An Algorithm for the Conduct of Multiple Simultaneous Multi-Party Elections Using a Microcontroller.

Jonathan A. Enokela, Ph.D[1*] and Charles C. Osuagwu, Ph.D.[2]

[1]Department of Electrical Engineering, University of Agriculture, P.M.B. 2373, Makurdi, Nigeria.
[2] Department of Electronic Engineering, University of Nigeria, Nsukka, Nigeria.

E-mail: jonajapeno@gmail.com*

ABSTRACT

Hitherto the electronic voting machines used for elections were designed to handle one or two elections at a time. The conduct of multiple simultaneous multi-party elections becomes a necessity not only to solve the problem of the bandwagon effect but also to reduce costs of organizing elections. An algorithm for the conduct of such elections is developed and implemented using a microcontroller, for cost effective solutions. The algorithm which is implemented as actual computer code forms part of the firmware of an electronic voting machine. The accuracy of the algorithm was verified by building a real system and burning the code into the program memory of a microcontroller. The implementation and incorporation of the algorithm described here into electronic voting machines would not only remove the constraint of holding one or two elections at a time but would also remove the negative psychological effects associated with such elections.

(Keywords: algorithm, microcontroller, elections, electronic voting)

INTRODUCTION

Many electronic voting machines (EVM) are designed to handle one or two elections at a time [1-3]. In election environments where there are many elected positions to be contested the organization of only one or two elections at a time could lead to the bandwagon effect. When a political party has won the first election it begins to let it be known to all and sundry that anyone who does not support it in subsequent elections risks being in opposition and thus may not benefit from government patronage. This forces voters into the party's bandwagon in subsequent elections [4]. The negative impacts of bandwagon effect are the reduction of voter turnout in subsequent elections and the biasing of voters minds in favor of the winning party thus making subsequent elections not to be free and fair. The organization of multiple simultaneous elections leads to the reduction of overall election costs due to the reduced costs of logistics and the involvement of fewer poll workers.

Most modern democracies are multi party in nature. Thus an electronic voting machine must not only be able to handle multiple simultaneous elections, it must also be able to consider the involvements of many political parties simultaneously. A detailed description of the architecture of such EVM is given in [4].

PROGRAM ROUTINES FOR THE ELECTRONIC VOTING MACHINE

When a voter presents himself at the terminal of the voting machine, he presses a button corresponding to a party to indicate his preference. In order to record the voter's preference correctly, various questions must be answered by the firmware of the EVM. These include determining the correct party for which the voter has cast his vote, registration of the vote in the correct memory location in the case of multiple elections, and ensuring that the votes cast cannot be more than the number of registered voters for the particular polling unit. The flow chart shown in Figure 1 gives an overview of the routines that the machine processes in order to answer these questions. The keyboard of the machine is monitored by the scan keyboard routine which determines if a key press is valid. The party for which the key is pressed is determined by the party score routine.

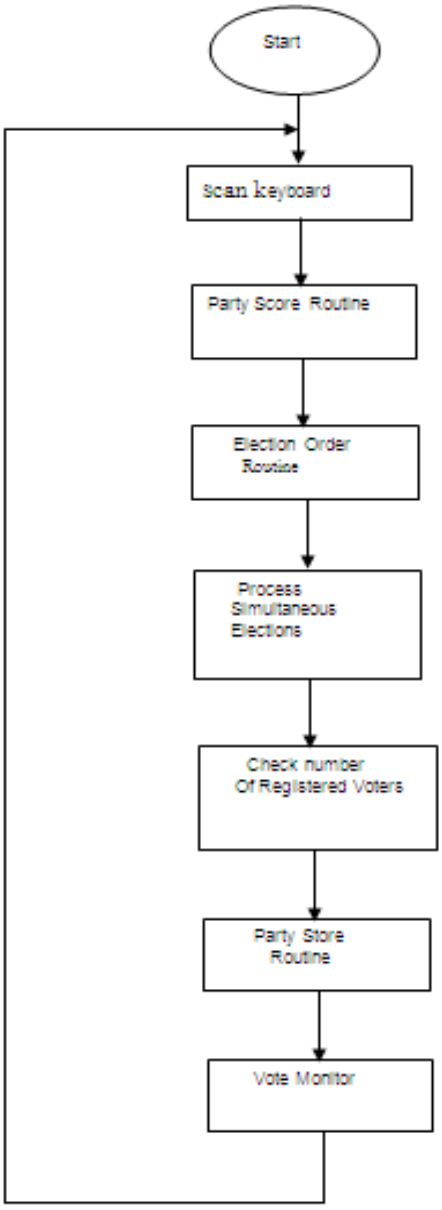

Figure 1: Major Routines for the EVM.

