DESIGN AND CONSTRUCTION OF AN INTRUDER DETECTOR CONTROLLED BY A MICROCOMPUTER

BY

HAMZA SALIFOU

REG. N^{O} 98 / 7697EE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SCHOOL OF ENGINEERING AND ENGINEERING TECHNOLOGY

FEDERAL UNIVERSITY OF TECHNOLOGY

MINNA, NIGER STATE,

NIGERIA

OCTOBER, 2003.

DECLARATION

I hereby declare that the project work is an original concept wholly carried out by HAMZA SALIFOU, under the supervision of Mr. RUMALA S.N. of the Department of Electrical and Computer Engineering, Federal University of Technology, Minna.

Hamza Salifou

17-10-2003

CERTIFICATION

I certify that this work was carried out by HAMZA SALIFOU with Reg. N° 98/7697EE of the Department of Electrical and Computer Engineering, Federal University of Technology, Minna.

....

M^r RUMALA S.N. date

(Supervisor)

.... 28/4/09

ENG. M. N. NWOHU Date

(H.O.D)

....

(External Examiner) Date

DEDICATION

I dedicate this to my country (NIGER REPUBLIC), my dear Father (Hamza Namata), my mothers, my brothers and sisters, my future wife Hadjaratou Abani. It is also for all those suffering of injustice, poverty and those lost their life in struggling for liberty and justice.

ACKNOWLEDGEMENT

First of all my profound gratitude and thanks goes to Almighty ALLAH, who by his endless mercy allow me to witness this great stage of my life, who gives me the strength, the courage and the guidance to withstand all obstacles on my way-all praises are due to him.

My heartfelt appreciation goes to my supervisor M^{R} RUMALA whose effort has been to bring out the best of me. I appreciate the effort made by the departmental lecturers in seeing that I reach this level.

I am sincerely indebted to Hamza Namata whose tireless effort brought me to this work. I also owe Mahamadou S.S, Ada T. Mariama Dan Djibo, Hadjia, Ajjiboye J., Zelia A. and Friends for their assistance in various ways. All those who one way or another directly or indirectly contributes for the achievement of this work. My gratitude also goes to Maharadji I., villagers, and all Niger Republic students for making my staying in campus easier. This acknowledgement will not be complete if I fail to appreciate the love and moral support of Zeinabou Sabi and the financial support of Niger Republic Government without which I will never reach this stage.

I must also thank all my classmates and the entire people of Nigeria particularly Bosso people who have shown me the legendary hospitality of Africans.

ABSTRACT

Nowadays, security is the major problem people encounter in towns, villages, and houses as well as in roads. The availability of computers at low price and its wide spread in homes and offices push me to think on how to use them to watch our homes and offices. This is achieved by using some switches well set at the entrances which states are continuously communicated to the computer, which in turn will inform the user either the entrances are closed or opened.

TABLE OF CONTENTS

CHAPTER ONE

1.1 INTRODUCTION

Man, since his creation, is looking for the ways to improve his conditions of existence. The first thing that man looks for was security whether against his fellow human being, animals or the nature. Security means measures taken to present spying, attacks, theft, etc... Then man, day after days improves his conditions of life by inventing machines. The most powerful and ingenuous invention of man is the computer, pile of electronic components that accepts data, processes them and gives them back to the user for further purposes. The continuous increasing in performance and price ratio of microcomputers makes them very attractive as intelligent controllers in domestic and industrial applications. Their digital programmability, flexibility and processing power combine with the availability of a wide range of commercial software packages make them suitable for applications in offices, laboratories, on shop floors, in our cars. Every aspect of our life to day is affected. In the ordinary way components are feed through the keyboard and responds through its monitor another more sophisticated way is to use sensors to input data and the response is used to control a process at the light of this analysis, the problematic is " how can we used microcomputers to assure our home security?"

1.2 LITERATURE REVIEW

The earliest proposal to use a computer operating in " real-time" as part of a control system was made in a proper by Brown and Campbell. The paper contains a diagram (below) which shows a computer in both the feedback and feedforward loops. Brown and Campbell (1950) assumed that analog computing elements were the most likely to be used but they did not rule out the use of digital computing elements. The first digital computers developed specifically for real-time control were for airborne operation, and in 1954 a digitrac digital computer was successfully use to provide an automatic flight and weapons control system.

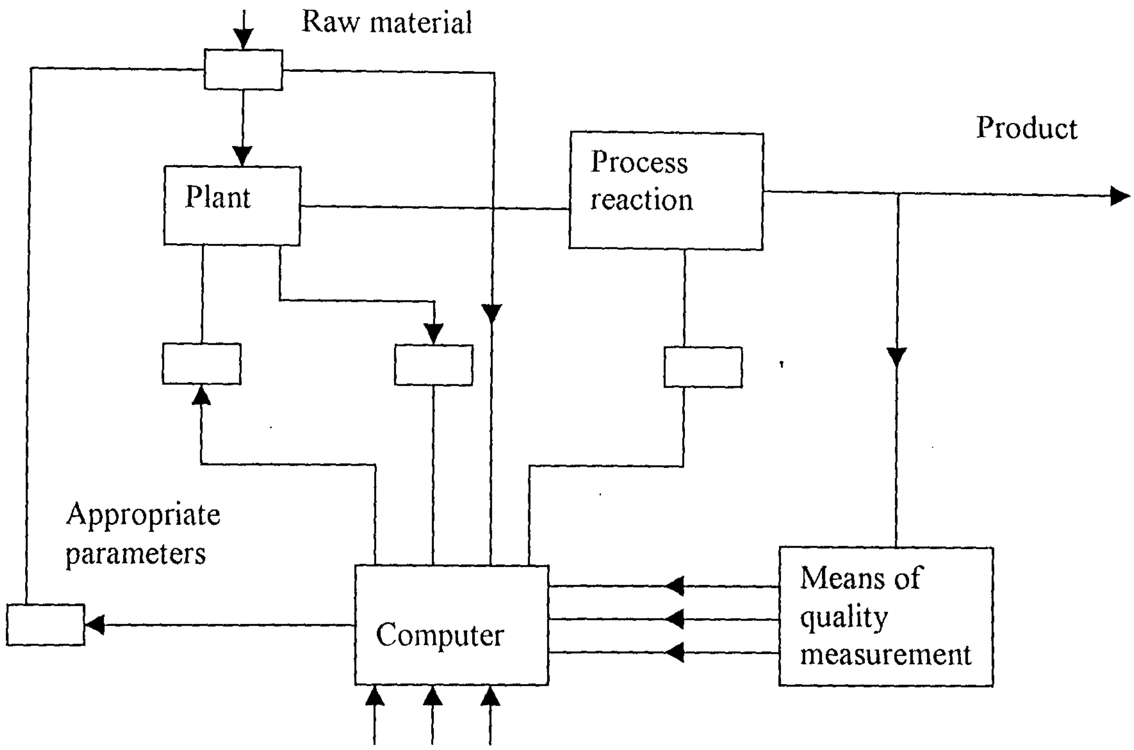


Fig 1.1 Computer used in control of plant (redraw from Brown and Campbell Mechanical Engineering, 72, 1950)

The application of digital computers to industrial control began in the late 1950 S. The initiative came, not from the process and manufacturing industries, but from the computer and electronic systems manufactures who were looking to extend their markets and to find outlets for equipment which had failed to be adopted by the military [Williams1983]. The first industrial installation of a computer system accrued in September 1958 when the Louisiana lower and light company installed a Daystorm computer system for plant monitoring at their power station in sterling, Louisiana. This was not a control system: the honor of the first industrial computer control installation went to the Texaco company who installed an RW. 300 system at their port Arthur refinery in Texas which achieved closed-loop control on March 15, 1959.

During 1957-8 the Monsanto chemical company in cooperation with the Ramo-wooldridge Company, studied the possibility of using computer control and in October 1958 decided to implement a scheme on the ammonia plant at Lulling, Louisiana. Commissioning of this plant began on January 20, 1960 and closed-loop control was achieved on April 4, 1960 after an almost complete rewrite of the control algorithm part of the program and considerable problems with noise on the measurement signals. This scheme, like the system installed by the B.F Goodrich company on their acrylanite plant at calvert city, Kentucky in 1959-60, and some 40 other systems based on the Rw-300, were supervisory control systems used for steady-state optimization calculations to determine the set-points for standard analog controllers, that is, the computer did not control directly the movement of the values or other plant actuators.

The first direct digital control computer system was the Fernati Argus 200 system installed in November 1962 at the ICI ammoniasoda plant at Fleetwood, Lancashire, the planning for which had begun in 1959. It was a large system with provision for 120 control loops and 256 measurements, of which 98 and 224 respectively were used on the Fleetwood system. In 1961 the Monsanto Company also began a DDC project for a plant in Texas City and a hierarchical control scheme for the petrochemical complex at chocolate Bayou.

The Ferranti Argus represented a change in computer hardware deign in that the control program was held in a ferrite core store rather than on a rotating drum store as used by RW-300 computer. The program was held in a programmable read-only memory. It was loaded by physically inserting pegs into a plug board, each peg respesenting one bit in the memory word. Although laborious to set up initially, the system proved to be very reliable in that destruction of the memory contents could only be brought about by the physical dislodgment of the pegs. In addition, security was enhanced by using special power supplies and switch-over mechanisms to protect information held in the main core store. This information was classified as follows:

1. Set points loss most undesirable
2. Valve demand presence after controlled stoppage allows computer to gain control of plant immediately and without disturbance bumbles transfer.
3. Memory calculations loss is tolerable, soon will be updated and only slight disturbance to plant and
4. Future development calculation extension to allow for optimization may require information to be maintained for log periods of time.

In addition to improved reliability the Argus system provided more rapid memory access than the drum stores of the RW-300 and similar machines and as such represented the beginning of the second phase of application of computers to real-time control.

The computers used in the early 1960S combined magnetic core memories and drum stores, the drums eventually giving way to hard disk drives. They included the general. Electric 4000 series, IBM 18000, CDC 1700, Foxboro Foxl and, the SDS and Xerox SIGMA series, Ferranti Argus series and Elliot Automation 900 series. The attempt to resolve some of the systems: the increase was such that frequency their use could be justified only if both DDC and supervisory control were performed by the one computer.

A consequence of this was the generation of further problems particularly in the development of the software. The programs for the early computers had been written by specialist programmers using machine code, and this was manageable because the tasks clearly were defined and the quantity of code of the two tasks are very different and relatively small.

In combining DDC and supervisory control, not only had the quantity of code for a given application increased, but the complexity of the programming also increased in that the tow tasks had very different time-scales; and the DDC control programs had to be able to interrupt the supervisory control programs. The increase in the size of the programs meant that not all the code could be stored in core memory. Provision had to be made for the swapping of code between the drum memory and core.

The solution appeared to lie in the development of general purpose operating systems and high level languages. In the late 1960 S real time operating systems were developed and various processes Fortran compilers made their appearance. The problems and the costs involved in attempting to do everything in one computer led users to retreat to smaller systems for which the newly developing minicomputer were to prove ideally suited. The cost of the minicomputer was small enough to avoid the need to load a large number of tasks onto one machine; indeed by 1970 it was becoming possible to consider having two computers on the system, one simply acting as a stand by in the event of failure. The advent of the microprocessor in 1974 led to further reappraisal of approach and the development of distributed systems [4].

The problems and the cost involved in attempting to do everything in one computer led users to retreat to smaller systems for which the newly developing minicomputer were to prove ideally suited. The cost of the minicomputer was small enough to avoid the need to load a large number of tasks onto one machine indeed by 1970 it was becoming possible to consider having to computers on the system, one simply acting as a stand-by in the event of failure. The advent of the microprocessor in 1974 led to further reappraisal of approach and the development of distributed systems.

