. When there is an input qualifier between the present state and another, the logic level of the qualifier determines the next state the machine goes to at the clock pulse. If there is no qualifier between the present state and the next, the machine must unconditionally transit from its present state to the adjacent state in the forward direction, when a clock pulse arrives.

The ASM chart of Fig. 2i has 7 link paths labeled L1 to L7. L1 is the link path from state 0 back to itself when the input qualifier is 0. Also L2 is the link path from state 0 to state 1 when the input qualifier Lth is 1. Similarly, L3 represents the transition from state ST1 to state 2 when the qualifier Lth is 0. L4 is the transition from state2 back to itself when the input qualifier Uth is 0 and L5 is the transition from state 2 to state 0 when Lth input qualifier Uth is 1. L6 is the transition from state 1 to state 3 which is the error state where the system stays until a key is pressed to return it to state 0 through link path L7.

In the STT of Table 1 a number of dashes appear under the columns headed by the input qualifiers Lth and Uth. A dash (_) implies that the input qualifier above that column is not relevant for the transition being made in the link path shown on the STT row where the dash appeared. That input qualifier may become relevant in some other link path transition and then its value would become 0 or 1, rather than a dash. A dash in the STT also means that the input qualifier that appears as column heading for that dash may be at logic 0 without affecting the control process at that material time.

A State Transition Table (STT) is said to fully-expanded when all the dashes on each row are given all possible combination of logic values, leading to new rows in the State Transition Table, one for each combination of the logic values for the dashes on that row. In this respect, a STT data row with one row when that (dashed) qualifier is given the logic value 0 and the other when that qualifier is given the logic level 1. Similarly a STT data row with 2 dashes expands into 4 STT rows. Assume the (dashed) qualifiers are represented by q1, q2. Then the first STT row in the expansion will give q1, q2 = 0, 0, the second would have q1, q2 = 0, 1, the third row would have q1, q2 = 1, 0, the fourth row would have q1, q2 = 1,1. Three dashes on an STT data row would in like manner lead to 8 rows in the fully expanded STT of Table 2 results.

IV. Microcontroller-Based Implementation

The high capacity of ROMs relative to the number of unique bit-patterns in a fully expanded STT suggests the use of a single ROM to store the fully expanded bit-pattern of all the processes. The link-path addressable word structure is based on storing the next state and the output for each link path in the ASM charts. The next-state portion of the ROM word is called the LINK PART, while the output portion of the ROM word is called the INSTRUCTION PART. Each address is a function of the present state and qualifier inputs and called a link-path address. In general any process described by an ASM chart can be implemented with this structure, called a link-path addressable ROM.

Since a microcontroller-based implementation is used, the same link path addressable ROM patterns will be programmed into its ROM or the Erasable and Programmable ROM (i.e. EPROM) or Flash Drive (now available in newer microcontrollers). The ADDRESS inputs to the Address Decoder part of the ROM device would now be input via an input port of the microcontroller which is then used to locate the corresponding NEXT STATE and STATE OUTPUT.

When a clock pulse occurs, the NEXT STATE pattern becomes the present state pattern. This joins the input qualifiers to comprise the next Address input to be used by the microcontroller. A power-up one-shot applied to the D flip flops that feedback the Next State as the present state when a clock pulse occurs, initializes the system to state 0 at the start up[6]. Thereafter, the control system behaves as defined by the ASM chart, with the help of the control software running in the microcontroller. The user may use a simple switch to stop the control software run.

A. *The Control Software*

A very important universal concept that is a natural outcome of the use of a fully expanded STT in a link path addressable ROM structure, as part of a microcontroller-based implementation is that a simple software can be developed which can be used without any modification in any control system, no matter how complex or how simple, and even when the numbers of the input qualifiers, state code, size and number of the output lines differs from one control system to the other, provided one input port is sufficient for address inputs and one output port for control pattern output.

The flowchart for this universally applicable control software is shown in the Fig. 3 below; with a pseudo code that explains what it does above it.

The control cycle time is the time delay required for the system to settle and become ready for the next round of input-output. Its default value is zero seconds.

TABLE 1. THE STT FOR TEMPERATURE LEVEL CONTROL.