This routine also monitors the number of times a voter has pressed the buttons on the keyboard so that together with the election order routine the voter's preference can be recorded in the appropriate location in the memory of the machine. Before the vote is recorded, however, the number of simultaneous elections being held and the number of voters registered in the particular polling unit must be checked to ensure that the voter can only vote as many times as the number of simultaneous elections available as well as to ensure that the number of votes recorded for the particular polling unit cannot exceed the number of registered voters. The voter's preference is finally stored in the memory using the party store routine. The vote monitor routine increments the number of voters by one to complete the process.

If there is only one election, the voter's preference would be recorded without difficulty. Let us suppose that there are many elections; say seven (7). The machine should be useful whenever any combinations of the seven elections are organized. In order to have an efficient coding a constraint must be imposed on the algorithm. This requires that the elections to be held should be arranged in a hierarchical order. Thus if seven simultaneous elections are being held, the order in which the button presses will be recorded for the parties are President (P), Governor (G), Chairman (C), Senate (S), Representative (R), Assembly (A), and Council (K), these also being the elective offices to be contested. These are the elective offices available in Nigerian elections [5], [6]. These offices can be varied for other countries as desired.

The locations (addresses) for each of these offices for every party are fixed in the EEPROM memory of the microcontroller. Let us consider the situation in which some elections are not held. Consider, for instance, that the governorship election is not being held or any combinations of elections are organized simultaneously while the others are not. It is necessary that if the first press of the button is for the highest of the offices being organized then the subsequent button presses must be recorded for subsequent parties in a hierarchical orderly manner. In other words if the elections being organized are President, Chairman, Senate, and Council, say, when the voter presses the button the first time, it must be recorded for President, the second time for Chairman, the third time for Senate and the fourth time for the Council of any party of the voter's choice.

When all the elections are being held simultaneously, the decision to be taken in order to assign votes to the correct party is straightforward. In a multiple election environment with combinations of these elections only being held, the decision making process becomes much more complex. It turns out that the number of decisions to be taken is a combinatorial function involving the maximum number of available elections N_E (in this case seven) and the number of button presses, N_B, i.e. the number of times a voter has pressed the buttons corresponding to the party of his choice on the interface of the EVM. The number of decisions that needs to be made for correct assignment of votes to the desired memory location at the n^{th} press of the button by the voter is given by Equation (1).

Number of decisions at n^{th} press of button is

$$N_D = N_E!/(N_E - N_B)!\, N_B! \cdots (1)$$

These various numbers of decisions computed in accordance with Equation (1) are displayed in Table 1.

Table 1: Number of Decisions for Correct Vote Assignment.

n^{th} Button press (hex)	Maximum Number of Decisions for Accurate Vote Registration
01	7
02	21
03	35
04	35
05	21
06	7
07	1

The decisions for the various n^{th} button presses are given in Tables 2, 3, 4, 5, 6, 7, and 8. In these tables a shepherd stroke (√) in a column indicates that the election to the office indicated in that column is being held while a cross (✕) indicates that the election is not among those being organized. A 'd' indicates a 'don't–care' condition. Accordingly case 1 of the fourth button press (BUTTONCOUNT = 04h) in Table 5 can be interpreted as:

> ***If the Presidential (P), Governorship (G), Chairmanship (C), elections in this order are being held and if the Senate (S) election is also being held, then the fourth press of the vote button should be recorded for the office of the Senate (S) of the party of the voter's choice irrespective of whether the Representative (R), Assembly (A), or Council (K) elections are being held or not.***

Each of the unique entries in Tables 2-8 can be developed into a subroutine which will be used for decision making. As Table 1 shows however, this would entail the writing of a total of 127 subroutines for this decision making arrangement.

Table 2: First Button Press
(BUTTONCOUNT = 01h).

Case	P	G	C	S	R	A	K	Office to Process
1.	✓	d	d	d	d	d	d	P
2.	×	✓	d	d	d	d	d	G
3.	×	×	✓	d	d	d	d	C
4.	×	×	×	✓	d	d	d	S
5.	×	×	×	×	✓	d	d	R
6.	×	×	×	×	×	✓	d	A
7.	×	×	×	×	×	×	✓	K

Table 3: Second Button Press
(BUTTONCOUNT = 02h).