1.3 OBJECTIVE OF THE PROJECT

The main aim of this project is to figure out how wide is the field where microcomputer can be used. From what we are seeing now, computers are under-used in our under-developed countries. They are mainly used as typewriter or data processors. Nowadays, microcomputers are found everywhere; so we are using this opportunity to make them assuring our home security in addition of its primary functions so in not shell the project is to show the optimum use of a microcomputer.

1.4 PROJECT LAYOUT

In this section we are going to introduce each chapter that will constitute this project. In chapter one we try to explain the topic, our objectives and why we choose the topic. Also we point out the evolution of the computer çontrolling process. Chapter two deals with the hardware part of the interface, which is the printer port of the microcomputer. Chapter three, which discusses on the software, interface that monitor the device through parallel port. Chapter four gives detail on the different components that constitute the device. Chapter five deals with the construction, how all the components are brought together to perform the desired work. It also shows the results and the result analysis of the testing device. Chapter six is the conclusion and recommendations that we made based on what we have encountered during our work.

CHAPTER TWO

2.0 HARDWARE INTERFACE OF THE PROJECT

2.1 INTRODUCTION

Interfacing is the design of circuitry between devices that shifts voltage and current levels to make them compatible. In interfacing between logic devices and the "outside world" the voltage and current characteristics are very important factors. It is used to increase the drive capabilities. This approach also reduces the probability of damaging the microcomputer or other devices as a result of a fault on the introduced interface circuit [9]. Interface allows the user to supply information to the computer which permit it to fulfil a certain requirement and vice-versa. In this project we use the parallel port a sour hardware interface.

2.2 GENERATING THE "DEVICE SELECTION" SIGNAL OF THE PARALLEL PORT.

When interfacing devices to a microcomputer, each device needs to be given a unique address. A logic circuit is required to produce a pulse only when that address is supplied by the microprocessor. This pulse selects a device to communicate with the microcomputer. During communication, the selected device supplies information to the microcomputer, or reads information from it. The microcomputer selects the device it wants to communicate with by supplying a suitable "device selection" pulse to the device enable line. There are two methods of interfacing a device: memory mapping in which the device is given one or more memory locations and input/output mapping.depending on several factors such as the microcomputer type and availability of address we can have different ways of describing selection signals. In order to interface the microprocessor needs to address the device and also control it. This shows that a "device selection" signal will have two components: device-control signal and addressing signal [3].

2.2.1 DERIVING DEVICE CONTROL SIGNAL

Microprocessor can read or write data to devices. This action is controlled by a control line called R/$\overline{W}$. It identifies the direction of transfer of information. Depending on the microprocessor these signals can be bore by a single line (MC68000) or two different lines (8086, 80186). Let us assume that we want a computer to read information from the parallel port status and write information to the parallel port output port. To do this the decoding circuit should enable the port by generating a pulse when their addresses are selected by the microprocessor and depending on which of the read or write line is active status port or output port is selected. Notice that they can have the same address as shown in the figure below:

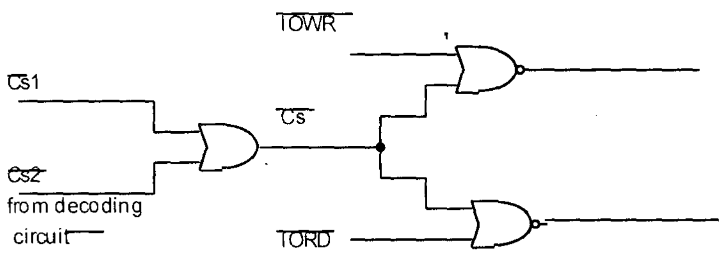


Fig. 2.1 generating control signal for parallel port.

In addition to the read and write signals some microprocessors include one or two control signal to permit supporting input/output mapping as well as memory mapping. For instance 8086 and 80186 have M / $\overline{IO}$ line, 8083 has IO/$\overline{M}$ line Z80 has 2 lines: IORQ and MEMRQ.

2.2.2 DERIVING ADDRESSING SIGNALS

A microprocessor uses an addressing signal to identify the location of a device within the addressing area. When interfacing a device to a microcomputer the designer needs to consult the memory map of the microcomputer to find out the area available for use in deriving addressing signals. The size and boundaries of the area vary from one microcomputer to another. The addressing capability of a microcomputer can be different from that of another for instance the 8086 provides 20 address line which are active during memory map to produce the required addressing signal. During an input / output mapping however, only the lower 16 address lines are active. Let us derive the addressing signal from the microcomputer that address the three ports of parallel port at any instant of time and used to address a port. The area between locations 378H and 37AH is available for addressing by 10 address lines A_0 - A_9. Using the 7415138 decoder we can easily generate the addressing signals by using all of the address lines. 741S138 is a 3 by 8 8 decoder with 3 enable lines. So the address in binary will give 1101111000, 1101111001 and 1101111010 so A0 and A1 will be used to distinguish between signals.

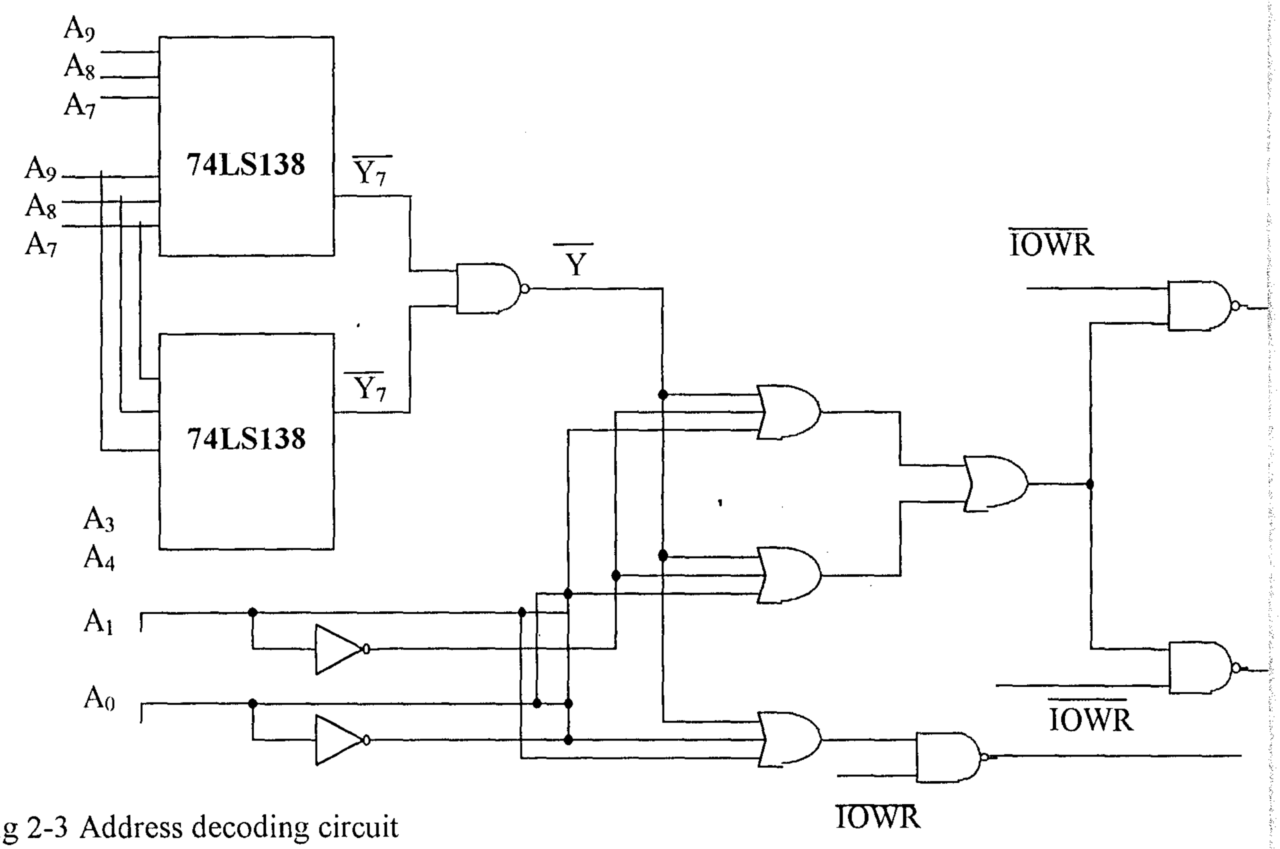


Fig 2-3 Address decoding circuit

2.2.3 INPUT PORT

An input port is a three-state buffer whose output is equal to its input only when enabled by a controlling device, such as a microcomputer. It normally consists of several channels. When enabled, the port supplies information from the device to the microcomputer through the data bus. It can be formed using one or many ICS of type 74 LS244, 74LS125. The buffers of the IC are non latching because an enable input port supplies information to a data bus which is time-shared by several devices and accepts signals from one device at a time.

2.2-4 OUTPUT PORT

The requirement imposed on an output port is different from that imposed on an input port. That is because the later needs to put signals on the data bus only for a very short period of time on the other hand, the former is supplied with signals from the data bus for a very short time and is required to latch those signals so that the output will be available as long as the buffer is supplied with power. Consequently, an output port consists of three-state latches: 74LS374.

2.2.5 ADDRESS AND DATA BUFFERS

The address and data buses of microcomputers are capable of driving several external devices in addition to those already connected inside. If many external devices are required to be connected to those buses, then buffers can be introduced to increase the drive capability of the microcomputer. As the address bus is unidirectional, it can be buffered by using a 74LS244 octal buffer IC. The data bus is bi-directional. It can use a buffer that allows flow in both directions so the 74LS245 octal bus transceiver IC can do the job.

2.3THE PARALLEL PORT FUNCTIONALITY

Let's look at the schematic diagram of the port:

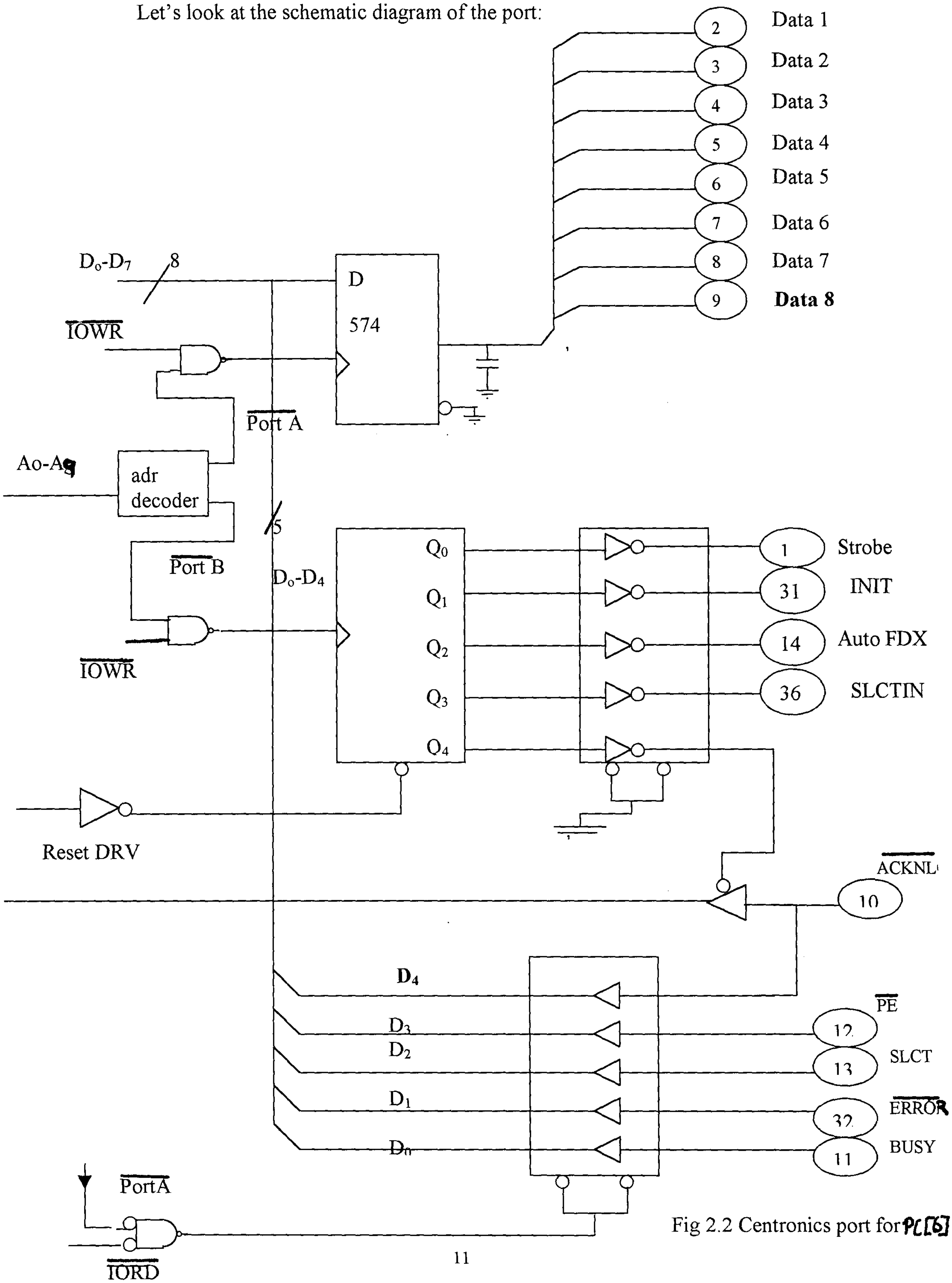


Fig 2.2 Centronics port for PC[6]

Fig 2.2 shows the pin connections PC connector. The data lines ($D_0 - D_7$) output data from the PC and each of the data lines has an associated ground lines (GND). The main handshaking lines are $\overline{\text{Ack}}$, Busy, and $\overline{\text{strobe}}$. Initially the computer places the data on the data bus, then it sets the $\overline{\text{strobe}}$ line low to inform the external device that the data on the data bus is valid. When the external device has read the data it sets the $\overline{\text{Ack}}$ line low to acknowledge that it has read the data. In the case of a printer the PC should wait for the busy line to be low before to send the followind byte of data. The parallel port has three registers each with its own address. The data register at 378 H, status register at 379H and the control register at 37AH for LPT 1. These registers can be grouped as output lines and imnput lines [2].

2.3.1 OUTPUT LINES

The data port register links to the output lines, writting a 1 into the port register sets the output line high, while a 0 sets it to low. The output data lines are each capable of sourcing 2.6 mA and sinking 24 mA. The control port also contains five output lines which the lower four bits are $\overline{\text{strobe}}$, $\overline{\text{auto feed}}$, INIT and $\overline{\text{select input}}$. These lines can be used as either control lines or as data outputs. The lines in the control port have invented logic, that is a 1 to a bit in the register causes a low output. The 5^{th} bit is the interrupt enable bit. It is used to cause the processor to be interrupted. The control lines are driven by open collector drivers pulled to + 5 vdc through 4.7 kΩ resistors. Each can sink approximately 7 mA and maintain 0.8 V down level.

2.3.3. INPUT LINES

There are five from the parallel port (Busy, $\overline{\text{Ack}}$, $\overline{\text{PE}}$, select , and $\overline{\text{error}}$). The status of these lines can be found by simple reaading the upper 5 bits of the status register. But the Busy line has an inveted status. Thus when low is presented on Busy, the bit will actually be read as 1.

2.3.4 INTERRUPT-DRIVEN PARALLEL PORT

Each parallel port is hooked to an interrupt, normally LPT1 is connected to IRQ7. To enable the IRQ the most significant bit of the interrupt mask register (21h) must be set to 0 and to disable it the EOI signal bit of the interrupt control regidster (20h) must be set to 1. The interrupt on the parallel port is caused by the ack line going from a high to a low . For this interrupt to be passed to the PC then bit 4 of the control port register must be set to 1.

2.3.5 MODE OF OPERATION OF A PARALLEL PORT

the parallel port can work in either of these modes:

- Compatibility mode: this mode defines the transfer of data between the PC and the printer.
- Nible mode: this mode defines how 4 bits are transferred at a time , using status lines for the input data. The nible mode can thus be used for bi-directional communication, with the data lines being used as output.
- Byte mode: this defines how 8 bits are transferred at a time. It is also known as the bi-directional port because it uses bi-directional data lines.
- Enhanced parallel port (EPP): this mode defines the standard bi-derctional communication and is used by many peripherals such as CD-ROMS, tape drives etc...
- The extanded capacity port protocol in an advanced mode for communication with printer and scanner type peripherals.
- The negociation mode allows the host to determine the attached peripherals and the method used to control them.