LINK PATHS	INPUT QUALIFIERS Lth	INPUT QUALIFIERS Uth	PRESENT STATE NAME	PRESENT STATE CODE B	PRESENT STATE CODE A	NEXT STATE NAME	NEXT STATE CODE B’	NEXT STATE CODE A’	STATE OUTPUT HHEATER
L1	0	_	ST0	0	0	ST0	0	0	0
L2	1	_	ST0	0	0	ST1	0	1	0
L3	0	_	ST1	0	1	ST2	1	1	1
L4	_	0	ST2	1	1	ST2	1	1	1
L5	_	1	ST2	1	1	ST0	0	0	1
L6	1	_	ST1	0	1	ST3	1	0	1
L7	_	_	ST3	1	0	ST0	0	0	0

TABLE 2. FULLY EXPANDED STT TABLE FOR THE SYSTEM

LINK PATH	LOCATION ADDRESS (hex)	ADDRESS PATTERN Lth Uth B A	CONTENT PATTERN B’ A’ H	LOCATION CONTENT (hex)
L1	0 0 0 4	0 0 0 0 0 1 0 0	0 0 0 0 0 0	0 0 0 0
L2	0 8 0 C	1 0 0 0 1 1 0 0	0 1 0 0 1 0	0 2 0 0
L3	0 1 0 5	0 0 0 1 0 1 0 1	1 1 1 1 1 1	0 7 0 7
L4	0 3 0 B	0 0 1 1 1 0 1 1	1 1 1 1 1 1	0 7 0 7
L5	0 7 0 F	0 1 1 1 1 1 1 1	0 0 1 0 0 1	0 1 0 1
L6	0 9 0 D	1 0 0 1 1 1 0 1	1 0 1 1 0 1	0 5 0 5
L7	0 2 0 6 0 E 0 A	0 0 1 0 0 1 1 0 1 1 1 0 1 0 1 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0

B. Pseudo Code

```
BEGIN
    Repeat
    Input Next Control Pattern Address from Input Port;
        Retrieve Next Control Pattern from Link-path
        Address Rom Structure;
    Output Next Control Pattern Address from Output Port;
    Delay for Control Cycle Time.
    Until HSTOP = 1
END
```

C. *Flow Chart*

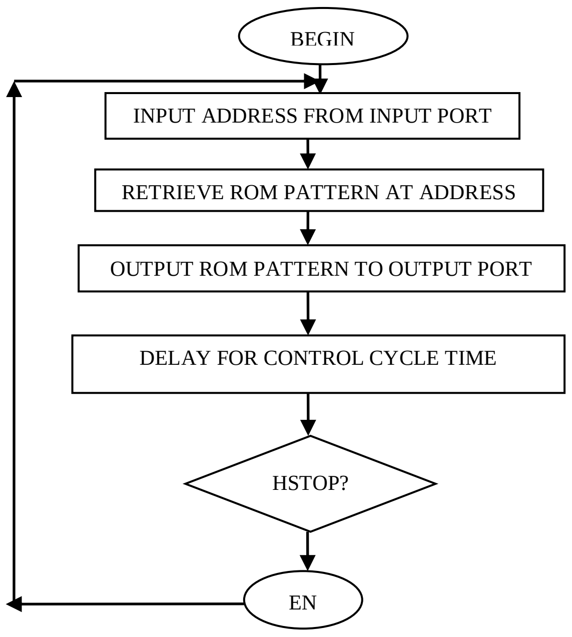
BEGIN
INPUT ADDRESS FROM INPUT PORT
RETRIEVE ROM PATTERN AT ADDRESS
OUTPUT ROM PATTERN TO OUTPUT PORT
DELAY FOR CONTROL CYCLE TIME
HSTOP?
EN

Figure 6. The Flowchart rep. of the control software

V. RESULTS AND DISCUSSION

A software-based universal digital process control system was achieved with the program design method presented. Following the procedure for developing a fully-expanded state transition table, which begins with representing the process in an ASM chart, developing a state transition table for the process from the ASM chart and then translating the state transition table to a fully-expanded state transition table, the fully-expanded state transition table for the other processes was also developed and all of them were shown in Table 3. The location address is the input address of the processes, which is used to locate where in the ROM memory the location content is stored. The location address and the location content will be burned into the ROM permanently.

The difference in terms of programming effort, between a software-based controller intended for just one control system and that designed for several control systems (each with a different number of input qualifiers) is minimal. Fig. 4 illustrates a typical microcontroller-based digital process control system. With this trainer the student can perform several experiments on each of the processes as covered in the laboratory manual developed for the trainer which enables him gain the needed practical knowledge on digital process control. The keyboard (Fig. 4) is used for selecting the particular process desired to be controlled. It also facilitates the input of variables or control parameters which make software-based controllers very flexible, while the LCD display enables the microcontroller to transmit responses to user commands in addition to providing the current status of the controlled device where necessary.

TABLE 3. LOCATION ADDRESS & CONTENT FOR THE CONTROLLER ROM FROM THE FULLY-EXPANDED STT OF THE 8 PROCESSES.

