Development of a Microcontroller Based Automatic Combinational Lock System.

Comfort O. Bolajoko, M.Sc.* **and Adewale M. Folorunso, B.Sc.**

Department of Electronic and Computer Engineering, Lagos State University, Lagos, Nigeria.

E-mail: comfortobolajoko@yahoo.com*
walebb@yahoo.com

ABSTRACT

The work presented in this paper discusses the microcontroller based automatic combinational lock. The need for automatic combinational lock has been on the increase in recent times. The system designed and constructed incorporates the use of a microcontroller.

A microcontroller based automatic combinational lock system is a lock system that can be used for numerous applications in which access to a particular target is to be restricted to a limited number of persons. It involves using four-digit decimal numbers as security access codes to open any door for which it is configured. It is unique in the sense that it is controlled by software, which can be modified anytime the system demands a change by reprogramming the default codes to the desired codes.

(Keywords: automatic combinational lock, microcontroller, software, access codes, system)

INTRODUCTION

A combinational lock system is a type of lock system in which a sequence of numbers is used to open the lock. This sequence of numbers may be entered using a keypad. This could be achieved using digital logic devices, microprocessor based system, or microcontroller. Of all these, the use of microcontroller and the trends towards it to achieve this objective will be discussed in the course of this paper.

A microcontroller is a single chip computer that contains many of the same items that a desktop computer has such as Microprocessor, Memory etc., but does not include any human interface devices like monitor, keyboard or mouse. "Micro" suggests that the device is small and "Controller" suggests that the device might be used to control objects, processes or events.

LITERATURE REVIEW AND THEORY

The work presented in this paper is a design and construction of a microcontroller based automatic combinational lock system. The functionality is implemented in the software as an access control system. The microcontroller is used to write in data values from the input device and interact with the outside world. The system accepts the codes supplied from the keypad placed at the door or entrance; compare the code with codes already stored in the microcontroller registers. If the code matches, then triggers access by opening the lock otherwise denies access.

The automatic combinational lock system comprises a power supply unit, a keypad input unit, a controller unit, and lock unit.

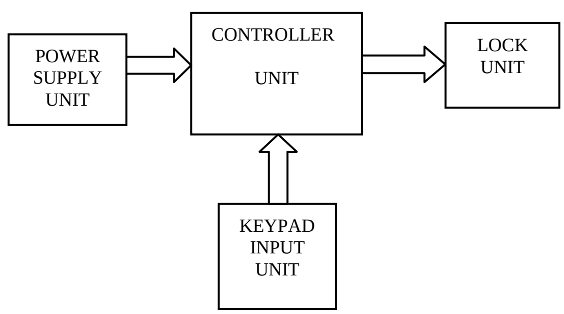

Figure 1: Block Diagram of the Combinational Lock System

The keypad input unit provides the input signal to the system. It is made up of switches arranged in rows and columns which sends codes to the input port of the microcontroller chip when they are pressed. The software causes the microcontroller to be checking the input ports for the keypad status information. The valid or correct code supplied a signal that triggers the lock unit. The code can be changed any time after entering the correct code.

MATERIALS AND METHODS

The system design is divided broadly into two parts; hardware and software.

Hardware Design

The Hardware Design is subdivided into the under listed sub-unit for easy design, analysis and integration.

- Power Supply unit
- Keypad unit
- Controller unit
- Lock unit

Power Supply Unit

For a proper function of any microcontroller, it is necessary to provide a stable source of supply according to technical specifications by the manufacturer of PIC microcontroller, supply voltage should move between 2.0V to 6.0V in all versions.

The components used to make up the power supply include: Transformer which steps down the input voltage, Rectifier which converts the ac voltage to a pulsating dc voltage, Capacitor which removes the ripples in the pulsating output voltage. The value of the capacitor can be determined using the following equation:

$$V_{dc} = V_m - (4.7x\ 10^{-3} I_{dc}) / C \qquad \textbf{(1)}$$

where V_m is the peak-rectified voltage in volts
I_{dc} is the load current in (mA)
V_{dc} is the dc voltage
C is the filter capacitor in microfarad (µF)

As it was stated earlier that a regulated power supply is required for the microcontroller to function properly, a voltage regulator is used to produce a stabilized output voltage.

Keypad Unit

A keypad is a common primary input device for embedded microcontrollers.

