

# Design of a Microprocessor Based Automatic Gate

O. Shoewu\* and O.T. Baruwa

Electronic and Computer Engineering Department, Lagos State University  
Epe, Nigeria

\*E-mail: [engrshoewu@yahoo.com](mailto:engrshoewu@yahoo.com)

## ABSTRACT

The work presented here outlines the development of a microprocessor based automatic gate. The inconveniences encountered in gate operations has called for an immense search for solutions. The microprocessor based automatic gate offers everything necessary to put an end to these inconveniences as it incorporates an intelligent device (microprocessor).

Specifically, the system described in this paper monitors two gates, the entrance and exit. The automatic gate senses any vehicle approaching it. It automatically opens, waits for a specified time, and closes after the time has elapsed. As soon as the gate closes, the system counts, registers, and displays the number of vehicles.

The system also serves as an automobile parking control unity by periodically checking the number of vehicles that have entered the area and computing the available space limit in the parking area. Once the available space limit is reached, the system triggers an alarm for a specified time and the entrance gate remains inaccessible until another vehicle comes out through the exit gate.

The automatic gate developed in this project is unique in that it is controlled by software, which can be modified any time the system demands a change.

(Keywords: automatic gate, microprocessor, automobile, traffic controllers, software, parking)

## INTRODUCTION

The need for automatic gates has been on the increase in recent times. The system described here incorporates the use of a microprocessor as a controller in achieving the aims of this project. It is no exaggeration to say that the microprocessor has revolutionized the electronics industry and has had a remarkable impact on many aspects of our lives (1999).

Almost all areas of technology have started taking advantage of the inexpensive computer control that microprocessors can provide. Some typical applications include: electronic games, CD players, automatic braking systems, industrial process controls, electronic measuring instruments, automobile emission controls, microwave ovens, traffic controllers, and a rapidly growing number of new products.

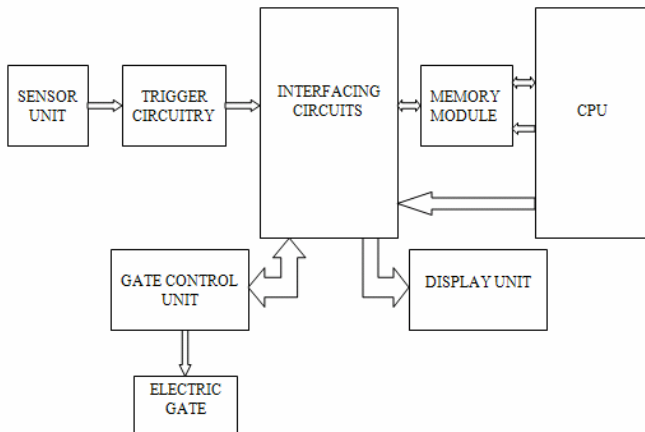
The automatic gate described here automates the entrances to parking lots of residential homes, organizations, automobile terminus, and public car parks. It uses a remote control convenience to avoid the stress of manually opening and closing the gate. The technology used eliminates gate monitoring and manning by human beings. The gate uses a state-of-the-art entry system. The gates have to perform gyrations – open, auto-reverse, stop, fully close and fully stop.

The automatic gate is not a security device and should not be construed as one. It provides convenient access and intelligent features that makes it distinct from all other gates which brings it so close to a security device.

## GENERAL OVERVIEW OF THE SYSTEM

The research work presented here is the design and development of a microprocessor based automatic gate. As a monitoring and control system, the microprocessor was used to read in data values from the input device and interact with the outside world.

The system senses, opens and closes the gate, counts, registers, and displays the number of vehicles crossing the gate (both entrance and exit) and triggers an alarm once the space limit is reached. Once triggered, the gate remains inaccessible until another vehicle leaves the park (1983). The automatic gate system comprises a sensor unit, a trigger circuitry, CPU module, memory module, display unit, gate control unit and the power supply unit as shown in the block diagram below.



**Figure 1:** Block Diagram of the System

The sensor provides an input signal to the system. It is an optical sensor which, when light rays are focused on it, has a low resistance and hence, causes the input to the trigger circuitry to be held "LOW". But, when a vehicle interrupts the beam, the resistance increases and reaches its dark resistance, thus, the input to the trigger circuitry is held "HIGH". The trigger circuitry serves as an Analog-to-Digital converter (ADC), which produces a HIGH signal when the beam is interrupted. The trigger circuitry sends a signal to the interface unit, which is made up of Programmable Input/Output Controller (PIO). The software causes the microprocessor to be check the input port of the interface unit for the sensor status information (the outputs of the trigger circuitry). A HIGH value causes the microprocessor to send a signal to the output port of the interface unit in order to activate the DC motor to control the gate (open and close). It equally sends a signal to the display unit for counting the number of vehicles. A LOW value will never activate the gate. False triggering is taken care of by the trigger circuitry. The power supply unit supplies the required DC voltage needed by the entire microcomputer system.

### HARDWARE AND SOFTWARE DESIGN CONSIDERATIONS

Certain specifications, parameters, and methods of implementation must be considered in system design and construction in order to give the expected result. The implementation of the design involves segmenting the overall system design into subsystems/modules/units, which are individually designed and tested before the integration of the various subsystems. The system design is divided into:

#### 1) HARDWARE DESIGN consisting of:

- Sensor Unit
- Trigger Circuit
- CPU Module
- Memory Module
- Display Unit
- Gate Control Unit
- Power Supply Unit