Location address (hex)	**Location content (hex)**	**Location address (hex)**	**Location content (hex)**
0 0 0	0 0	4 0 0	0 0
0 0 4	0 0	4 0 4	0 0
0 0 8	0 2	4 0 8	0 2
0 0 C	0 0	4 0 C	0 0
0 0 1	0 7	4 0 1	0 7
0 0 5	0 7	4 0 5	0 7
0 0 3	0 7	4 0 3	0 7
0 0 B	0 7	4 0 B	0 7
0 0 7	0 1	4 0 7	0 1
0 0 F	0 1	4 0 F	0 1
0 0 9	0 5	4 0 9	0 5
0 0 D	0 5	4 0 D	0 5
0 0 2	0 0	4 0 2	0 0
0 0 6	0 0	4 0 6	0 0
0 0 E	0 0	4 0 E	0 0
0 0 A	0 0	4 0 A	0 0
1 0 4	0 2	5 0 0	0 2
1 0 8	0 2	5 0 4	0 2
1 0 C	0 2	5 0 8	0 2
1 1 0	0 6	5 0 C	0 2
1 1 4	0 6	5 1 0	0 6
1 1 8	0 6	5 1 4	0 6
1 1 C	0 6	5 1 8	0 6
1 0 1	0 4	5 1 C	0 6
1 0 5	0 4	5 0 1	0 4
1 1 0	0 4	5 0 5	0 4
1 1 5	0 4	5 1 0	0 4
1 0 9	0 C	5 1 5	0 4
1 0 D	0 C	5 0 9	0 C
1 1 9	0 C	5 0 D	0 C
2 0 3	0 D	6 0 3	0 D
2 0 B	0 D	6 0 B	0 D
2 1 3	0 D	6 1 3	0 D
2 1 B	0 D	6 1 B	0 D
2 0 7	0 9	6 0 7	0 9
2 0 F	0 9	6 0 F	0 9
2 1 7	0 9	6 1 7	0 9
2 1 F	0 9	6 1 F	0 9
2 0 2	0 8	6 0 2	0 8
2 0 6	0 8	6 0 6	0 8
2 1 2	0 8	6 1 2	0 8
2 1 6	0 8	6 1 6	0 8
2 0 A	0 0	6 0 A	0 0
2 0 E	0 0	6 0 E	0 0
3 0 4	0 0	7 0 0	0 0
3 0 8	0 2	7 0 4	0 0
3 0 C	0 0	7 0 8	0 2
3 0 1	0 7	7 0 C	0 0
3 0 5	0 7	7 0 1	0 7
3 0 3	0 7	7 0 5	0 7
3 0 B	0 7	7 0 3	0 7
3 0 7	0 1	7 0 B	0 7
3 0 F	0 1	7 0 7	0 1
3 0 9	0 5	7 0 F	0 1
3 0 D	0 5	7 0 9	0 5

TABLE 4. COST IMPLICATIONS OF THE PROJECT

	ITEM	QTY	Unit Cost (Naira)	Cost (Naira)
1	Sensors	7	400	2800
2	ADC0804	1	200	200
3	Microcontroller, AT89C51	1	350	350
4	4x20 Liquid crystal display	1	2500	2500
8	DC relays (12V & 6V)	2	70	140
12	10k ohm resistors	7	5	35
13	Transistor, BC337	5	5	25
14	Diodes, 1N4001	6	5	30
15	10kohm Variable resistor	2	20	40
16	10uf & 33pf capacitor	2	20	40
19	7805	1	40	40
21	Copper clad board	1	100	100
22	40 pin IC socket	1	40	40
23	Soldering Lead	1	150	150
25	Casing materials	-	1000	1000
	TOTAL			7490NGN

Table 4 shows the cost implication of the project. The total cost implication of the project is 7490 NGN (Nigerian Naira). When compared to the cost implication of the same project using other methods such gate-oriented designed method, multiplexer and ROM-based design method as shown in Table 5, it is cheaper thereby justifying the cost-effectiveness of the approach used in this work.

TABLE 5. COST COMPARISON WITH OTHER METHODLOGIES

Methodology	Cost (NGN)
Gate-oriented design method	12,350
Multiplexer-based design method	10,200
ROM-based design method	9,400
Microcontroller-based design method	7,490

VI. CONCLUSION

The use of a single microcontroller to control several processes, based on storing the fully expanded State Transition Tables of those processes in its ROM or flash drive makes possible the realization of a low-cost universal processes control trainer.

REFERENCES

[1] Inyiama H. C, Okezie C.C, Designing microcontroller-based universal process control systems, Volume 2 Number 2 (Electroscope), November 2007. Department of Electrical and Electronics Engineering, Nnamdi Azikiwe University Awka. Pp.11-26.