Case	P	G	C	S	R	A	K	Office to Process
1.	✓	✓	d	d	d	d	d	G
2.	✓	×	✓	d	d	d	d	C
3.	✓	×	×	✓	d	d	d	S
4.	✓	×	×	×	✓	d	d	R
5.	✓	×	×	×	×	✓	d	A
6.	✓	×	×	×	×	×	✓	K
7.	×	✓	✓	d	d	d	d	C
8.	×	✓	×	✓	d	d	d	S
9.	×	✓	×	×	✓	d	d	R
10.	×	✓	×	×	×	✓	d	A
11.	×	✓	×	×	×	×	✓	K
12.	×	×	✓	✓	d	d	d	S
13.	×	×	✓	×	✓	d	d	R
14.	×	×	✓	×	×	✓	d	A
15.	×	×	✓	×	×	×	✓	K
16.	×	×	×	✓	✓	d	d	R
17.	×	×	×	✓	×	✓	d	A
18.	×	×	×	✓	×	×	✓	K
19.	×	×	×	×	✓	✓	d	A
20.	×	×	×	×	✓	×	✓	K
21.	×	×	×	×	×	✓	✓	K

Table 4: Third Button Press
(BUTTONCOUNT = 03h).

Case	P	G	C	S	R	A	K	Office to Process
1.	✓	✓	✓	d	d	d	d	C
2.	✓	✓	×	✓	d	d	d	S
3.	✓	✓	×	×	✓	d	d	R
4.	✓	✓	×	×	×	✓	d	A
5.	✓	✓	×	×	×	×	✓	K
6.	✓	×	✓	✓	d	d	d	S
7.	✓	×	✓	×	✓	d	d	R
8.	✓	×	✓	×	×	✓	d	A
9.	✓	×	✓	×	×	×	✓	K
10.	✓	×	×	✓	✓	d	d	R
11.	✓	×	×	✓	×	✓	d	A
12.	✓	×	×	✓	×	×	✓	K
13.	✓	×	×	×	✓	✓	d	A
14.	✓	×	×	×	✓	×	✓	K
15.	✓	×	×	×	×	✓	✓	K
16.	×	✓	✓	✓	d	d	d	S
17.	×	✓	✓	×	✓	d	d	R
18.	×	✓	✓	×	×	✓	d	A
19.	×	✓	✓	×	×	×	✓	K
20.	×	✓	×	✓	✓	d	d	R
21.	×	✓	×	✓	×	✓	d	A
22.	×	✓	×	✓	×	×	✓	K
23.	×	✓	×	×	✓	✓	d	A
24.	×	✓	×	×	✓	×	✓	K
25.	×	✓	×	×	×	✓	✓	K
26.	×	×	✓	✓	✓	d	d	R
27.	×	×	✓	✓	×	✓	d	A
28.	×	×	✓	✓	×	×	✓	K
29.	×	×	✓	×	✓	✓	d	A
30.	×	×	✓	×	✓	×	✓	K
31.	×	×	✓	×	×	✓	✓	K
32.	×	×	×	✓	✓	✓	d	A
33.	×	×	×	✓	✓	×	✓	K
34.	×	×	×	✓	×	✓	✓	K
35.	×	×	×	×	✓	✓	✓	K

A more economical code development can be obtained if we carefully observe the frequency of the occurrence of the "office to process" in Tables 2-8.

Let us consider the case of the fourth button press (BUTTONCOUNT = 04h) in Table 5 again. In this table there is one (1) occurrence of Senate (S), 4 occurrences of Representatives (R), 10 occurrences of Assembly (A) and 20 occurrences of Council (K). If we deal with the cases of Senate, Representatives, and Assembly then any other case must fall into the category of Council, if indeed the Council election is being organized.

CODING, RESULTS AND DISCUSSIONS

The coding for the decision making arrangement contained in Tables 2-8 was done in assembly language using the instruction set of the microcontroller PIC18F2685 [7]. The code was edited in the MPLAB IDE (Integrated Development Environment) version 7.50 [8] and software simulation was carried out using the embedded MPLAB SIM after the edited code had been assembled [9].