#### 2) SOFTWARE

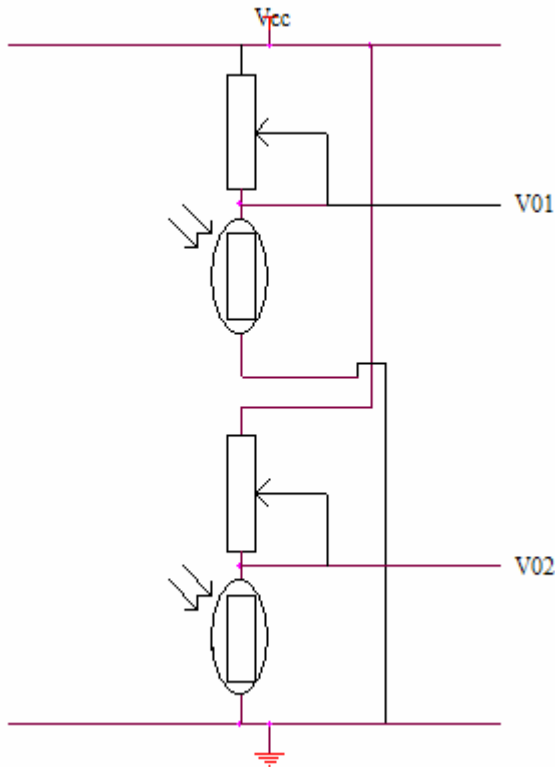
### SENSOR UNIT

This module makes use of an optical sensor, specifically a light dependent resistor (photo conductive cell), whose resistance changes with the intensity of light. The type used is ORP12 and it has a dark resistance of 10MΩ. The sensor unit is shown in Figure 2. When light rays are focused on the LDR, the resistance becomes very low (0-500Ω) but when the rays are interrupted, the resistance increases to its dark resistance. The variable resistor is used to vary the sensitivity of the LDR. It is otherwise called Dark Activated Sensor.

Two pairs of sensors (4 total) were used for the entire system; each pair for the entrance and exit gates and the outputs from the sensor units and is part of the trigger circuitry. The sensor unit is arranged in such a way that it consists of two pairs of LDRs to provide signals for the trigger circuitry whenever there is an obstruction through the entrance or exit gate. For the design, two conditions are considered: first, when light rays are focused on the ORP12, and secondly, when the rays are being interrupted.

When light rays of great intensity are focused on the ORP12, the output voltages, V01 and V02 are low (approximately 0V). When the light beams are interrupted, the output voltages increase to 5V approximately.

The circuit has the ability to detect only the passage of an automobile through the entrance and exit gates (1976). Each pair of sensors are separated by a reasonable distance such that the passage of a person or other moving objects cannot obstruct the sensor pair separation. If this happens, only one sensing unit is activated and is processed by the trigger circuitry so that there will be no triggering. Also, the height of the sensors is considered such that only the body of a vehicle can interrupt the light beams of the sensors and not the tires or its windows. To avoid false triggering, the two sensors must be interrupted at the same time.



**Figure 2:** Sensor Unit.

### TRIGGER CIRCUITRY

This is made up of a Schmitt-trigger, Quad two-input NAND gate (74LS132), to form a de-bounced switch circuit. It accepts the output of the two sensor units. Its function is to NAND the two inputs from the sensing unit, clip, and shape the pulse into square waves. It is configured in such a way that only when there is an output from the two sensing units will the trigger circuitry go HIGH, else it remains at LOW level.

Figures 3a and 3b shows the Schmitt-trigger and the waveform. The Schmitt-trigger accepts slow changing signals and produces an output that has oscillation-free transitions.

When traffic interrupts the light beams of the sensor units, the voltage at the output of the Schmitt-trigger goes HIGH. The circuit resets to a LOW voltage level when the interrupt is removed. The truth table of the Schmitt-trigger is shown below in Table 1.

### CPU MODULE

This module provides the system clock, reset, and access provision to address, data, and control buses architecture. Additional circuitries

are needed for the Z80 CPU to perform its functions. These circuitries, with their design, are discussed as follows.

### CLOCK CIRCUITRY

The Z80 requires a single-phase 0V to 5V clock signal. This can be generated with a quartz crystal and 74LS14 with some RC connections or with the use of a crystal oscillator which is more reliable. The crystal oscillator was used to implement this clock circuitry.

Because the typical high level output voltage of a TTL gate is only 3.3V or 3.4V and the minimum high level required at the Z80 clock input is 4.4V, a 330Ω pull up resistor is required.

In this design, a Z80 CPU was used, which has a clock cycle of 2.5MHz. Thus, a crystal oscillator of 5MHz was used and passed through a D flip-flop to give 2.5MHz (divide by 2), which is able to drive the Z80 clock rate. This is shown in Figure 4.

### RESET CIRCUITRY

This circuit functions to initialize the Z80 after power is supplied and also to revitalize it if a "HALT" occurs or after some other catastrophic event. When the Z80 is reset, it begins execution from memory location 0000H because the program counter is cleared to zero and forced to begin at location 0000H. It also disables interrupts and clears the interrupt enabled flip-flop.

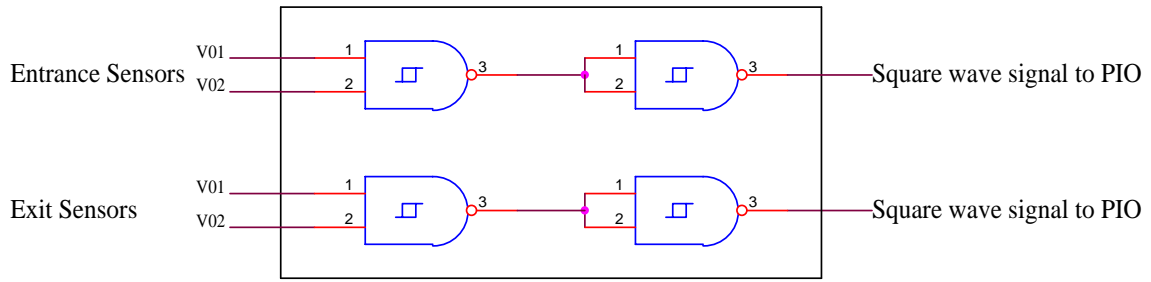
Figure 5 shows the reset circuit. The RC network connected to the reset input as shown makes the capacitor hold the reset pin low for several time constants when power is first applied. Because the reset signal obtained across the capacitor is an exponential rise and not a sharp square wave, a Schmitt-trigger 74LS14 is used to shape the pulse into square wave.