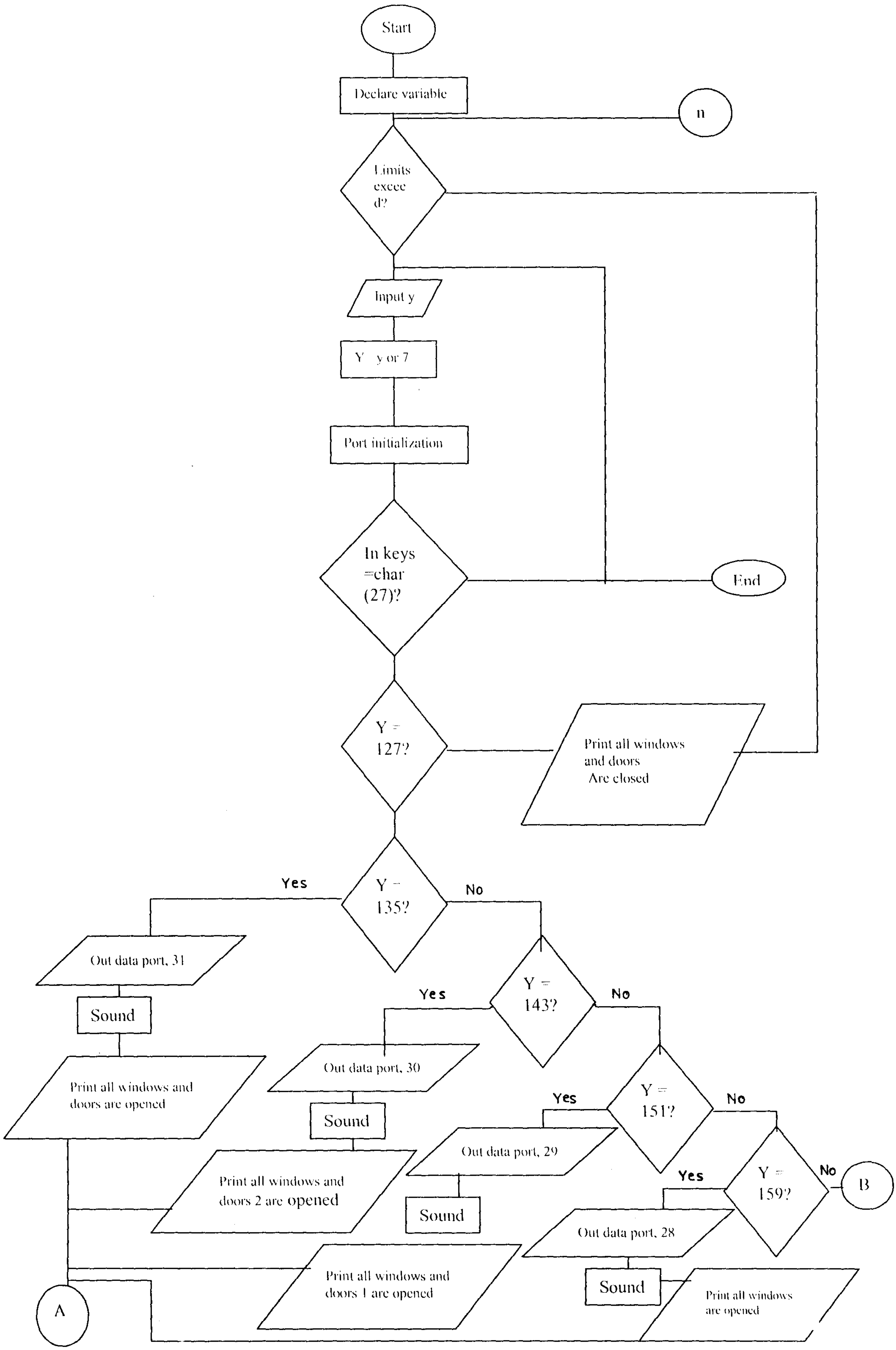
CHAPTER THREE

3.0 SOFTWARE INTERFACING DESIGN

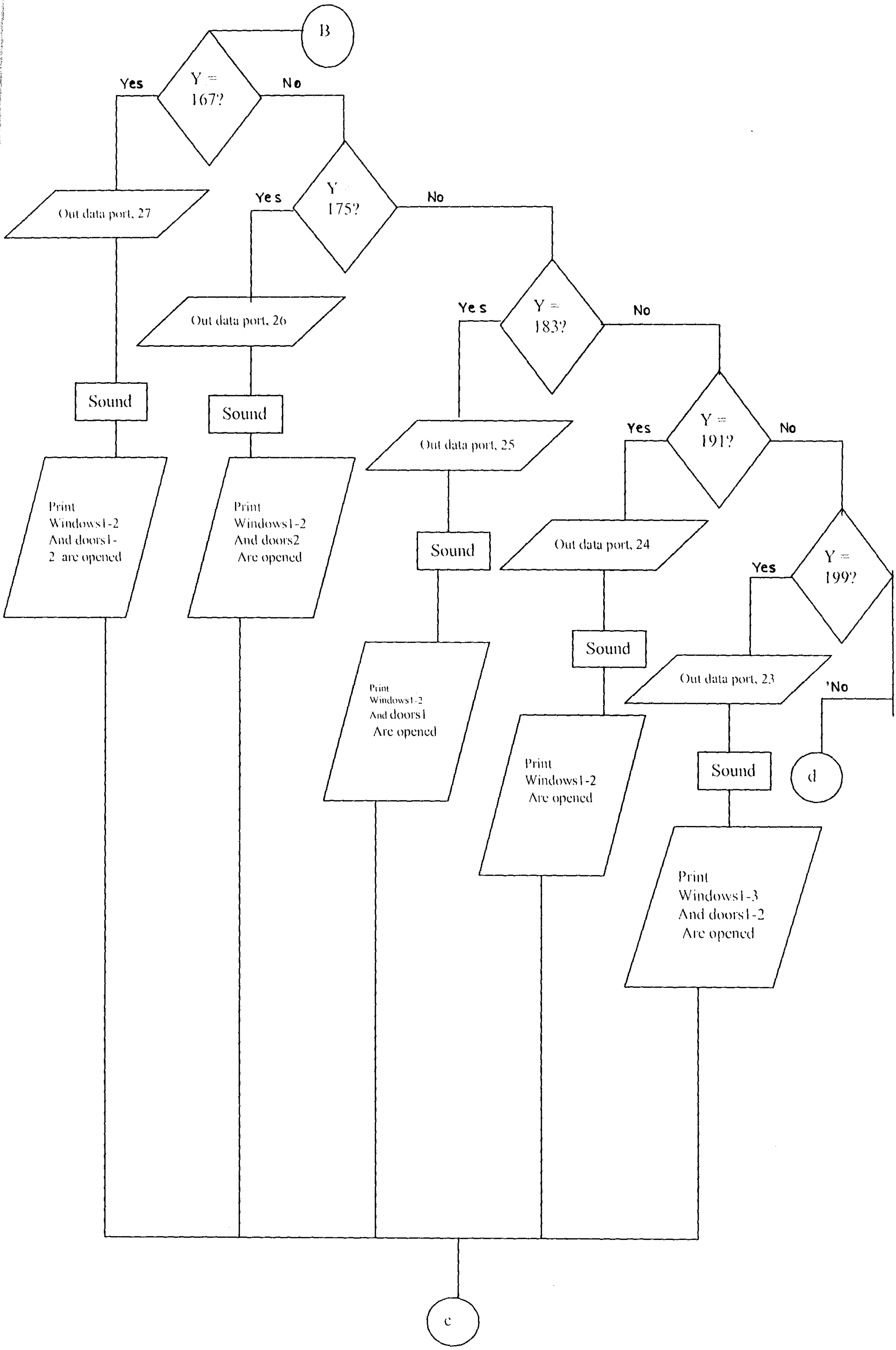
3.1 INTRODUCTION

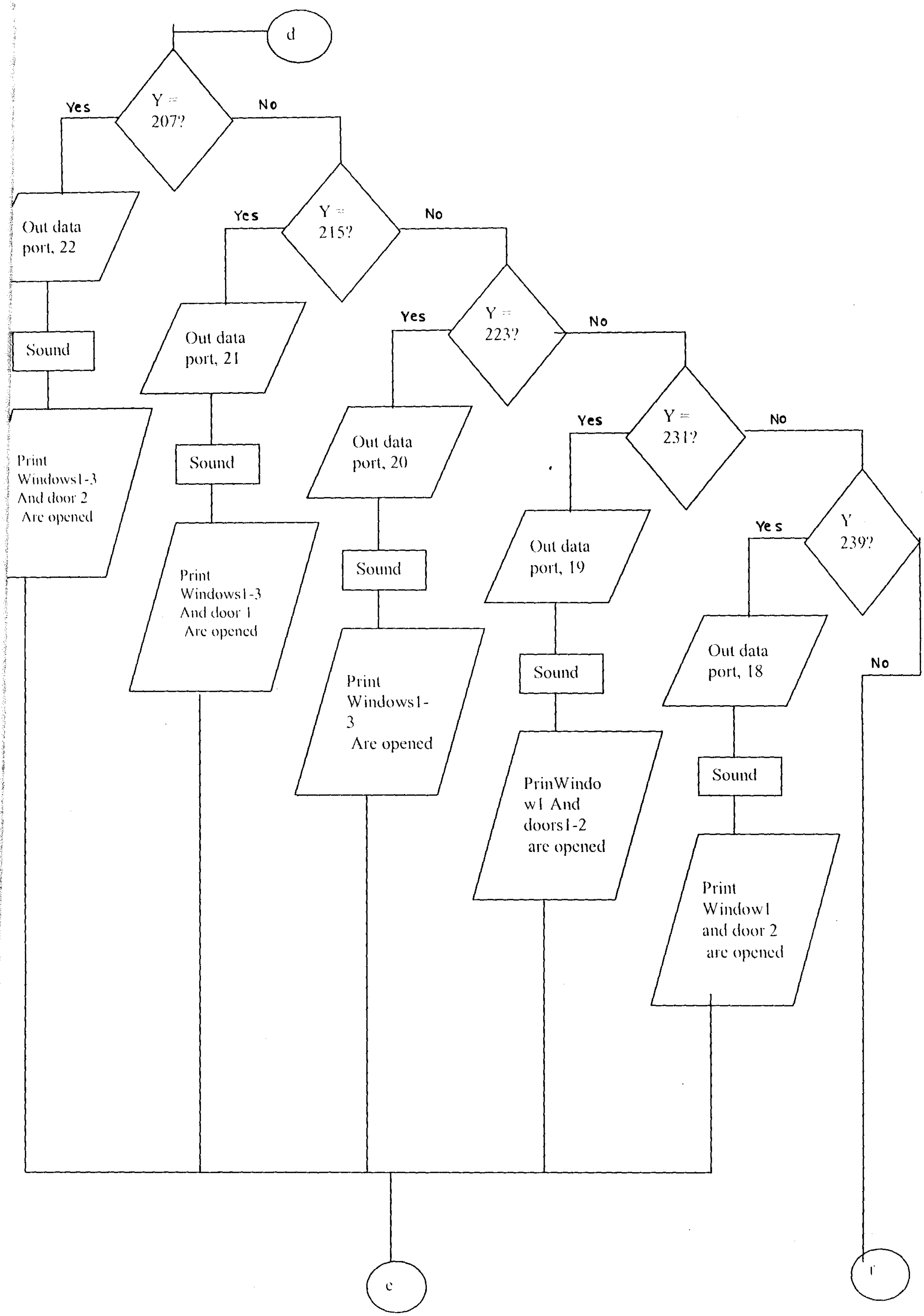
A computer is consisted of two main parts: hardware and software. The later is a set if insructions that the user uses to make the hardware do whatever he wants it to do. The software is also called program. So they are means by which the user communicates with the computer. To communicate with someone you should understand each other language. The mother tongue of the computer is the machine language. This consists of strings of binary numbers and every computer has its own operation code. Each instruction in the machine language has at least two parts: the command and the operand. Programming in this language is very tedious, time consuming, expensive and error prone. To ease the programmer's burden, assembly languages was developed. It consists of mnemonics,operations codes, and symbolic addresses. These mnemonics are converted into machine language by assembler before being meaningfull to the computer. The drowback of assembly languages is that it is machine oriented. This limitation was overcome by developping high level languages. These use english words and have to be compiled or translated before to be understandable to the computer[8]. An example is BASIC. We are going to use it to develop the program that will monitor our device because of it is interactive, simple and also available[1].