A Keypad actually consists of push buttons or press-to-make switches arranged in a column-by -row matrix (In this case four by three matrix) as shown in the diagram below. One terminal of each switch is connected to microcontroller via a pull-up resistor, and the other terminal is connected to ground. When the switch is not pressed, it sends logic '1' (high signal) to the microcontroller else it sends logic '0' (low signal). Figure 2 shows the keypad unit. It serves as an interface between man (user) and the controller unit.

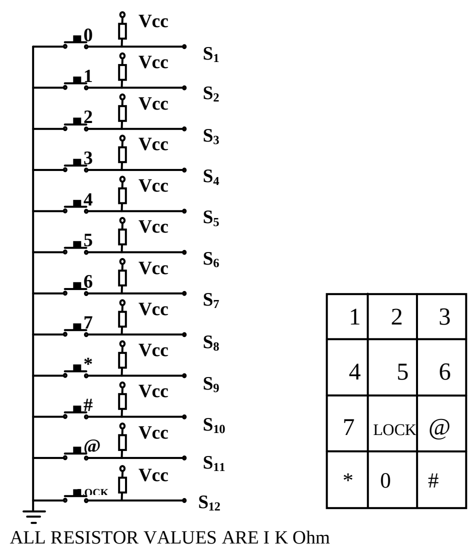

Figure 2: Keypad Circuit Diagram and Layout

The keypad contains four lines, each consisting of three characters. The keypad wires are connected to the controller by the Ports (PORTA3, A4, and PORTB0 – B7) via pull-resistors each of 1K ohm resistance. The '#' key serves as enter, while the '*' key enables one to change the codes of the lock. The '#' key can also serve as key 9 for the input code and the '*' key can also serve as key 8 for the input code. 'LOCK' key on the keypad is used to roll the electric motor anti-clock-wisely back to its lock state. The '@' key is the reset key which is used to reset the microcontroller to the default code in case the present code is forgotten.

Controller Unit

Figure 3 shows the circuit diagram of the controller unit. It consists of the microcontroller and the oscillator circuit. Pins RA3, RA4 and pins RB0 to RB7 are configured as input ports and they are connected to the keypad to serve as the input to the microcontroller. Pin RA1 is configured as the output port which is connected to the code change indicator LED. Pin RA2 is also configured as the output port and connected to the relay that triggers the lock. Pin RA0 is configured as the output port and is connected to the sound buzzer. The buzzer beeps at power - up and anytime a key is pressed on the keypad.

The oscillator circuit is a crystal oscillator connected to pins 15 and 16 (OSC IN & OSC OUT) of the microcontroller. This generates pulses to clock the microcontroller for program execution.

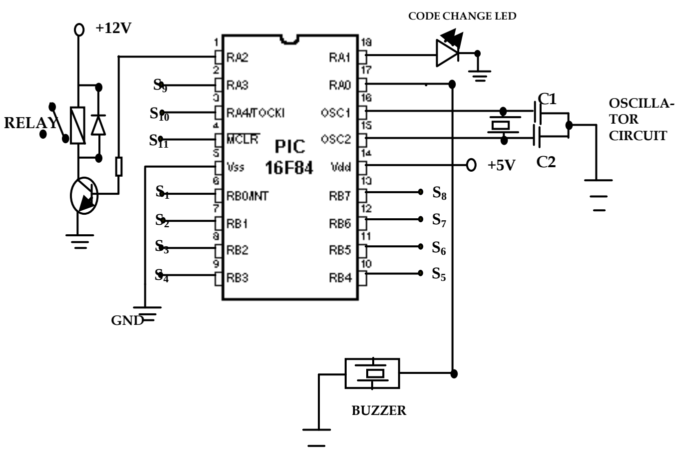

Figure 3: Controller.

The PIC16F84 microcontroller was used in this paper. The architecture of PIC series of microcontroller is completely different from other microcontrollers. In this design, the data bus and the address bus are separate which makes a greater flow of data possible through the CPU, hence a greater speed of work. The PIC16F84 contains the Program Memory, RAM Memory, and Ports.