### MEMORY MODULE

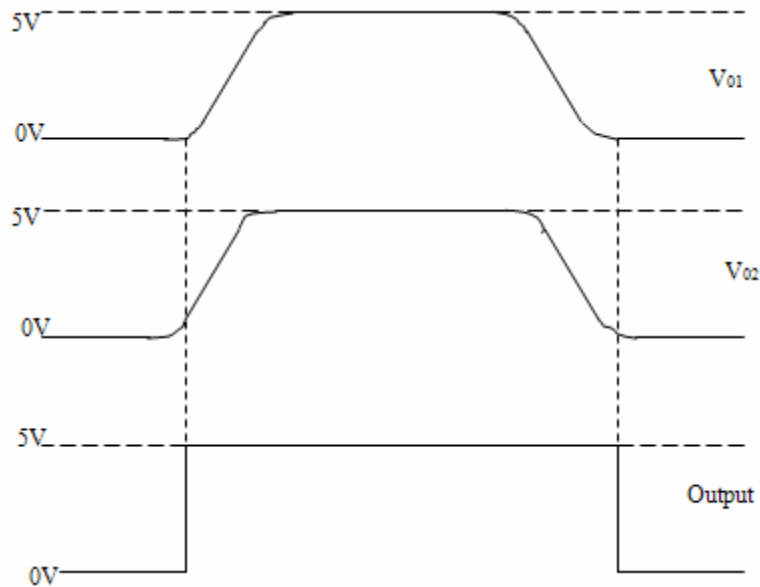
The memory devices interfaced to the CPU module consists of:

- a. 2K X 8 ROM (2716)
- b. 2K X 8 RAM (6116)

In order to interface a microprocessor to ROM/RAM chips, two address decoding techniques are usually employed.



**Figure 3a:** Schmitt-Trigger Quad two-input NAND gate (74LS132)



**Figure 3b:** Waveform Schmitt-Trigger

**Table 1:** Truth Table for Schmitt-Trigger

Inputs		Output
V01	V02	F
0	0	0
0	1	0
1	0	0
1	1	1

1. Linear select decoding technique
2. Fully decoding technique

In linear select decoding, each bit on the address bus can select a device. This is employed in very small systems with few devices. It does not require any decoding hardware but wastes a large amount of address space and sometimes causes bus conflict.

Fully decoding memory addressing requires the use of a decoder to select a memory device. In the hardware design implementation, the fully decoding technique was employed for memory decoding and input/output decoding for the PIOs. Address bus lines A0-A7 were used for the input/output mapping.

### ADDRESS DECODER

This indicates that a particular area of memory is being addressed or pointed to by the microprocessor. This can be realized by use of combinational circuits. For the implementation of this design, the 74LS138 decoder was used. The most useful features of this device are the multiple enable inputs. The decoder has an active low output that only becomes active when all of the enables are at their active levels.

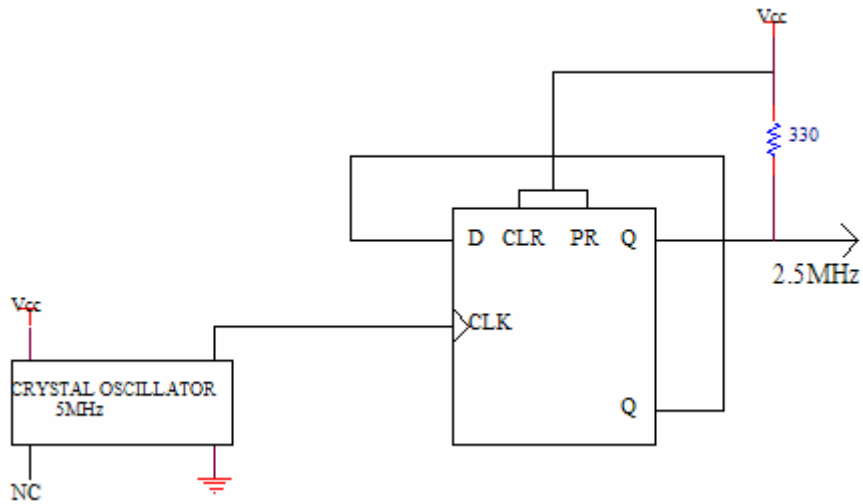


Figure 4: Clock Circuitry.

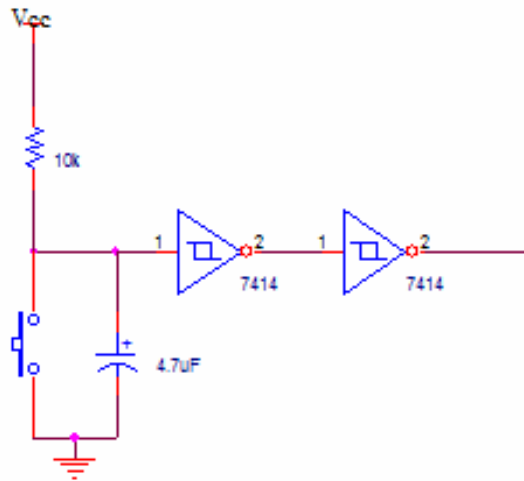


Figure 5: Reset Circuit.

Whenever this device is enabled, the 3-bit binary number present at the address inputs causes one of the output pins to become active.

## MEMORY AND I/O MAPPING

The memory mapping of the system design (Figure 6) is shown in Table 2. Interfacing both memory devices to the CPU does not require any additional circuitry to generate wait states. The reason being that both devices meet the memory read/write timing specification of the Z80 CPU.

The I/O mapping for the two PIOs used in the design is shown in Table 3.

## DISPLAY UNIT

One of the most popular methods for displaying information (numerical data) in a form that can be understood readily by the user or operator uses a 7-segment configuration to form the decimal characters 0 through 9 and sometimes the hex characters A through F.