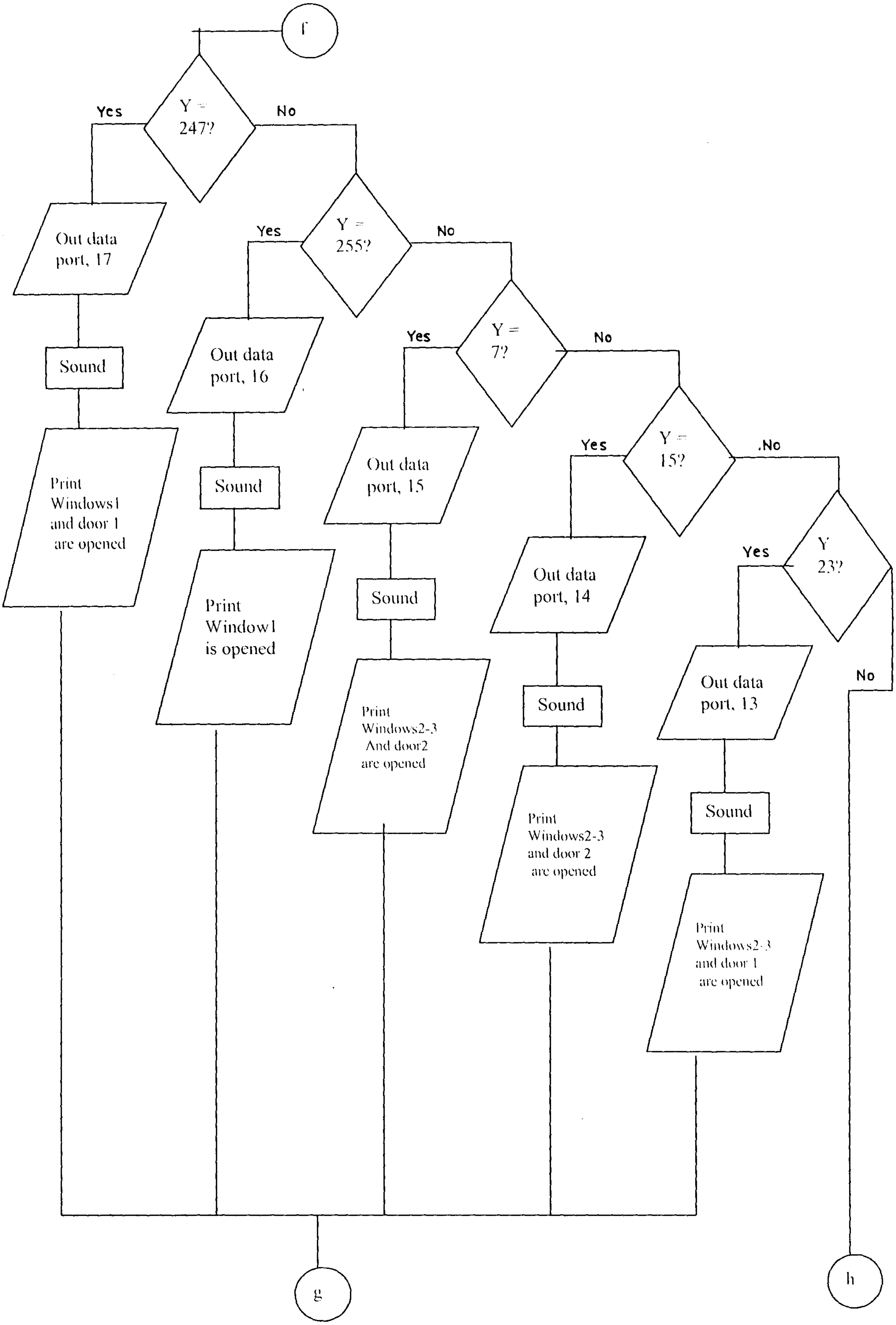
3.2 PROGRAM FLOWCHART

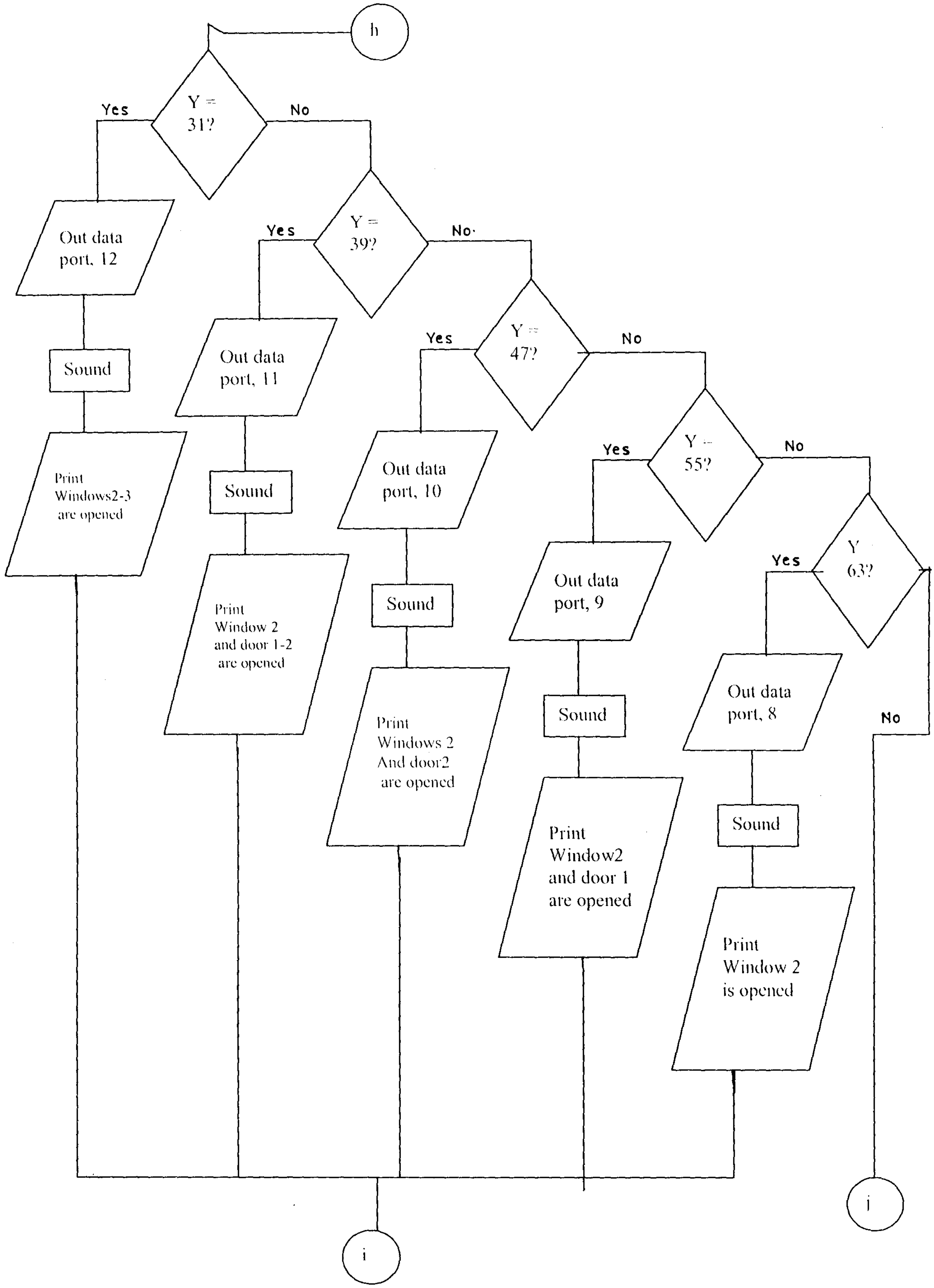


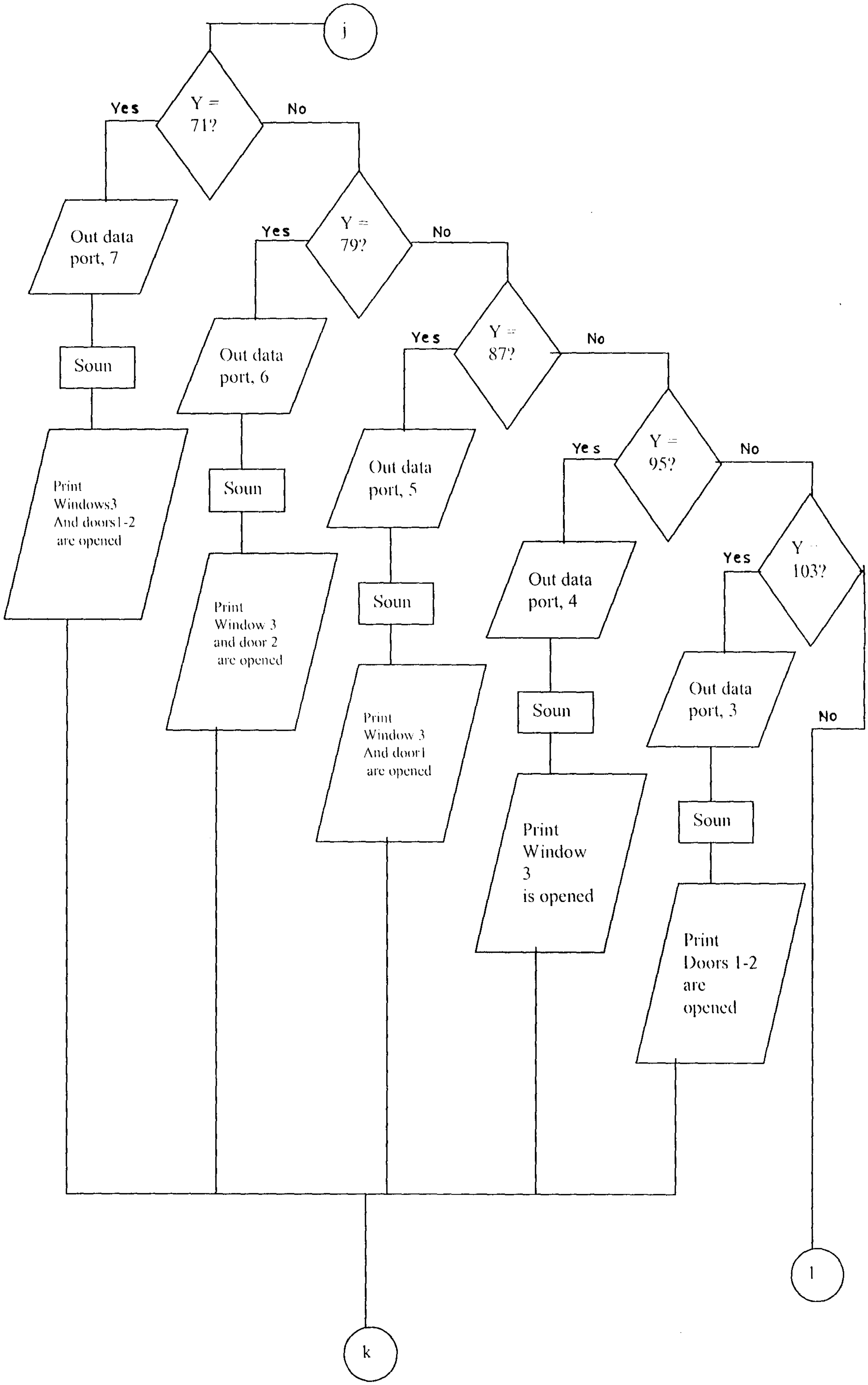
Start
Declare variable
n
Limits excee d?
Input y
Y y or 7
Port initialization
In keys =char (27)?
End
Y = 127?
Print all windows and doors Are closed
Yes
Y = 135?
No
Out data port, 31
Sound
Print all windows and doors are opened
Yes
Y = 143?
No
Out data port, 30
Sound
Print all windows and doors 2 are opened
Yes
Y = 151?
No
Out data port, 29
Sound
Print all windows and doors 1 are opened
Yes
Y = 159?
No
B
Out data port, 28
Sound
Print all windows are opened
A

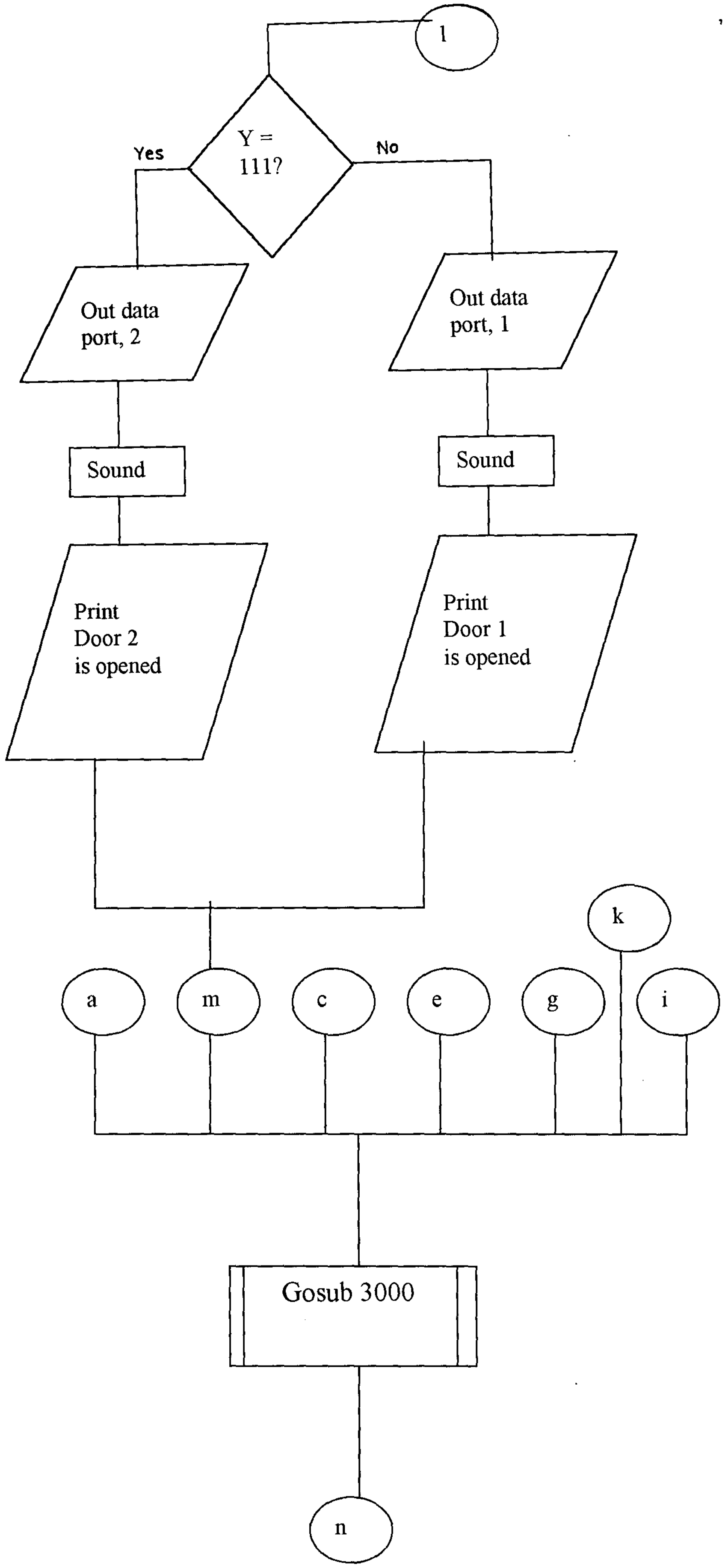

B
Y = 167?
Yes
No
Out data port, 27
Sound
Print Windows1-2 And doors1-2 are opened
Y = 175?
Yes
No
Out data port, 26
Sound
Print Windows1-2 And doors2 Are opened
Y = 183?
Yes
No
Out data port, 25
Sound
Print Windows1-2 And doors1 Are opened
Y = 191?
Yes
No
Out data port, 24
Sound
Print Windows1-2 Are opened
Y = 199?
Yes
No
Out data port, 23
Sound
Print Windows1-3 And doors1-2 Are opened
d
c