Table 5: Fourth Button Press
(BUTTONCOUNT = 04h).

Case	P	G	C	S	R	A	K	Office to Process
1.	✓	✓	✓	✓	d	d	d	S
2.	✓	✓	✓	×	✓	d	d	R
3.	✓	✓	✓	×	×	✓	d	A
4.	✓	✓	✓	×	×	×	✓	K
5.	✓	✓	×	✓	✓	d	d	R
6.	✓	✓	×	✓	×	✓	×	A
7.	✓	✓	×	✓	×	×	✓	K
8.	✓	✓	×	×	✓	✓	d	A
9.	✓	✓	×	×	✓	×	✓	K
10.	✓	✓	×	×	×	✓	✓	K
11.	✓	×	✓	✓	✓	d	d	R
12.	✓	×	✓	✓	×	✓	d	A
13.	✓	×	✓	✓	×	×	✓	K
14.	✓	×	✓	×	✓	✓	d	A
15.	✓	×	✓	×	✓	×	✓	K
16.	✓	×	✓	×	×	✓	✓	K
17.	✓	×	×	✓	✓	✓	d	A
18.	✓	×	×	✓	✓	×	✓	K
19.	✓	×	×	✓	×	✓	✓	K
20.	✓	×	×	×	✓	✓	✓	K
21.	×	✓	✓	✓	✓	d	d	R
22.	×	✓	✓	✓	×	✓	d	A
23.	×	✓	✓	✓	×	×	✓	K
24.	×	✓	✓	×	✓	✓	d	A
25.	×	✓	✓	×	✓	×	✓	K
26.	×	✓	✓	×	×	✓	✓	K
27.	×	✓	×	✓	✓	✓	d	A
28.	×	✓	×	✓	✓	×	✓	K
29.	×	✓	×	✓	×	✓	✓	K
30.	×	✓	×	×	✓	✓	✓	K
31.	×	×	✓	✓	✓	✓	d	A
32.	×	×	✓	✓	✓	×	✓	K
33.	×	×	✓	✓	×	✓	✓	K
34.	×	×	✓	×	✓	✓	✓	K
35.	×	×	×	✓	✓	✓	✓	K

Table 6: Fifth Button Press
(BUTTONCOUNT = 05h).

Case	P	G	C	S	R	A	K	Office to Process
1.	✓	✓	✓	✓	✓	d	d	R
2.	✓	✓	✓	✓	×	✓	d	A
3.	✓	✓	✓	✓	×	×	✓	K
4.	✓	✓	✓	×	✓	✓	d	A
5.	✓	✓	✓	×	✓	×	✓	K
6.	✓	✓	✓	×	×	✓	✓	K
7.	✓	✓	×	✓	✓	✓	d	A
8.	✓	✓	×	✓	✓	×	✓	K
9.	✓	✓	×	✓	×	✓	✓	K
10.	✓	✓	×	×	✓	✓	✓	K
11.	✓	×	✓	✓	✓	✓	d	A
12.	✓	×	✓	✓	✓	×	✓	K
13.	✓	×	×	✓	✓	✓	✓	K
14.	×	✓	✓	✓	✓	✓	d	K
15.	×	✓	✓	✓	✓	×	✓	K
16.	×	✓	✓	✓	×	✓	✓	K
17.	×	✓	✓	×	✓	✓	✓	K
18.	×	✓	×	✓	✓	✓	✓	K
19.	×	×	✓	✓	✓	✓	✓	K
20.	✓	×	✓	×	✓	✓	✓	K
21.	✓	×	✓	✓	×	✓	✓	K

Table 7: Sixth Button Press
(BUTTONCOUNT = 06h).

Case	P	G	C	S	R	A	K	Office to Process
1.	✓	✓	✓	✓	✓	✓	d	A
2.	✓	✓	✓	✓	✓	×	✓	K
3.	✓	✓	✓	✓	×	✓	✓	K
4.	✓	✓	✓	×	✓	✓	✓	K
5.	✓	✓	×	✓	✓	✓	✓	K
6.	✓	×	✓	✓	✓	✓	✓	K
7.	×	✓	✓	✓	✓	✓	✓	K

The code here is part of a larger firmware for the electronic voting machine. The entire firmware was burned into the program memory of the microcontroller and after insertion into the machine circuit (Figure 2) the code was hardware debugged using the MPLAB ICD2 [10]. The picture of the complete electronic voting machine is shown in Figure 3. Typical collation of results with the algorithm integrated into the EVM is captured by the screen shot shown in Figure 4.