The display unit comprises the following:

1. Z80 PIO
2. BCD-to-7-Segment Decoder/Driver
3. 7-Segment display

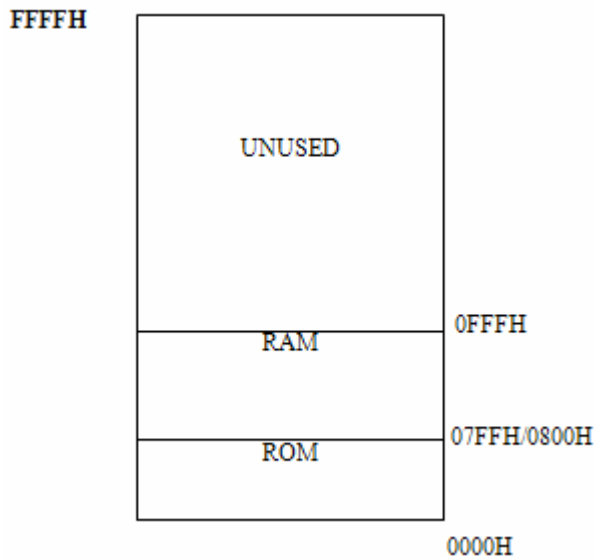
The Z80 PIO used in the display unit provides two 8-bit I/O ports, which have been programmed as output ports. The output of the PIO cannot be fed directly to the 7-segment display, therefore, it needs a driver. The unit sends signals to the driver each time a vehicle crosses the gate.

A BCD-to-7-Segment Decoder/Driver is used to take a four-bit BCD input and provide the outputs that will pass through the appropriate segments to display the decimal digits.

Figure 7 shows the BCD-to-7-Segment Decoder/Driver (74LS47) being used to drive a 7-Segment common anode LED readout. Each segment consists of one or two LEDs. The anodes of the LEDs are all tied to Vcc (+5V). The cathodes of the LEDs are connected through current-limiting resistors to the appropriate outputs of the decoder/driver.

**Table 2: Memory Mapping.**

ADDRESS LINES																	
MEMORY	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	RANGE
ROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	07FFH
RAM	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0800H
	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFFH



**Figure 6: Memory Mapping.**

**Table 3: I/O Mapping**

ADDRESS LINES									
I/O	A7	A6	A5	A4	A3	A2	A1	A0	RANGE
PIO1	PortA	0	0	0	0	0	0	0	00H
	PortB	0	0	0	0	0	0	1	01H
PIO2	PortA	0	0	0	0	1	0	0	04H
	Port B	0	0	0	0	1	0	1	05H

The decoder/driver has active-LOW outputs, which are open-collector driver transistors that can sink a fairly large current. This is because LED readouts may require 10 to 40mA per segment, depending on their type and size.

The display unit is used to show, in decimal values, the number of vehicles that passed through the entrance gate (number of vehicles coming in) and the number of vehicles going out through the exit gate. The difference between

the two gives the number of vehicles in the facility at any time. This serves as a counter.

The system is designed in such a way that it monitors the space available in the park. Once the limit is reached, the alarm system will alert and the system will disallow any additional vehicle that wants to enter unless another vehicle leaves through the exit gate. This feature makes the design work like a monitoring system.

The gate control unit is made up of:

- a. PNP and NPN transistors
- b. Diodes
- c. Motor

Figure 8 shows the gate control unit. The PNP and NPN transistors are arranged in such a way that a pair (PNP and NPN) controls the opening of the gate through the motor and the other pair reverse the polarity of the motor by rotating it in the opposite direction to close the gate. There is a time interval of 10 seconds between the opening and closing of the gate. The software varies this. The arrangement of the diodes serves to protect the transistors from reverse-bias polarity and the resistors serve to improve switching times.

The motor is used to control the opening and closing of gate. The electric (DC) motor used is one that has the ability to rotate in both directions simply by reversing the polarity.

A LOW signal output from a transistor buffer through the Z80 PIO applied to point A bases the NPN and PNP transistors and these cause the motor to rotate in a particular direction. Similarly, a LOW signal applied to point B reverse (change) the rotation of the motor in the opposite direction. The control circuit is used for both entrance and exit gates.

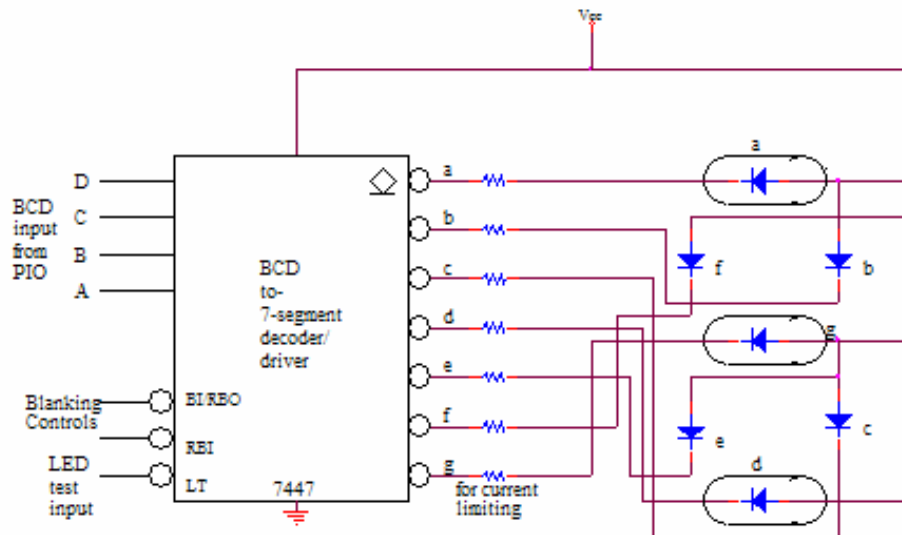


Figure 7: Display Unit.