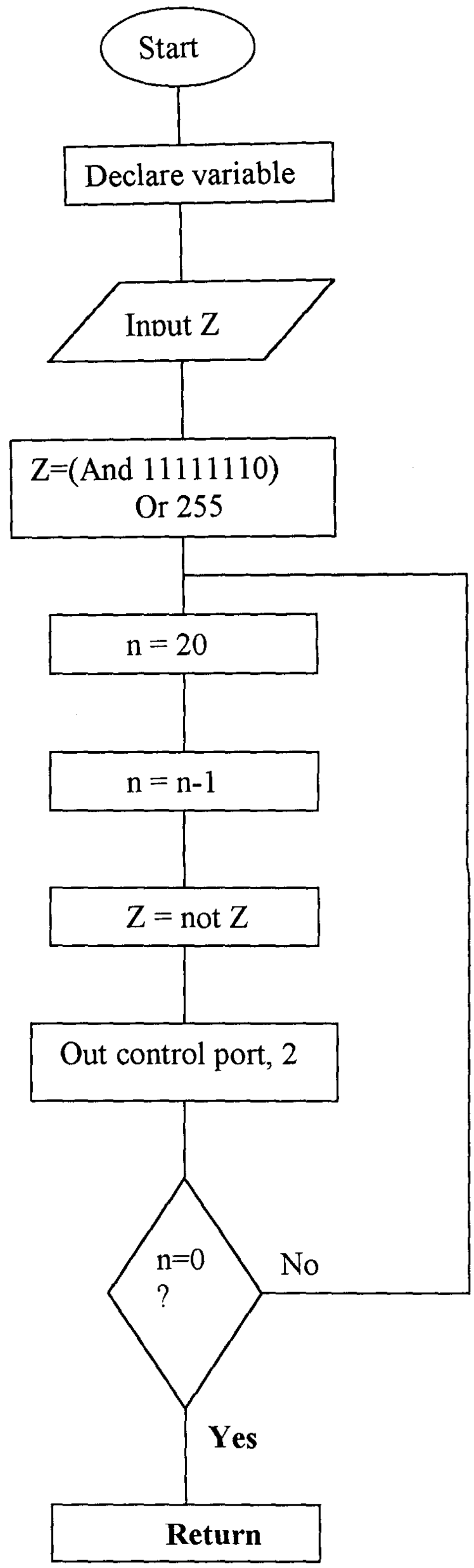

d
Y = 207?
Yes
No
Out data port, 22
Sound
Print Windows1-3 And door 2 Are opened
Y = 215?
Yes
No
Out data port, 21
Sound
Print Windows1-3 And door 1 Are opened
Y = 223?
Yes
No
Out data port, 20
Sound
Print Windows1-3 Are opened
Y = 231?
Yes
No
Out data port, 19
Sound
PrinWindow1 And doors1-2 are opened
Y 239?
Yes
No
Out data port, 18
Sound
Print Window1 and door 2 are opened
e
f


f
Y = 247?
Yes
No
Out data port, 17
Sound
Print Windows1 and door 1 are opened
Y = 255?
Yes
No
Out data port, 16
Sound
Print Window1 is opened
Y = 7?
Yes
No
Out data port, 15
Sound
Print Windows2-3 And door2 are opened
Y = 15?
Yes
No
Out data port, 14
Sound
Print Windows2-3 and door 2 are opened
Y 23?
Yes
No
Out data port, 13
Sound
Print Windows2-3 and door 1 are opened
g
h


h
Y = 31?
Yes
No
Out data port, 12
Sound
Print Windows2-3 are opened
Y = 39?
Yes
No
Out data port, 11
Sound
Print Window 2 and door 1-2 are opened
Y = 47?
Yes
No
Out data port, 10
Sound
Print Windows 2 And door2 are opened
Y = 55?
Yes
No
Out data port, 9
Sound
Print Window2 and door 1 are opened
Y 63?
Yes
No
Out data port, 8
Sound
Print Window 2 is opened
i
j


j
Yes
Y = 71?
No
Out data port, 7
Soun
Print Windows3 And doors1-2 are opened
Yes
Y = 79?
No
Out data port, 6
Soun
Print Window 3 and door 2 are opened
Yes
Y = 87?
No
Out data port, 5
Soun
Print Window 3 And door1 are opened
Ye s
Y = 95?
No
Out data port, 4
Soun
Print Window 3 is opened
Yes
Y = 103?
No
Out data port, 3
Soun
Print Doors 1-2 are opened
k
l


l
Y = 111?
Yes
No
Out data port, 2
Out data port, 1
Sound
Sound
Print Door 2 is opened
Print Door 1 is opened
k
a
m
c
e
g
i
Gosub 3000
n


Start
Declare variable
Input Z
Z=(And 11111110) Or 255
n = 20
n = n-1
Z = not Z
Out control port, 2
n=0 ?
No
Yes
Return

3.3 PROGRAM DESIGN

Declare port address

For k=1to 5

Repeat

Read status ports and put it in y

Set the three least significant bits of the status bits to 1

Initialize data port

Stop the program when escape is pressed

If y=127 then PRINT all windows and doors are closed

Loop until y<>of 127

Case y of

When y=135

Write to data port: 31

PRINT all windows and doors are opened

Make sound

Call of subroutine

When y=143

Write to data port: 30

PRINT all windows and door 2 are opened

Make sound

Call subroutine

When y=151

Write to data port: 29

PRINT all windows and door 1 are opened

Make sound

Call subroutine

When y=159

Write to data port: 28

PRINT all windows are opened

Make sound

Call subroutine

When y=167

Write to data port: 27

Print windows1-2 and doors 1-2 are opened

Make sound

Call subroutine

When y=175

Write to data port: 26

PRINT windows 1-2 and door 2 are opened

Make sound

Call subroutine

When y=183

Write to data port: 25

PRINT windows 1-2 and door 1 are opened

Make sound

Call subroutine

When y=191

Write to data port: 24

PRINT windows 1-2 are opened

Make sound

Call subroutine

When y=199

Write to data port: 23

PRINT windows 1-3 and doors 1-2 are opened

Make sound

Call subroutine

When y=207

Write to data port: 22

PRINT windows 1-3 and door 2 are opened

Make sound

Call subroutine

When y=215

Write to data port: 21

PRINT windows 1-3 and door 1 are opened

Make sound

Call subroutine

When y= 223

Write to data port: 20

PRINT windows 1-3 are opened

Make sound

Call subroutine

When y=231

Write to data port: 19

PRINT window 1 and doors 1-2 are opened

Make sound

Call subroutine

When y= 239

Write to data port 18

PRINT window 1 and door 2 are opened

Make sound

Call subroutine

When y=247

Write to data port: 17

PRINT window 1 and door 1 are opened

Make sound

Call subroutine

When y=255

Write to data port: 16

PRINT window 1 is opened

Make sound

Call subroutine

When y=7

Write to data port: 15

PRINT windows 2-3 and doors 1- 2 are opened

Make sound

Call subroutine

When y=15

Write to data port: 14

PRINT windows 2-3 and door 2 are opened

Make sound

Call subroutine

When y=23

Write to data port: 13

PRINT windows 2-3 and door 1 are opened

Make sound

Call subroutine

When y=31

Write to data port: 12

PRINT window2-3 are opened

Make sound

Call subroutine

When y=39

Write to data port: 11

PRINT window 2 and doors1-2 are opened

Make sound

Call subroutine

When y=47

Write to data port: 10

PRINT window 2 and door 2 are opened

Make sound

Call subroutine

When y=55

Write to data port: 9

PRINT window 2 and door 1 are opened

Make sound

Call subroutine

When y=63

Write to data port: 8

PRINT window 2 is opened

Make sound

Call subroutine

When y=71

Write to data port: 7

PRINT window 3 and doors1-2 are opened

Make sound

Call subroutine

When y=79

Write to data port: 6

PRINT window 3 and door 2 are opened

Make sound

Call subroutine

When y=87

Write to data port: 5

PRINT window 3 and door 1 are opened

Make sound

Call subroutine

Wheny=95

Write to data port:4

PRINT window 3 is opened

Make sound

Call subroutine

Wheny=103

Write to data port: 3

PRINT doors 1-2 are opened

Make sound

Call subroutine

When y=111

Writ to data port: 2

PRINT door 2 is opened

Make sound

Call subroutine

Else

Write to data port :1

PRINT door 1 is opened

Make sound

Call subroutine

Next k

END

3.4 CODING THE PROGRAM

```
10 Rem "program to monitor the device"

20 Let dataport = 888

30 statusport = dataport + 1

35 For k = 1 To 5

40 Do

50 y = INP(statusport)

60 y = y Or 7

70 OUT dataport, 0

75 If INKEY$ = Chr$(27) Then GoTo 1640

79 If y = 127 Then Print "all windows and doors are closed"

80 Loop Until y <> 127

90 If y <> 135 Then GoTo 140

100 OUT dataport, 31

110 Print "all windows and doors are opened"

120 SOUND 800, 364

130 GoSub 3000

140 If y <> 143 Then GoTo 190

150 OUT dataport, 30

160 Print "all windows and door 2 are opened"

170 SOUND 800, 364

180 GoSub 3000

190 If y <> 151 Then GoTo 240

200 OUT dataport, 29

210 Print "all windows and door 1 are opened"

220 SOUND 800, 364
```

```
230 GoSub 3000

240 If y <> 159 Then GoTo 290

250 OUT dataport, 28

260 Print "all windows  are opened"

270 SOUND 800, 364

280 GoSub 3000

290 If y <> 167 Then GoTo 340

300 OUT dataport, 27

310 Print " windows 1-2 and doors 1-2 are opened"

320 SOUND 800, 364

330 GoSub 3000

340 If y <> 175 Then GoTo 390

350 OUT dataport, 26

360 Print " windows 1-2 and door 2 are opened"

370 SOUND 800, 364

380 GoSub 3000

390 If y <> 183 Then GoTo 440

400 OUT dataport, 25

410 Print " windows 1-2 and door 1 are opened"

420 SOUND 800, 364

430 GoSub 3000

440 If y <> 191 Then GoTo 490

450 OUT dataport, 24

460 Print " windows 1-2  are opened"

470 SOUND 800, 364

480 GoSub 3000
```

```
490 If y <> 199 Then GoTo 540

500 OUT dataport, 23

510 Print " windows 1-3 and doors 1-2 are opened"

520 SOUND 800, 364

530 GoSub 3000

540 If y <> 207 Then GoTo 590

550 OUT dataport, 22

560 Print " windows 1-3 and door 2 are opened"

570 SOUND 800, 364

580 GoSub 3000

590 If y <> 215 Then GoTo 640

600 OUT dataport, 21

610 Print " windows 1-3 and doors 1 are opened"

620 SOUND 800, 364

630 GoSub 3000

640 If y <> 223 Then GoTo 690

650 OUT dataport, 20

660 Print " windows 1-3  are opened"

670 SOUND 800, 364

680 GoSub 3000

690 If y <> 231 Then GoTo 740

700 OUT dataport, 19

710 Print " windows 1 and doors 1-2 are opened"

720 SOUND 800, 364

730 GoSub 3000

740 If y <> 239 Then GoTo 790
```

```
750 OUT dataport, 18

760 Print " windows 1 and doors 2 are opened"

770 SOUND 800, 364

780 GoSub 3000

790 If y <> 247 Then GoTo 840

800 OUT dataport, 17

810 Print " windows 1 and door 1 are opened"

820 SOUND 800, 364

830 GoSub 3000

840 If y <> 255 Then GoTo 890

850 OUT dataport, 16

860 Print " windows 1 is opened"

870 SOUND 800, 364

880 GoSub 3000

890 If y <> 7 Then GoTo 940

900 OUT dataport, 15

910 Print " windows 2-3 and doors 1-2 are opened"

920 SOUND 800, 364

930 GoSub 3000

940 If y <> 15 Then GoTo 990

950 OUT dataport, 14

960 Print " windows 2-3 and door 2 are opened"

970 SOUND 800, 364

980 GoSub 3000

990 If y <> 23 Then GoTo 1040

1000 OUT dataport, 13
```

```
1010 Print " windows 2-3 and door 1 are opened"
1020 SOUND 800, 364
1030 GoSub 3000
1040 If y <> 31 Then GoTo 1090
1050 OUT dataport, 12
1060 Print " windows 2-3  are opened"
1070 SOUND 800, 364
1080 GoSub 3000
1090 If y <> 39 Then GoTo 1140
1100 OUT dataport, 11
1110 Print " windows 2 and doors 1-2 are opened"
1120 SOUND 800, 364
1130 GoSub 3000
1140 If y <> 47 Then GoTo 1190
1150 OUT dataport, 10
1160 Print " window 2 and door 2 are opened"
1170 SOUND 800, 364
1180 GoSub 3000
1190 If y <> 55 Then GoTo 1240
1200 OUT dataport, 9
1210 Print " window 2 and door 1 are opened"
1220 SOUND 800, 364
1230 GoSub 3000
1240 If y <> 63 Then GoTo 1290
1250 OUT dataport, 8
1260 Print " window 2 is  opened"
```

```
1270 SOUND 800, 364
1280 GoSub 3000
1290 If y <> 71 Then GoTo 1340
1300 OUT dataport, 7
1310 Print " window 3  and doors 1-2 are opened"
1320 SOUND 800, 364
1330 GoSub 3000
1340 If y <> 79 Then GoTo 1390
1350 OUT dataport, 6
1360 Print " window 3 and door 2 are opened"
1370 SOUND 800, 364
1380 GoSub 3000
1390 If y <> 87 Then GoTo 1440
1400 OUT dataport, 5
1410 Print " window 3 and door 1 are opened"
1420 SOUND 800, 364
1430 GoSub 3000
1440 If y <> 95 Then GoTo 1490
1450 OUT dataport, 4
1460 Print " window 3 is opened"
1470 SOUND 800, 364
1480 GoSub 3000
1490 If y <> 103 Then GoTo 1540
1500 OUT dataport, 3
1510 Print " doors 1-2 are opened"
1520 SOUND 800, 364
```

```
1530 GoSub 3000

1540 If y <> 111 Then GoTo 1590

1550 OUT dataport, 2

1560 Print " door 2 is opened"

1570 SOUND 800, 364

1580 GoSub 3000

1590 OUT dataport, 1

1600 Print "door 1 is opened"

1610 SOUND 800, 364

1620 GoSub 3000

1630 Next k

1640 End

3000 Rem " sound"

3010 Let controlport = 890

3020 z = INP(controlport)

3030 z = (z And 11111110) Or 255

3040 Let n = 20

3050 Do

3060 n = n - 1

3070 z = Not z

3075 OUT controlport, z

3080 Loop Until n = 0

3090 RETURN 1630
```