Instruction Set

Microcontrollers are needed to be programmed in order to put them into operation to perform specific task. Since PIC16F84 is a member of the RISC family of microcontrollers, it makes use of a reduced set of 35 instructions. Each instruction takes one clock cycle to complete. Four oscillator clocks make up one instruction cycle. If we are using 4MHz crystal oscillator, then each instruction will take:

$$4 \times (1/4\text{MHz}) = 1\mu\text{secs} \quad \textbf{(2)}$$

to complete. These instructions can be categorized as follows:

- Data Transfer
- Arithmetic Operations Shift and Rotate
- Status testing
- Flow Control Instruction
- Bit Operations

The Lock Unit

Figure 4 depicts the circuit diagram of the lock unit. It consists of three relays and an electric motor. The relays act as switches to roll the electric motor in clockwise and anti-clockwise directions to unlock and lock the door respectively. The relay consists of a coil of insulated wire on a metal core, and a metal armature with one or more contact. When a supply voltage is delivered to the coil, current flows and a magnetic field will be produced that moves the armature to close one set of the contact and/or opens another set. When power is removed from the relay, the magnetic flux in the coil collapses and produces a fairly high voltage in the opposite direction. This voltage can damage the driver transistor and thus a reverse bias diode is connected across the coil to shut out the spike when it occurs.

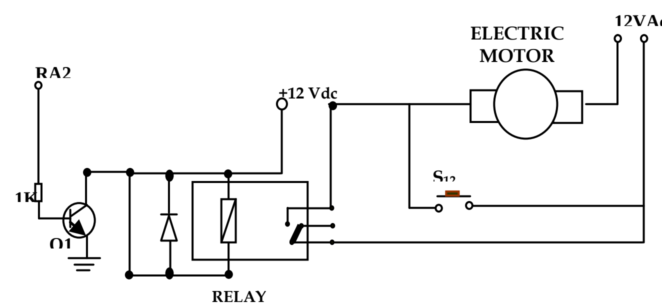

Figure 4: Lock Unit Circuit Diagram.

Since microcontroller cannot provide sufficient current to the coil (approx. +100mA is required; microcontroller pin can provide up to 25mA), a transistor is used for adjustment purposes, its collector circuit contains the relay coil. When a logical one is delivered to transistor base, transistor activates the relay, which then using its contacts, connects other elements in the circuit. Purpose of the resistor at the transistor base is to keep a logical zero on base to prevent the relay from activating by mistake. This ensures that only a clean logical one on RA2 activates the relay.

A 12V AC electric motor is used and it rolls clock wisely to unlock the door anytime the correct keys are pressed on the keypad. It then rolls in an anti-clockwise direction to initial position when the lock key on the keypad is pressed.

A small metal plate is mounted on the rotor of the electric motor. This makes the locking and the unlocking states possible.

The relays and the electric motor take in their input voltage from the 12V output of the Transformer.

THE SOFTWARE DESIGN

The Software design involved the system models design, Algorithm, Flowchart, Assembly language code, linking of various part of the program, final compiling of program and final program of the microcontroller.

The software is decomposed into modules so that each module can be individually tested as a unit and debugged before the modules are integrated and tested as a whole in order to ensure that the software design meets its specification. The program for this system was written in Assembly language for speed optimization.

When writing an assembly program into a microcontroller, the assembly code must first be converted into the machine code understood by the microcontroller (i.e. hex file): using an assembler program (in this case MPLAB and PIC simulator IDE). The hex file is loaded into the microcontroller via a parallel port interface using a PIC programmer (in this case Piccall PIC programmer by microchip).

The program modules were modeled into six segments namely:

a) Main program
b) Delay subroutine
c) Beep subroutine
d) Key scan subroutine
e) Correct code subroutine
f) Code change subroutine

The software was designed using the following steps:

1) Algorithm
2) Flowchart

<u>Algorithm for Implementing the Lock System</u>

The algorithm used to implement the program for the system is shown in the flowchart in Figure 6 and described as follows:

1. Start

2. Initialize: Store the default codes into registers code1, code2, code3 and code4

3. Configure the input and output port of the microcontroller from RP0 of the status register.

4. Initialize counter for scanning.

5. (a) Scan the keys pressed on the keypad
(b) Sound the buzzer for each key pressed by sending a signal to the buzzer.
(c) Store the values of the key pressed in appropriate registers.

6. (a) Compare the key pressed with the default codes
(b) If the code matches, and an '# ' is pressed then send a signal to the Relay output. Wait for 3seconds and clear the output pin.
(c) Else go to step 6.

7. (a) If the code matches and ' * ' is pressed then send a high to the code change output. Else go to step 4.
(b) Scan the keypad for new codes and over write the default codes, if the new codes are entered twice and ' # ' is pressed as enter key.
(c) Else retain the default codes.