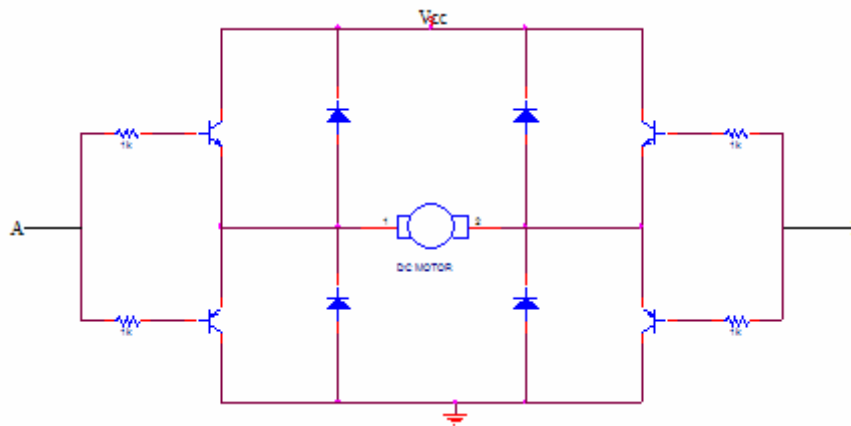


Figure 8: Gate Control Circuit.

## POWER SUPPLY UNIT

A microprocessor based system design has to be activated with a clean power supply of good regulation characteristics. A transient on the power line could send the microprocessor wandering, resulting in system failure. Z80 operates on a voltage  $V_{CC} = 5V \pm 10\%$  and as a result of this, the power supply unit designed is 5V DC and is not affected by variation in the AC voltage serving as input to the transformer. The components used in the power supply (Figure 9) include:

### TRANSFORMER

A 220/240V transformer is used with output voltage of 12V.

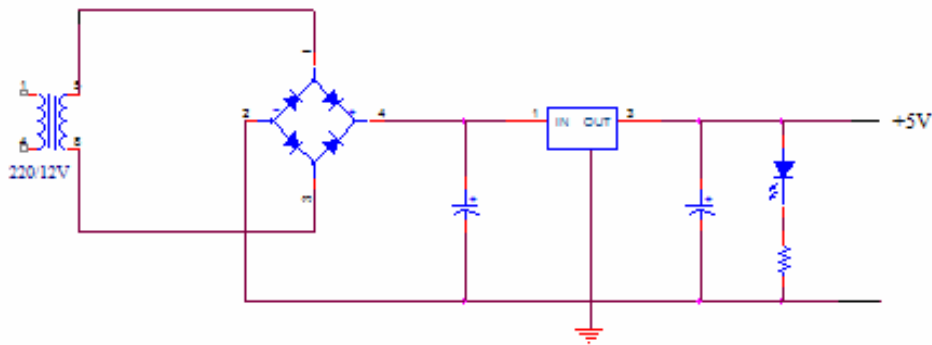
### DIODE 1N4007

This converts the AC current to DC and satisfies charging current demands of the filter capacitor. The arrangement of the diodes is called a bridge rectifier. Rectification is done by the PN junction diodes. The DC voltage varies above and below an average value. This variation is called ripple voltage. In order to reduce ripple voltage to a very small value, the DC voltage needs to be filtered.

### FILTER CAPACITOR

Filter capacitors were chosen to be large enough to reduce the ripple voltage contained in a rectified voltage, to a relatively filtered voltage which resembles a smooth DC voltage as much as possible.





**Figure 9:** Power Supply Unit.

Capacitors of low values were used at the output of the regulator in order to give the power supply a low AC output impedance. To determine the proper value of capacitors used, the equation given below is employed:

$$V_{DC} = V_m - (4.7 \times 10^{-3} I_{DC}) / C$$

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  $\mu F$

#### REGULATOR (7805)

The regulator receives the input of a fairly constant DC voltage and supplies, as output, a somewhat lower value of DC voltage, which it maintains fixed or regulated over a wide range of load current or input variation. The 7805 regulator maintains a 5V DC supply voltage to the system. The main circuit diagram of the system is shown in figure 10.

### **SOFTWARE DESIGN**

#### SOFTWARE DEVELOPMENT PROCEDURE

Designing software for the automatic gate was not a trivial task. In the development cycle of a microprocessor-based system, decisions are made on the parts of the system to be realized in the hardware design and the parts to be implemented in software. 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 software system in order to ensure that the software design meets its specification.

The program for the system is written in assembly language for speed optimization. Assembly code represents halfway position

between machine code and a high level language. The assembly code is usually a mnemonic derived from the instruction itself, i.e. LDA is derived from Load the Accumulator. Assembly code is thus very easy to remember and use when writing programs.

When entering an assembly program into a microprocessor, the assembly code must first be converted into machine code (1998). For short programs, of a few lines, this is relatively easy and usually requires that the programmer construct a table which contains the assembly mnemonics and the equivalent machine code. This technique is known as *Hand Assembly* and is limited to programs of about one hundred lines or less.

For longer programs, a separate program called an assembler program, is used to convert the assembly code into machine code, which is placed directly into the microprocessor memory. The program modules are segmented into:

- a. Main program
- b. Sensor subroutine
- c. Delay subroutine
- d. Output (Gate Control) subroutine

The software was designed using the following steps:

1. Algorithm
2. Flowchart
3. Assembly Language Codes.



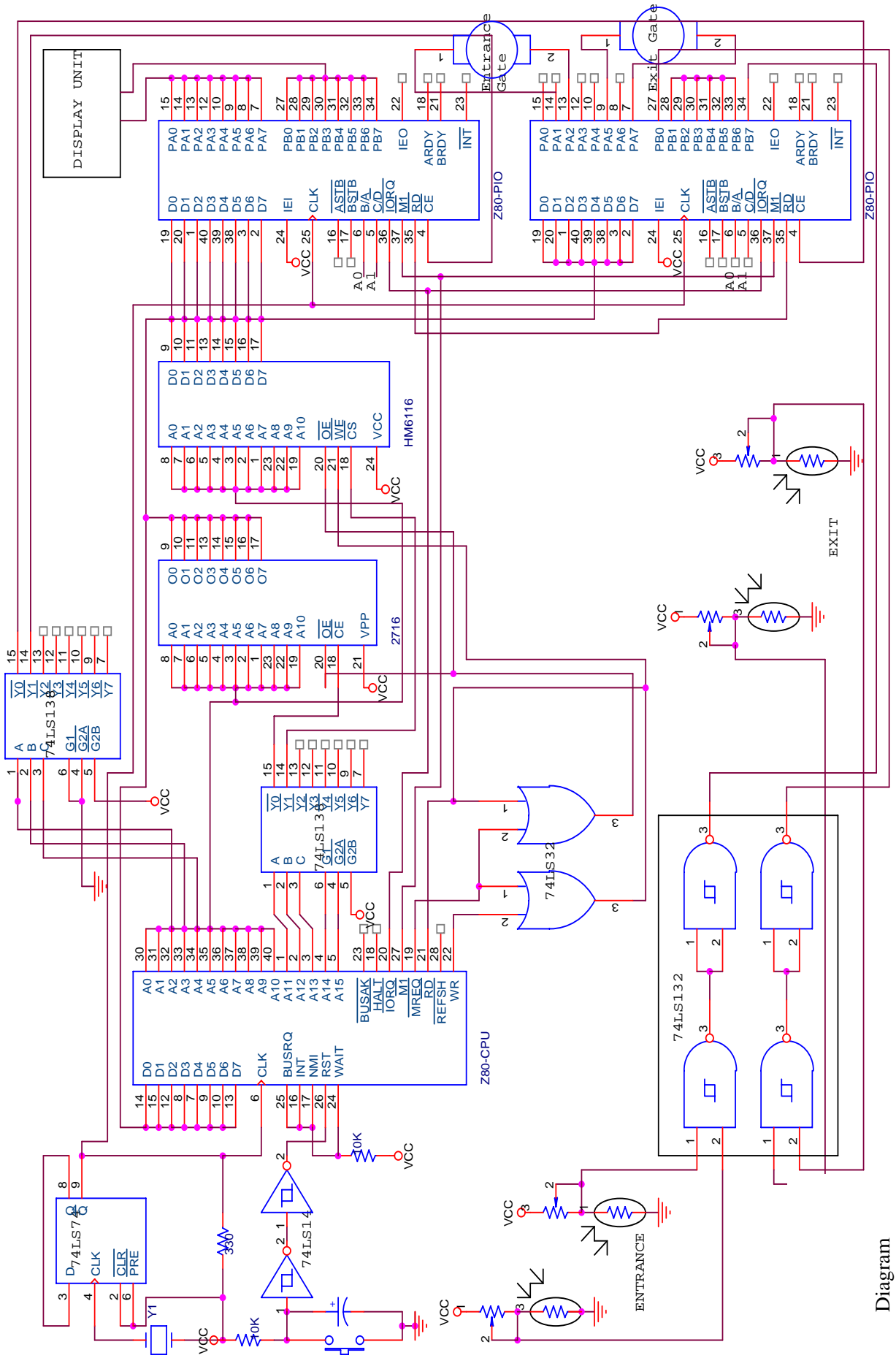


Figure 9: Main Circuit Diagram.

## ALGORITHM

The algorithm used to implement the program for the system described in this paper is as follows:

### START

1. Initialize counter1 = 0, counter2 = 0, spacelimit = 20

2. Fetch the status of the sensor bit

3. Compare the status of the sensor bit with entrance and exit codes

a. If status = entrance code then step 5  
b. Elseif status = exit code then step6

4. Go to step 2

5a. Open, wait and close

b. Increment counter1 and display  
c. Go to step 7

6a. Open, wait and close

b. Increment counter2 and display

7. Subtract counter2 from counter1

8. Compare result with spacelimit

a. If result = spacelimit then step 9  
b. Else go to step 2

9. Fetch status of the sensor bit

10. Compare status of the sensor bit

a. If status = exit code then step6  
b. Else raise alarm

11. Goto step 9

## FLOWCHART

The flowchart for the program is illustrated in Figures 11a, 11b, and 11 c

## THE ASSEMBLY LANGUAGE

### INITIALIZATION

```
LD SP, 0FFFH
LD A, 3FH
OUT 03H
OUT 06H
OUT 07H
LD A, 7FH
OUT 02H
LD H, 00H
LD L, 00H
```

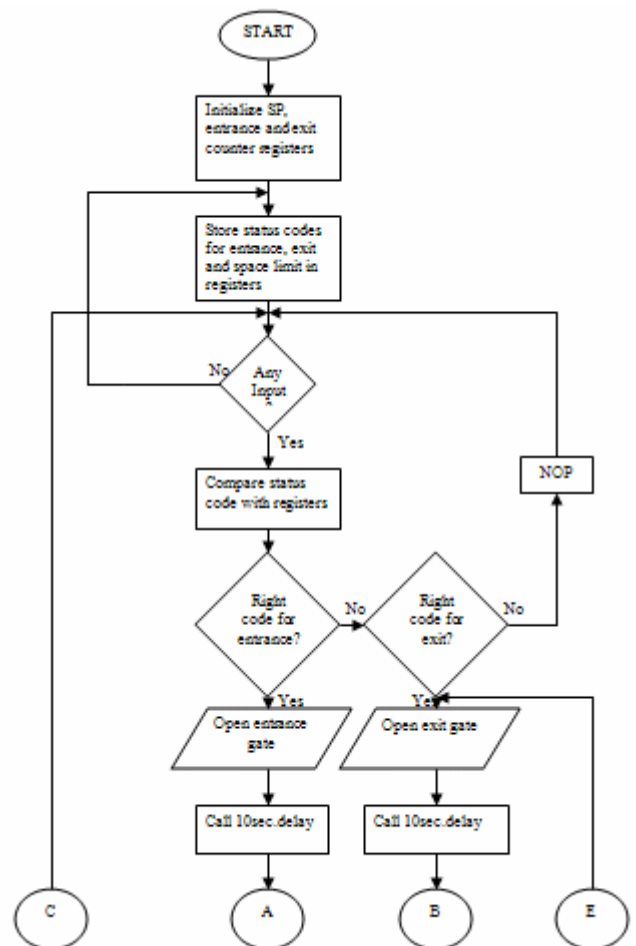


Figure 11a: Flow Chart (Part A).