CHAPTER FOUR

4.0 PROJECT DESIGN

4.1 INTRODUCTION

This chapter deals with the project design which explain how and why components are put together. It encompasses the calculation of currents, voltages and some components value. First we have to look at the block diagram. We notice that it consists of five units. These are display unit, alarm unit, sensing unit and the power unit. The circuit diagram of the system is shown in figure 4.8 at the end of the chapter.

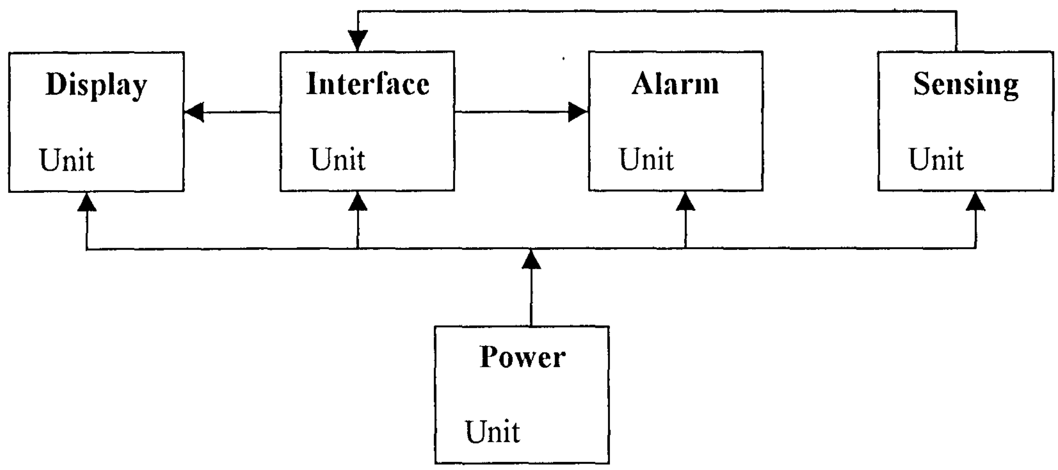


Fig.4.1 block diagram of the circuit

4.2 COMPONENTS AND THEIR FUNCTIONS

4.2.1 POWER UNIT

i) THE TRANSFORMATION STAGE

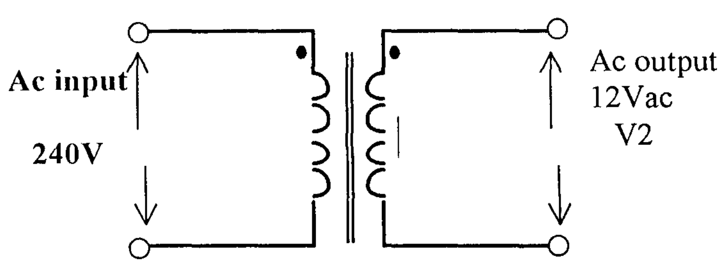


Fig.4.2 transformer diagram

It is a step down transformer that reduces the voltage from 240Vac to 12Vac. The transformer has two windings known as primary and secondary. The primary voltage is labeled as V_1 and is the main outlet voltage. V_2 is the secondary voltage. From the data of the transformer $V_1 = 240$ V and $V_2 = 12$ V and $I_2 = 500$mA.

So, $V_2 = 1/20\ V_1 \Rightarrow V_2/V_1 = 1/20$

The voltage at the primary has a frequency of 50Hz, which is also equal to the secondary frequency.

F1 = F2 = 50Hz

The fundamental relationship that governs transformers is

$V_1/V_2 = N_1/N_2 = I_2/I_1 = n = 20$

Where n is the turn ratio of the transformer N_1 number of turns in the primary winding and N_2, number of turns in the secondary windings.

I2 = 500mA

Thus $I_2/I_1 = 20 \Rightarrow I_2 = 20I_1 \Rightarrow I_1 = I_2/20$

$I_1 = 800\ 10^{-3}/20$

$= 25\ 10^{-3}$A

= 25mA

We also know that the input power equals the output power

$P_1 = P_2$

$P_2 = I_2V_2 = I_1V_1 = 500\ 10^{-3}*12 = 6W$

ii) RECTIFICATION STAGE

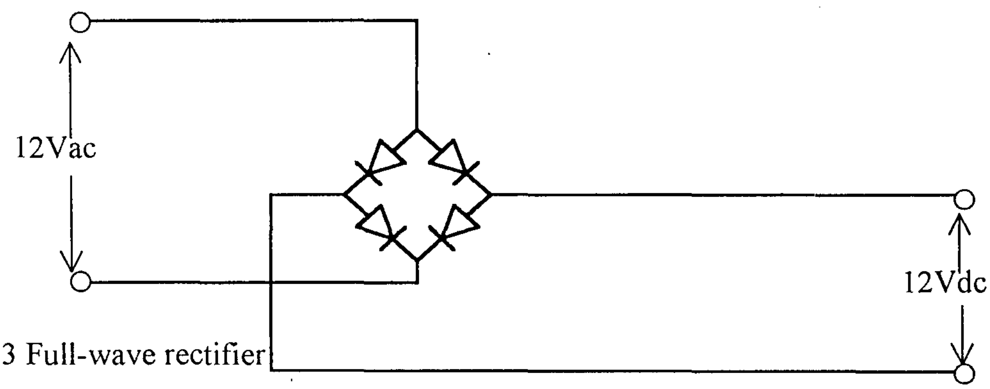


Fig.4.3 Full-wave rectifier

The average and RMS values of the voltage are given by equation:

$Vp = Vrms/\sqrt{2} \Rightarrow 0.7o7Vp = Vrms$

$Vav = 2/11Vp \Rightarrow Vav = 0.637Vp$

$Vav = 0.9Vrms$

Where Vp is the peak voltage [7]

III) FILTERING STAGE

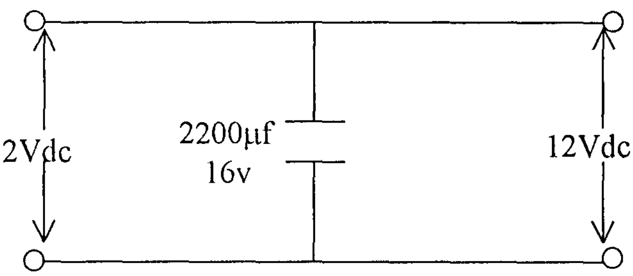


Fig.4.4 Filtering Capacitor

The effectiveness of a capacitive filter is determined by three factors which are related by:

$T = RC$

Where T = time constant in second

R = resistance in Ohms

C = capacitance in Farad

The capacitance is calculated by

$C = I/Vp\text{-}p * T$ [5]

Where Vp-p = peak-to-peak ripple in volts

$C = 500\ 10^{-3}/24 * 1 / 50$

$=416.66\ \mu F$

But we choose 2200µF because the more the capacitance is high more the rectification is good.

IV. REGULATION STAGE

A 7805 voltage regulator does the regulation. Its rating from ECG is:

Vout = 5v

Iout = 0.1A

Vin max = 30v

Vin min = 7v

Po = 0.7W

This voltage regulator gives the voltage seen by the load and holds the voltage at 5v.

4.2.2 DISPLAY UNIT

This unit is consisted of :

i) THE ASTABLE MULTIVIBRATOR

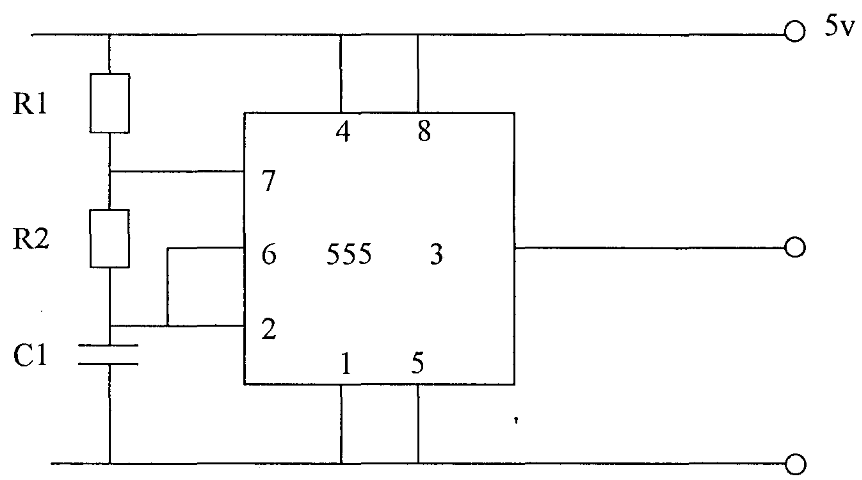


Fig.4.5 astable vibrator circuit to generate square wave form

The frequency of the ripple is

F =0.72 /R2 C1 as R2 >> R1

= 0.72 / 100 10-3 * 10 10^{-6} = 0.72Hz

The mark M = 0.693 (R1+R2) C1

= 0.693 (10^3+100 10^3) 10 10^{-3} = 0.699

The space S = 0.693 (R2)C1

= 0.693 (100 10^3) 10 10^{-6}

= 0.693

ii) THE LED DISPLAY

They are used to identify which of the entrance or windows are opened. A resistor protects them. Knowing that a standard LED requires a typical value of I_f of 10mA, V_f of 2V and Vsat of the transistor is 0.2v.