8. Stop

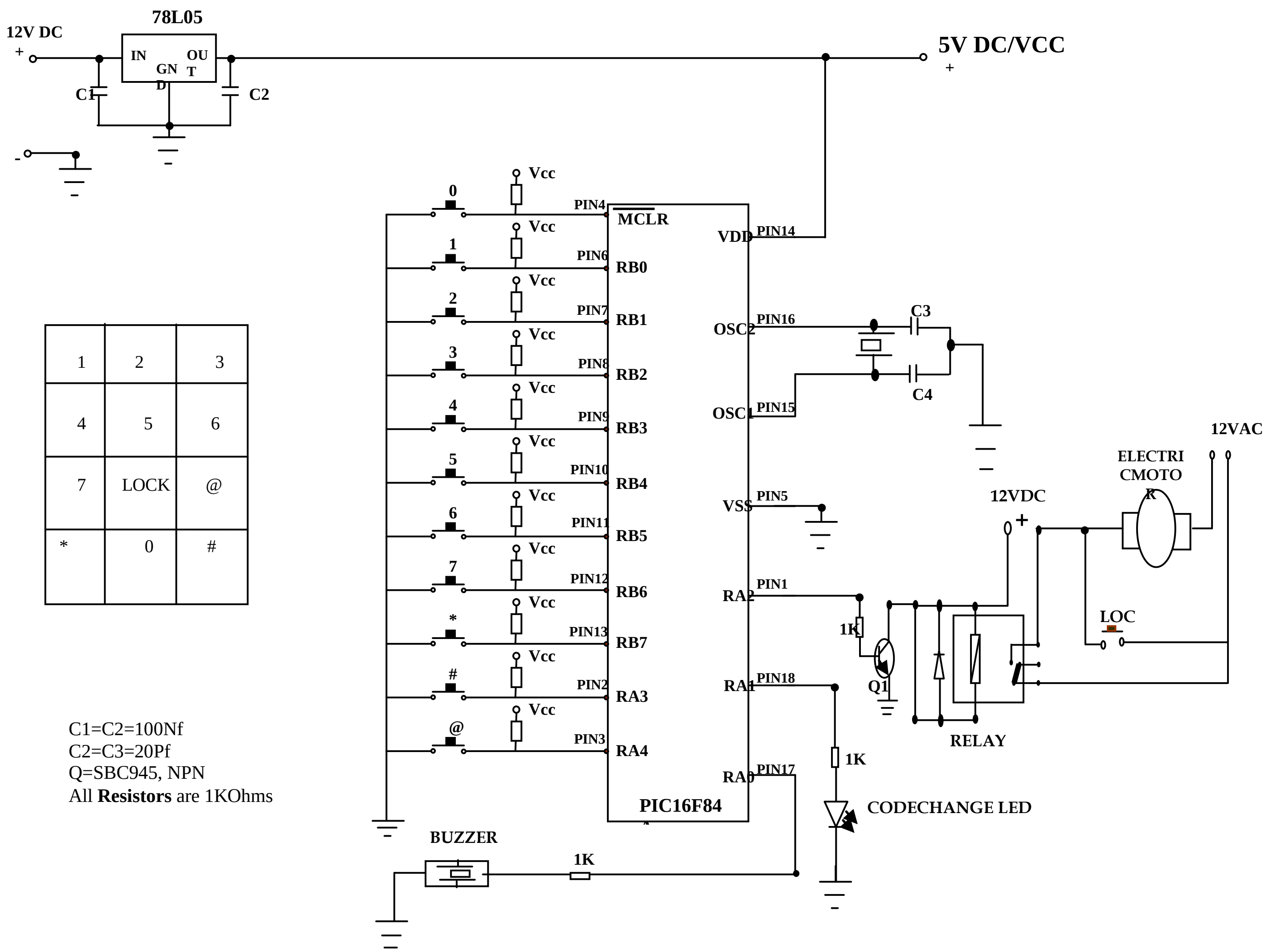

Figure 5: Complete Circuit Diagram of the System.

TESTING AND RESULTS

The testing of the entire circuit was carried out in stages.

After the proper testing of the peripherals and found to be working perfectly, the entire circuit was tested.

Series of programs (software) were written and simulated using PIC simulator IDE and MPLAB before the working program was finally achieved and then transferred to the microcontroller chip using Piccall software/hardware parallel port interface PIC programmer. The circuit worked perfectly as intended.