## MAIN PROGRAM

### START

```
LD B, 80H
LD C, 01H
LD D, 14H
LD A, 00H
OUT(01H)
OUT(04H)
OUT(05H)
```

### MAIN

```
IN A, (00H)CP 00H
JP Z, INPUT
LD E, A
LD A, B
CP E
JP Z, GATE
```

### ENTRANCE

```
LD A, C
CP E
JP Z, GATE EXIT
NOP
NOP
JP START
```

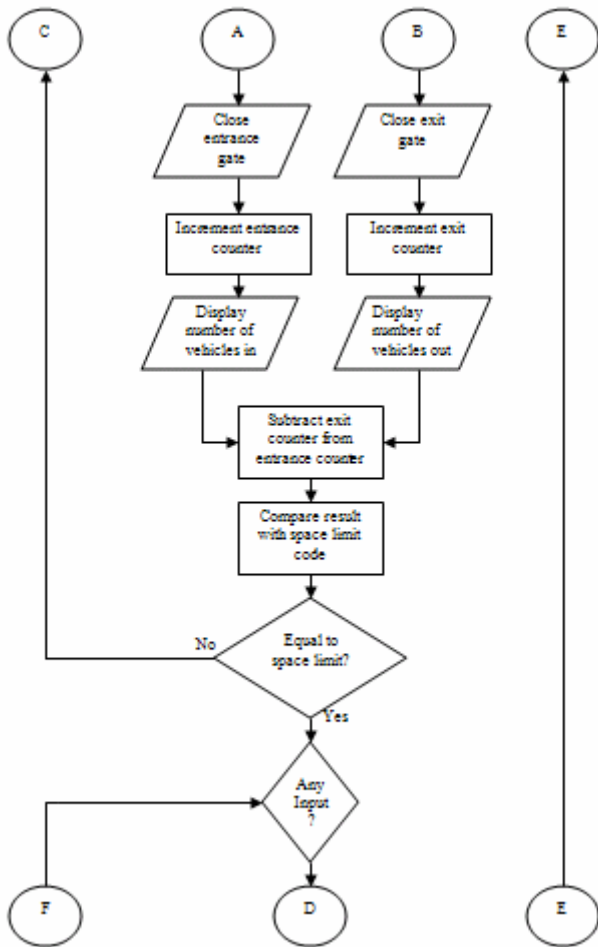


Figure 11b: Flow Chart (Part B).

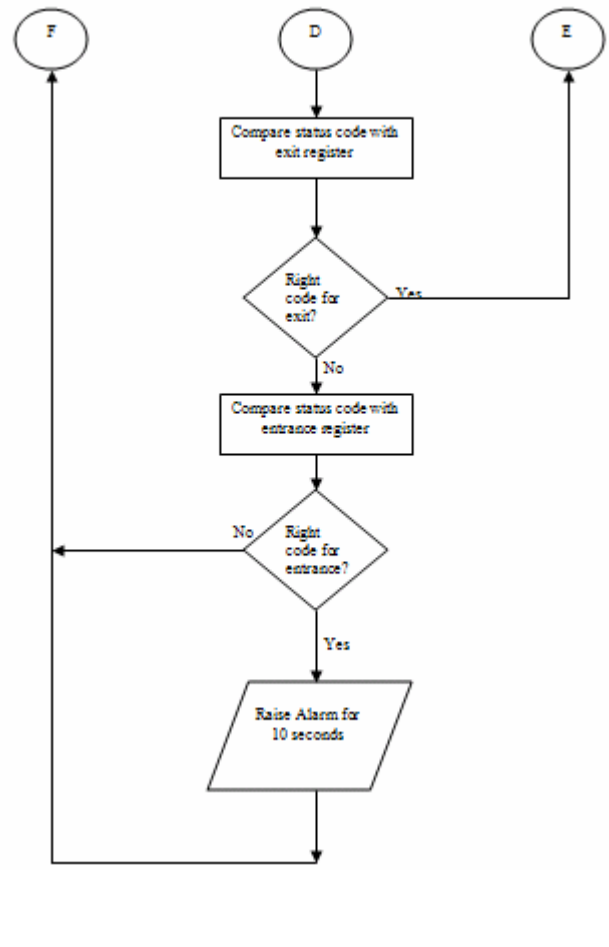


Figure 11c: Flow Chart (Part C).

**INPUT**

```

NOP
NOP
NOP
NOP
JP MAIN

```

**GATE ENTRANCE**

```

LD A, 20H
OUT (01H)
CALL 10sec. DELAY
LD A, 40H
OUT (01H)
NOP
NOP
NOP
INC H
LD A, H
DAA
OUT (05H)
JP SPACE

```

**GATE EXIT**

```

LD A, 02H
OUT (01H)
CALL 10sec.DELAY
LD A, 08H
OUT (01H)
NOP
NOP
NOP
INC L
LD A, L
DAA
OUT (04H)
JP SPACE

```

**SPACE**

```

LD A, H
SUB L
CP D
JP NZ, START

```

**LIMIT**

```
IN A, (00H)
CP OOH
JP Z, LIMIT
LD E, A
LD A, C
CP E
JP Z, GATE EXIT
LD A, B
CP E
JP Z, ALARM
NOP
NOP
JP LIMIT
```

**ALARM**

```
LD A, 01H
OUT (01H)
CALL 10sec.DELAY
JP LIMIT
```

**10sec.DELAY**

```
LD C, 0AH
LD DE, FFFFH
DEC DE
LD A, D
OR E
CP OOH
JPNZ TUNDE2
DEC E
LD A, E
CP 00H
JPNZ TUNDE1
LD C, 01H
LD D, 14H
RET
```

**TUNDE1**

**TUNDE2**

**TESTING AND RESULTS**

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

- Each of the components was first tested using a multimeter in order to check for their state of performance and accurate values.
- The connection of each component on the vero board was then tested. This was done in order to assess the continuity, which is meant for proper connection of the circuit, and to detect any wrong connections.
- The sensor unit circuitry was tested to ascertain the degree of sensitivity. A small prototype car (object) was placed between the two pairs of the photoconductive cell to obstruct light rays. The voltage levels at the

output were observed with the aid of a digital multimeter. The result is shown in Table 4.