Table 8: Seventh Button Press
(BUTTONCOUNT = 07h).

Case	P	G	C	S	R	A	K	Office to Process
1.	✓	✓	✓	✓	✓	✓	✓	K

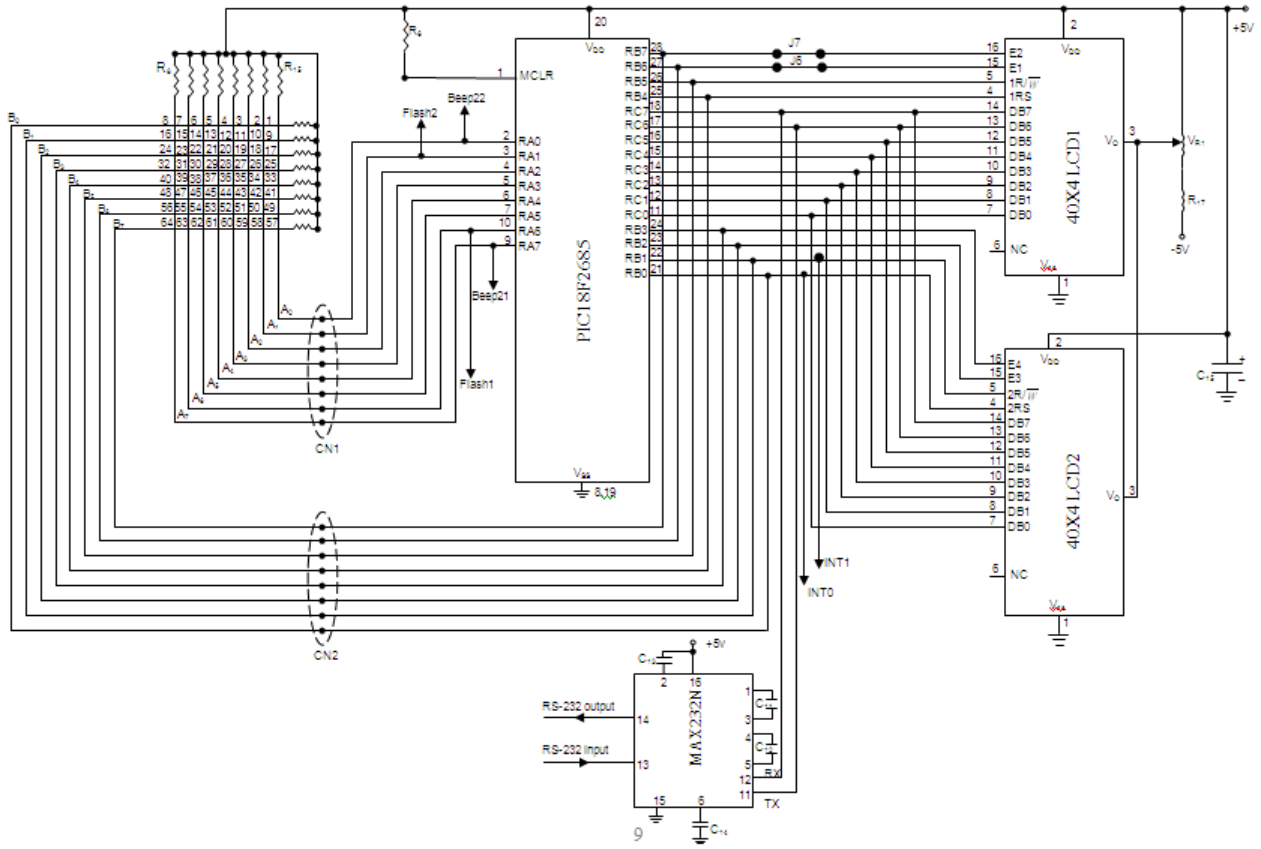

Figure 2: Schematic Diagram of the Voting Circuit.

Figure 3: Electronic Voting Machine showing Party Buttons and LCD Display Modules.