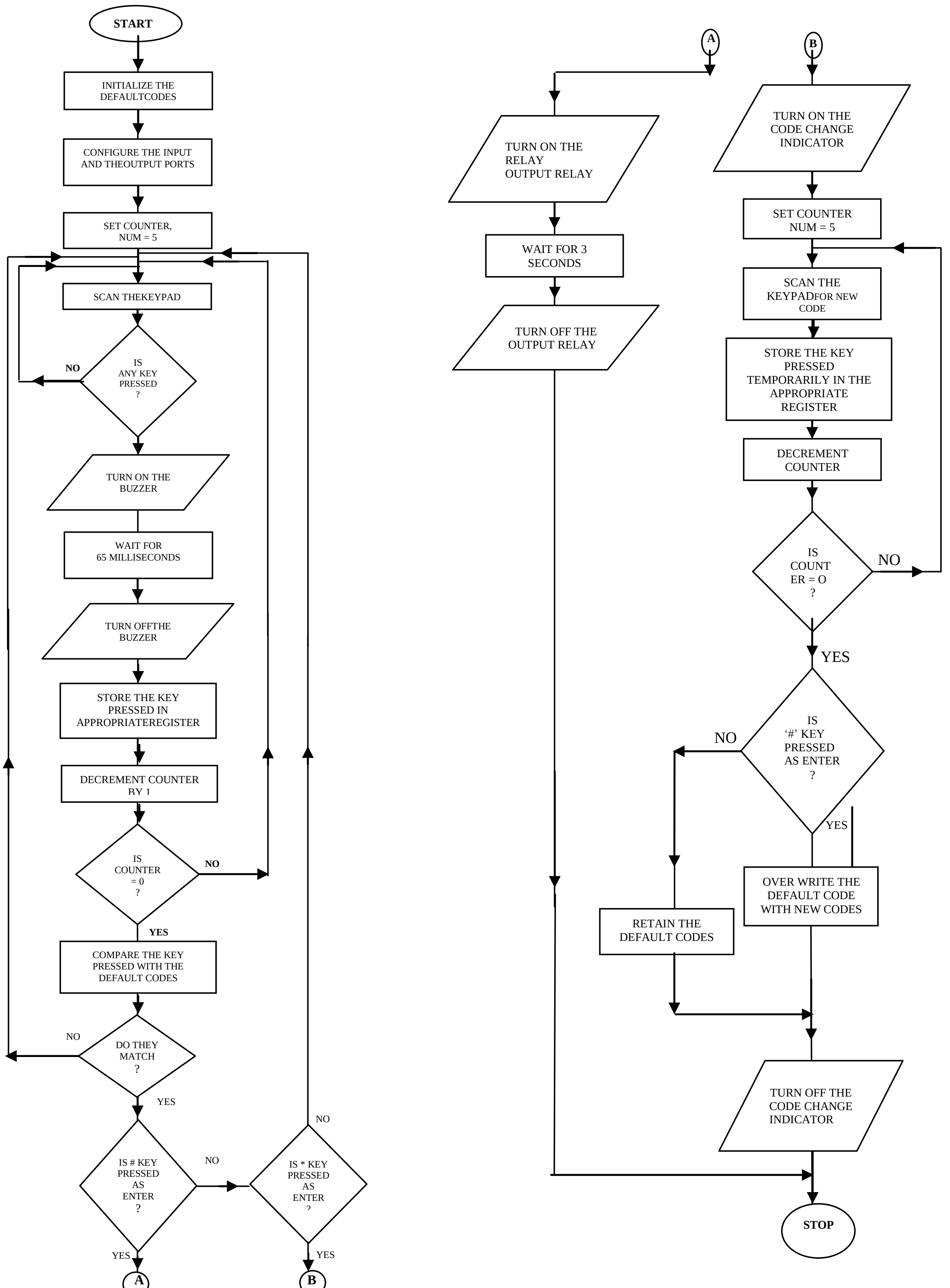

Figure 10: Flow Chart.

Table 1: Result of Voltage Levels of the Input Ports of the Microcontroller when a Key is Pressed and when No Key is Pressed on the Keypad.

Port	Voltage Level when No Key is pressed	Voltage Level when a key is pressed on the keypad
RA3	+4.5V	+0V
RA4	+4.5V	+0V
RB1	+4.5V	+0V
RB2	+4.5V	+0V
RB3	+4.5V	+0V
RB4	+4.5V	+0V
RB5	+4.5V	+0V
RB6	+4.5V	+0V
RB7	+4.5V	+0V

Table 2: Result of Voltage Levels of the Output Ports of the Microcontroller when a Key is Pressed and when No Key is Pressed on the Keypad.