- Before the results were obtained, the variable resistor was adjusted to obtain the output voltages (2004).
- The outputs of the trigger circuitry were tested by connecting an LED across the circuit to check if it lights or not. A light indicates the presence of HIGH logic while a non-light indicates the presence of LOW logic as shown in Table 5.
- Also, the gate control circuit was tested by applying logic 1 or 0 to points A and B of the circuit. When logic 1 is applied to point A, the motor rotates in a clockwise direction while logic 1 at point B changes the direction of the motor. Logic 0 at both points will never activate the motor. The result is shown in Table 6.
- After the peripherals were tested and found to be in working order, the entire circuit was tested. A series of programs (software) were written and tested before the working program was finally achieved. The circuits worked perfectly as designed. The display unit was also observed during the testing and found to be working correctly.

**Table 4:** Voltage Levels of the Sensor Unit.

Test	Result
Without Object	0.8V
With Object	4.39V
Object Removed	0.8V

**Table 5:** Trigger Circuitry Logic Level

Sensor Voltage Levels		Output Logic
V01	V02	F
0.80	0.80	0
0.80	4.39	0
4.39	0.80	0
4.39	4.39	1

**Table 6:** Gate Control Circuit Truth Table.

A	B	Motor Direction
0	0	Inactive
0	1	Anticlockwise
1	0	Clockwise
1	1	Inactive

## CONCLUSION

The design and implementation of a microcomputer system had been achieved in this project. This design can be easily adapted to any electric gate and any form of control which requires the use of sensors. To effectively design this kind of system, it is necessary to understand the basic sensor characteristics, microprocessor input and output interfacing, and assembly language principles, utilized in the system plan. Sensors serve as a transducer for vehicle detection while the programming language is fundamental to software design based on the system requirements, specifications, and planned operation of the system. There is total agreement between the system designed and the required operation of the system.

Every good project has limitations; the limitation of this design lies in the effectiveness of the sensor. The sensor will work most effectively if operated under high intensity light. The automatic gate designed in this research can be employed in organizations, public car parks, residential parking lots, and automobile termini where no form of security measure is required.

## RECOMMENDATION

For an improved, effective, and security gate system to be implemented and achieved, the following suggestions should be considered for further work.

1. A form of vehicle identification should be provided for security purposes. For instance where a vehicle stands still at the focus of the sensors.
2. A better sensor is recommended to achieve new functionality. For instance, a suitable sensor such as radar sensor that could detect contraband goods in any vehicle.

3. To achieve full automation, a real time system should be employed and a Closed Circuit Television (CCTV) system provided for proper monitoring and security purposes. This can be helpful in detecting the presence of vehicles before the system is activated.

4. Upgrading the system using a higher bit microprocessors for speed optimization.

With the revolutionary impact that microprocessors have had on the electronics industry, it is not unreasonable to expect that everyone working in electronics and related areas will have to become knowledgeable with the operation and integration of microprocessors.

## REFERENCES

- Access Automation, LTD. 2006. "Commercial Barriers". Sommerset, UK. <http://www.access-automation.co.uk>.
- Baruwa, Olatunde. T. 2004. "Design and Construction of a Microprocessor Based Automatic Gate". Unpublished B.Sc. Project. Lagos State University: Epe, Nigeria.
- Hall, Douglas V. 1991. *Microprocessors and Interfacing Programming and Hardware. 2nd edition.* Gregg College Division: New York, NY.
- Krutz, R.L. 1980. *Microprocessor and Logic Design.* John Wiley & Sons, Inc.: New York, NY.
- Leventhal, Lance A. 1978. *8080A, 8085 Assembly Language Programming.* McGraw-Hill, Inc: New York, NY.
- McGlynn, Daniel R. 1976. *Microprocessor Technology and Application.* John Wiley & Sons, Inc: New York, NY.
- Philips ECG. 2000. *ECG Data Book.* Bloomfield, NJ.
- Private Door Openers. 2006. "Private Door Information". Lombard, IL. <http://www.privatedoor.com>.
- Stewart, P.M. 1983. "Techniques for Vehicle Detection Report". Unpublished.
- Theraja, A.K and Theraja, B.K, 1999. *Electrical Technology. 3rd Edition.* S. Chand and Co.: New Delhi, India.
- Tocci, Ronald J. and Neal S.Widmer. 1998. *Digital Systems; Principles and Application. 7th Edition.* Prentice-Hall International: Princeton, NJ.
- Tokheim, Roger L. 1988. *Digital Electronics; Principles and Applications. 5th Edition.* McGraw-Hill, Inc: New York, NY.
- Tseng, Vincent. 1982. *Microprocessor Development and Development Systems.* McGraw-Hill, Inc: New York, NY.

## ABOUT THE AUTHORS

**O. Shoewu, M.Sc., B.Sc.(Hons), MNSE**, serves as a Lecturer at the Lagos State University. He earned his M.Sc. and B.Sc. from the Lagos State University and the University of Lagos in 1992 and 1995, respectively. He is presently a postgraduate student working towards a Doctoral degree in Electronics. His research interests are in the areas of electronics, computers, and communications technology.

**O.T. Baruwa**, is a researcher in the Electronic and Computer Engineering Department at Lagos State University in Epe, Nigeria. His research interests are in the areas of electronics and computer controlled applications.

## SUGGESTED CITATION

Shoewu, O. and O.T. Baruwa. 2006. "Design of a Microprocessor Based Automatic Gate". *Pacific Journal of Science and Technology*. 7(1):31-44.

 [Pacific Journal of Science and Technology](http://www.akamaiuniversity.us/PJST.htm)