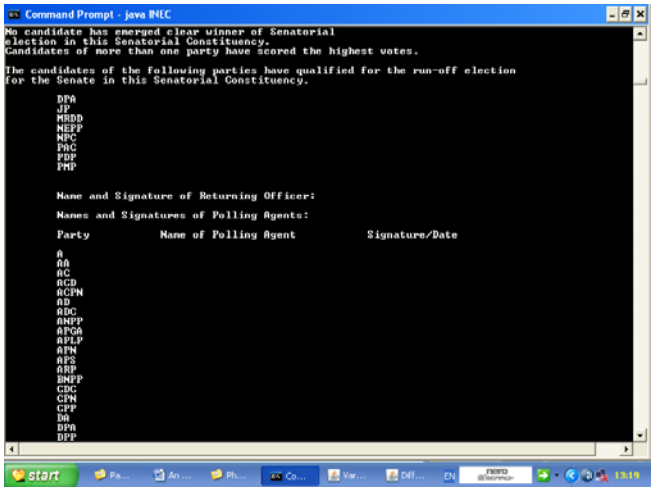

Figure 4: Concluding Section of one of Multi-Party Multiple Election Results.

CONCLUSION

The number of multiple simultaneous elections considered in this work is seven but fewer or more can be used depending on the requirements of the particular election environment as long as the program memory of the microcontroller is adequate. Since results of elections should be reproducible and, therefore, should be stored in non-volatile memory such as EEPROM, the size of this memory should be adequate for the number of political parties under considerations. A prototype of the Electronic Voting Machine shown in Figure 3 has actually been built. The machine was exhibited at the third Nigerian Universities Research and Development fair (NURESDEF) held in Lagos in March 2008. It has also been demonstrated to the Electoral Reform Committee (ERC) set up by the President of the Federal Republic of Nigeria.

REFERENCES

1. Election Commission of India. 2010. "Frequently Asked Questions – Electronic Voting Machines [EVMS]". http://www.eci.gov.in/faq/elecvtmach.htm

2. Beroggi, G.E.G. 2008. "Secure and Easy Internet Voting". *Computer*. IEEE Computer Society Publications. 52-56.

3. He, M., R. Almeida, and E. Gissoni. 2002. "National Semiconductor and Unisys Equip Brazil with New Voting Machines for Fast and Accurate Election Results in the Fall". http://www.national.com/news/item/o,1735,757,00.html

4. Enokela, J.A. and C.C. Osuagwu. 2010. "Architecture of a Low Cost Electronic Voting Machine Suitable for Deployment in Developing Countries". *Pacific Journal of Science and Technology*. 11(1):649-657.

5. Federal Republic of Nigeria. 2010. "Electoral Act 2010, Section 111". Government Printers, Federal Republic of Nigeria: Lagos, Nigeria.

6. Federal Republic of Nigeria. 2000. "The Constitution of the Federal Republic of Nigeria, 1999: Sections 48, 49, 90, 130, and 176". Government Printers, Federal Republic of Nigeria: Lagos, Nigeria.

7. Microchip Technology Inc., "PIC18F2685 Data Sheet". MTI: Chandler, AZ. http://www.microchip.com.

8. Microchip Technology, Inc. 2004. "MPLAB IDE User's Guide", MTI: Chandler, AZ. http://www.microchip.com.

9. Microchip Technology, Inc. 2005. "MPASM Assembler, MPLINK Object Linker, MPLIB Object Librarian User's Guide". MTI: Chandler, AZ. http://www.microchip.com.

10. Microchip Technology, Inc. 2006. "MPLAB ICD 2 User's Guide". MTI: Chandler, AZ. http://www.microchip.com.

ABOUT THE AUTHORS

Engr. Dr. Jonathan A. Enokela, is a Lecturer in the Department of Electrical/Electronic Engineering at the Federal University of Agriculture, Makurdi, Nigeria. He has taught various aspects of analog and digital system design to both undergraduate and postgraduate students. He is a registered engineer with the Council for the Regulation of Engineering in Nigeria (COREN) and has a wide range of practical experiences. His research interests include embedded systems design and applications.

Engr. Prof. Charles C. Osuagwu, is a Professor in the Department of Electronic Engineering at the University of Nigeria, Nsukka where he is also director of the Computer Communications Centre. His areas of interests include digital signal processing and software engineering.

SUGGESTED CITATION

Enokela, J.A. and C.C. Osuagwu. 2011. "An Algorithm for the Conduct of Multiple Simultaneous Multi-Party Elections Using a Microcontroller". *Pacific Journal of Science and Technology*. 12(2):253-259.

Pacific Journal of Science and Technology