Port	Voltage Level when No Key is pressed	Voltage Level when a key is pressed on the keypad
RA0	+0V	+4.5V
RA1	+0V	+4.5V
RB2	+0V	+4.5V

DISCUSSION OF RESULTS

The result obtained shows that each switch on the keypad send +4.5V to the microcontroller input ports (PORT RA3, RA4, RB0 – RB7) when the switch is not pressed. At press of each switch, a +0V is obtained at the corresponding microcontroller input port.

A +4.5V is obtained at pin RA0 of the microcontroller to turn ON the green LED and sound the buzzer at power-up and at the press of each button on the keypad; otherwise a 0V is obtained at this output pin (RA0).

The controller unit is connected such that the input voltage from the keypad unit is +4.5V. When no key is pressed on the keypad; the microcontroller sends a 0V to its output ports (PORT RA0, RA1 and RA2). When the keypad is pressed; the input voltage on the microcontroller input port voltage changes to zero and there is +4.5V at the pin RA0 output of the microcontroller to sound the buzzer.

A +4.5V is obtained at pin RA2 of the microcontroller to turn ON the red LED and to trigger the relay in the lock unit when the correct code is inputted and '#' was pressed as enter key on the keypad otherwise a 0V was obtained at this output pin (RA2).

A +4.5V was obtained at the pin RA1 of the microcontroller to turn On the code change indicator LED (yellow) when the correct codes were inputted and '*' was pressed on the keypad otherwise a 0V was obtained at this output pin.
When the correct codes were pressed, +4.5V was obtained at pin RA2 output pin of the microcontroller, which triggered the relay controlling the AC motor which rolled clock wisely to unlock the door. When the door was closed and lock button on the keypad was pressed, the electric motor rolled anti-clockwise to lock the door.

CONCLUSION

With the implementation of our security system in this case microcontroller based automatic combinational lock system; it is possible to secure the building or target from unauthorized people.

With all afore said above, it can be inferred that microcontroller based automatic combinational lock system still need to make more use of the controller than it is presently used for enhancement performance.

Also, it was noted that there is hope for interactive programming, which is used in this application. A design of this particular project modified by incorporating an electromagnetic lock/magnetic lock instead of the electric motor which was used in the locking and unlocking state of the system is desirable and effective.

REFERENCES

1. Tait, D. 2013. "Topic and Simple demo Programs". http://www.man.ac.uk

2. El Cheapo PIC. 2013. "Microcontroller Programmer". http://www.myke.com/ElCheapo.html

3. Jap.hu. 2013. "Electronics Projects with Microcontroller". http://www.jap.hu/electronics.html

4. Quasare. 2013. "Introduction to PIC Programming". http://www.quasare/eletronics.com

5. MPLAB. 2013. "MPLAB IDE v5.1 and v7.20". http://www.microchip.com

6. Tait, D. 2013. "My Microcontroller based Projects". http://www.ke4ngv.com

7. Jakab, P. 2013. "Peter Jakab's Electronic Project". http://jap.hu/electronic/combination_lock.html

8. MPLAB. 2013. "PIC16F84A Datasheet" http://ww1.microchip.com/downloads/en/devicedoc/35007b.pdf

9. Boondog.com. 2013. "PIC16F84 Fundamentals". http://www.boondog.com

10. OSHONSOFT. 2013. "PIC Simulator IDE with Basic Compiler, Assembler and De-assembler". http://www.oshonsoft.com

11. UCT. 2013. "Programming PIC16f84". http://web.uct.ac.za/depts/ebecourses/eee350w/PICARCHIVE/picprog.html

12. Circuits-Dias.com. 2013. "Programmable Digital Code". http://www.electronic.circuits-dias.com

13. UBasics.com. 2013. "Programming PIC16F84 Micro-controllers". http://www.ubasics.com

14. Electronicdesign.com. 2013. "Simple Electronic Lock using Single Transistor Circuit". http://www.electronicdesign.com

ABOUT THE AUTHORS

Comfort O. Bolajoko, is an Assistant Lecturer at the Department of Electronic and Computer Engineering, Faculty of Engineering Lagos State University, Epe Campus. She holds a Master of Science (M.Sc.) in Electronic and Computer Engineering from Lagos State University. Her area of research is in communication.

Adewale M. Folorunso, is a graduate in Electronic and Computer Engineering, Faculty of Engineering, Lagos State University, Epe Campus. His research interests are in communication and networking.

SUGGESTED CITATION

Bolajoko, C.O. and A.M. Folorunso. 2013. "Development of a Microcontroller Based Automatic Combinational Lock System". *Pacific Journal of Science and Technology*. 14(2):234-241.

Pacific Journal of Science and Technology