[2] Curtis D Johnson, Process Control Instrumentation Technology, 8th Edition, 2006, Prentice-Hall Inc. pp. 1-10.

[3] Clare C.R, Designing logic systems using state machines (U.K, MacGraw-Hill, 1973) pp 1-108.

[4] Roger L. Tokheim, Digital Electronics, Principles and Applications Fifth Edition,1999, Glencoe McGraw-Hill. pp269-276.

[5] Williams, G.E., "Digital Technology, Principles and Practice". Science. Research Associats Inc., (1974), pp. 96-105,202-245.

[6] Walter G. Jung IC timer cookbook (4300West 82nd St. Indianapolis 48258 USA, Howard W. Sams & Co. Inc. 1977) pp1-36.

[7] Michael J. Pont, Embedded C, 2002, Pearson Education Limited, pp 17-34.

[8] Bertram J. E: 'The concept of state in the the analysis of discrete time control system,' 1962 Joint Autom. Control Conf. New York University (June 27-29), Paper No. 11-1

[9] Inyiama H.C, Unpublished lecture Notes on Real-Time computing and control, Department of Electronic and Computer Engineering, Nnamdi Azikiwe University Awka 2008.

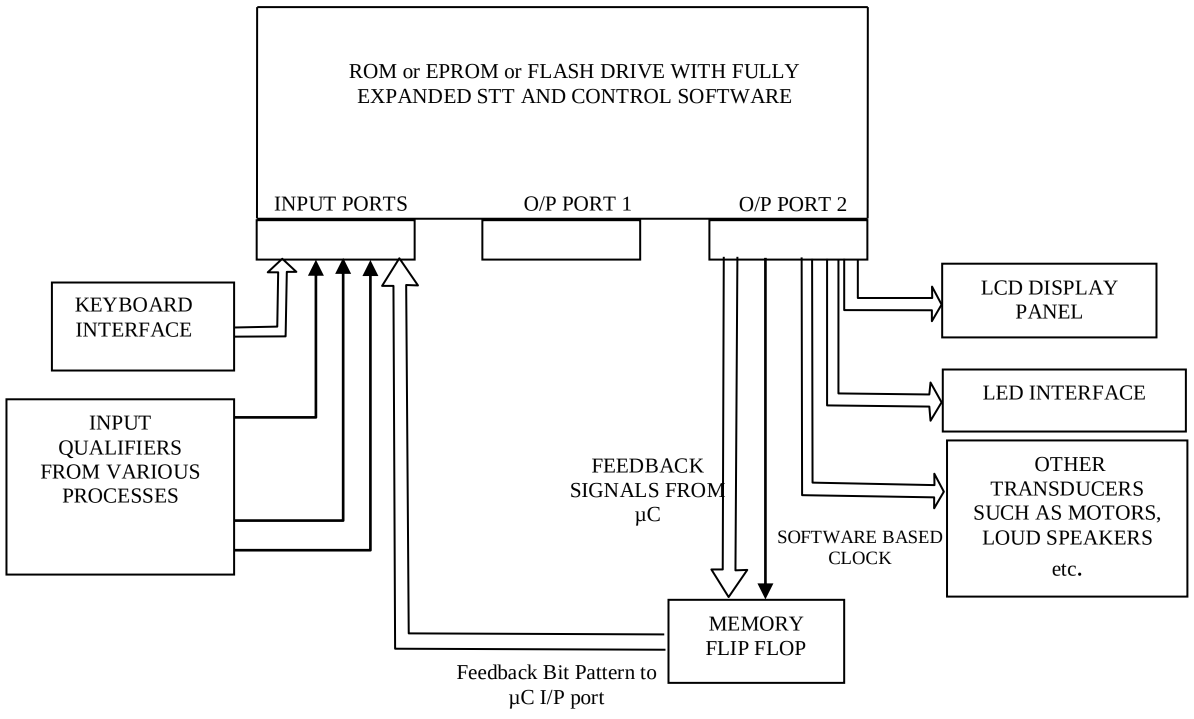
ROM or EPROM or FLASH DRIVE WITH FULLY EXPANDED STT AND CONTROL SOFTWARE
INPUT PORTS
O/P PORT 1
O/P PORT 2
KEYBOARD INTERFACE
INPUT QUALIFIERS FROM VARIOUS PROCESSES
LCD DISPLAY PANEL
LED INTERFACE
OTHER TRANSDUCERS SUCH AS MOTORS, LOUD SPEAKERS etc.
FEEDBACK SIGNALS FROM µC
SOFTWARE BASED CLOCK
MEMORY FLIP FLOP
Feedback Bit Pattern to µC I/P port

Figure 4. Microcontroller-based digital process control system