Ip = (Vcc-Vp-Vsat) / R ⇒ R = (Vcc-Vf-Vsat) / Ip.

R = (5-2-0.2) / 0.01

= 280Ω

In practice we choose a 220Ω resistor.

iii) THE TRANSISTORS

They are used as switch in the circuit and their ratings are:

V_{CBO} =100V

V_{CEO} = 80v

V_{EBO} = 7v

I_{max} = 1A

P_{max} = 1W

Frequency = 100MHz

Current gain = h_{FE}= 100min

A protecting resistor is used to secure the base

R_{Bmax} =0.5 B_{min}Rc = 0.3 * 100 * 220 = 11000Ω

4.2.3 ALARM UNIT

This unit is responsible of the sound that calls the attention of the operator.

It consists of:

i) MONOSTABLE VIBRATOR

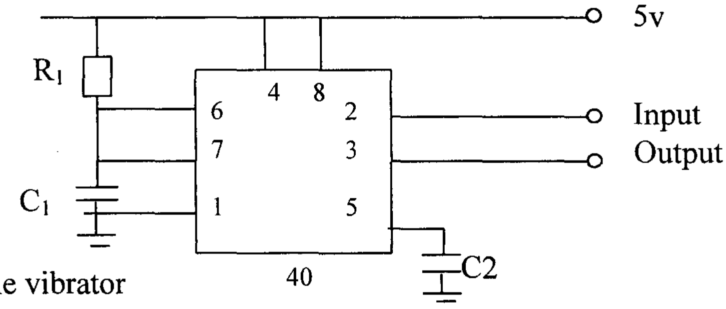


Fig.4.6 monostable vibrator

This device is drive by a square wave signal generated by the computer. The duration of the output pulse is

T = 1.1 (R1C1)

2 = 1.1 ($10\ 10^3$ * C1)

2 = $11 10^3$C1

$\Rightarrow$ C1 = 2/11 10^3

= 181.81μF

We choose the standard value of 220μF

ii) SPEAKER

Resistance = 4Ω

Power = 3W

4.2.4 SENSING UNIT

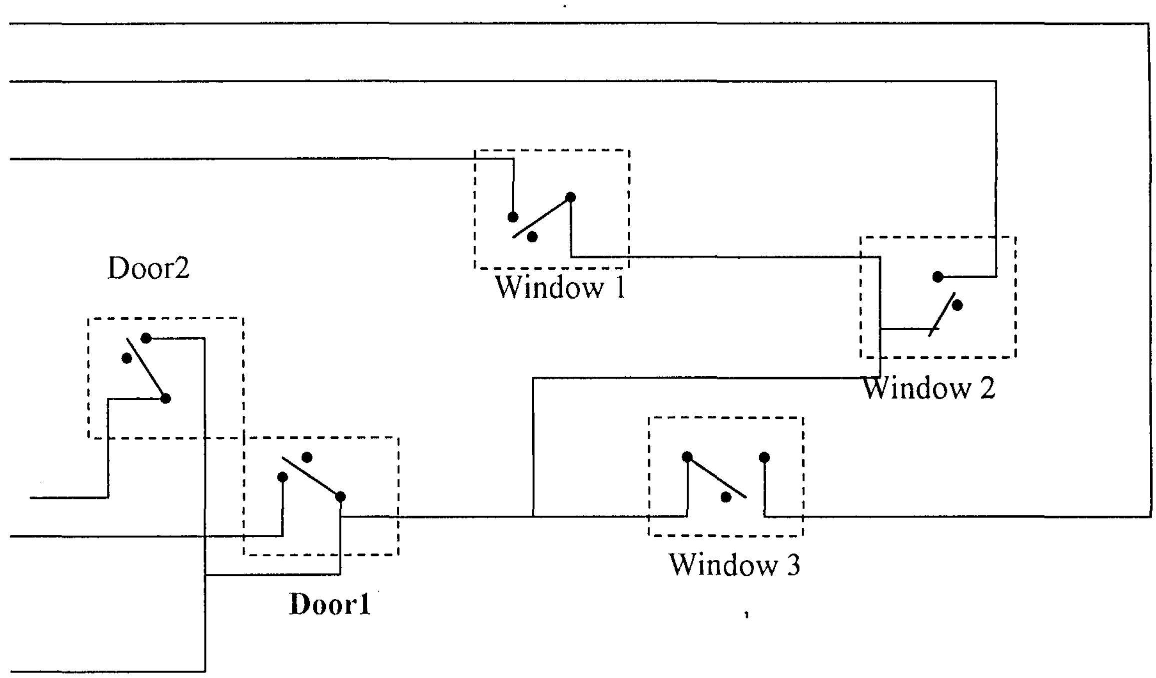


Fig. 4.7 switch position on windows and doors.

It consists of switches that close when a window or a door is opened. This will pull the connected line to low and the computer set the display and the alarm unit.

2.5 INTERFACE UNIT

It consists of two octal buffer ICs, 74LS 244 which are used to protect the parallel port from being damaged by the circuit. When enable the input and output lines at the same level.

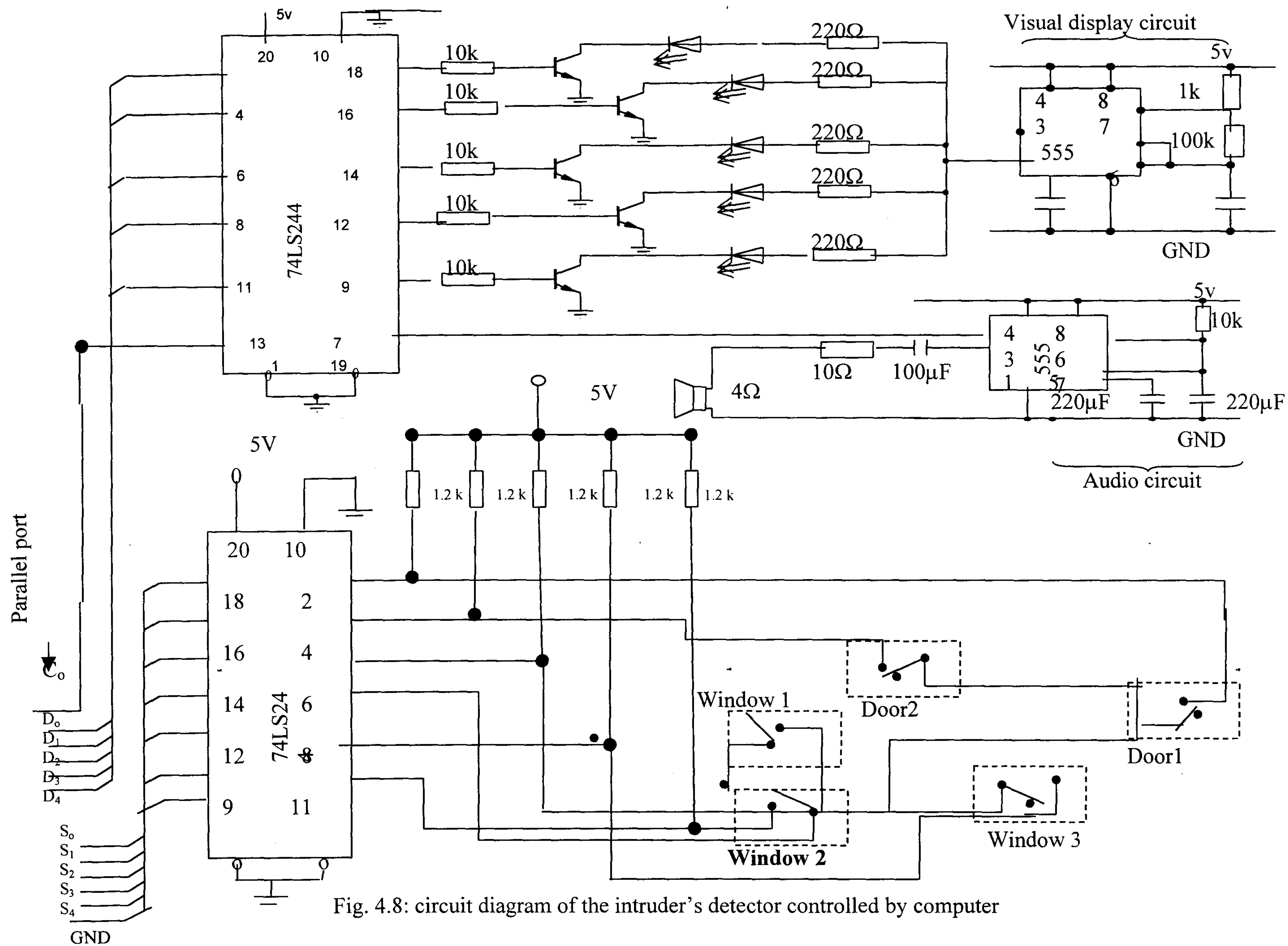


Fig. 4.8: circuit diagram of the intruder's detector controlled by computer

CHAPTER FIVE

5.0 PROJECT CONSTRUCTION AND TESTING

5.1 CONSTRUCTION OF THE POWER

In this subtitle we are going to show how the different parts of the power unit are brought together to make the conversion from the ac to dc voltage usable by the device.

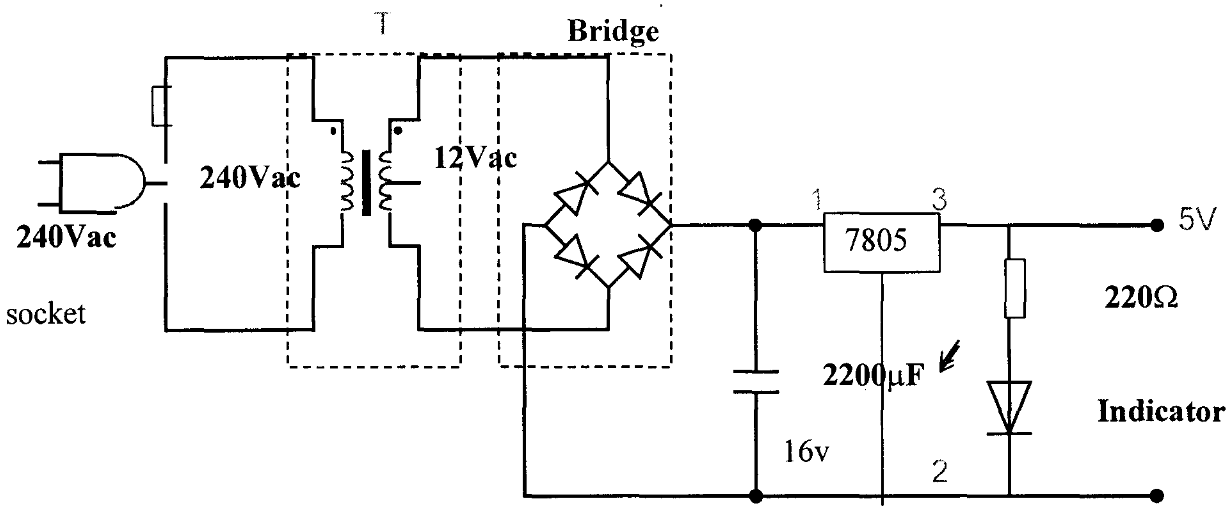


Fig 5-1 power unit

5.2 CONSTRUCTION OF THE DISPLAY UNIT

The astable multivitbrator is supplying current to LEDs, which remain off until the transistor, which is acting as a switch is on.

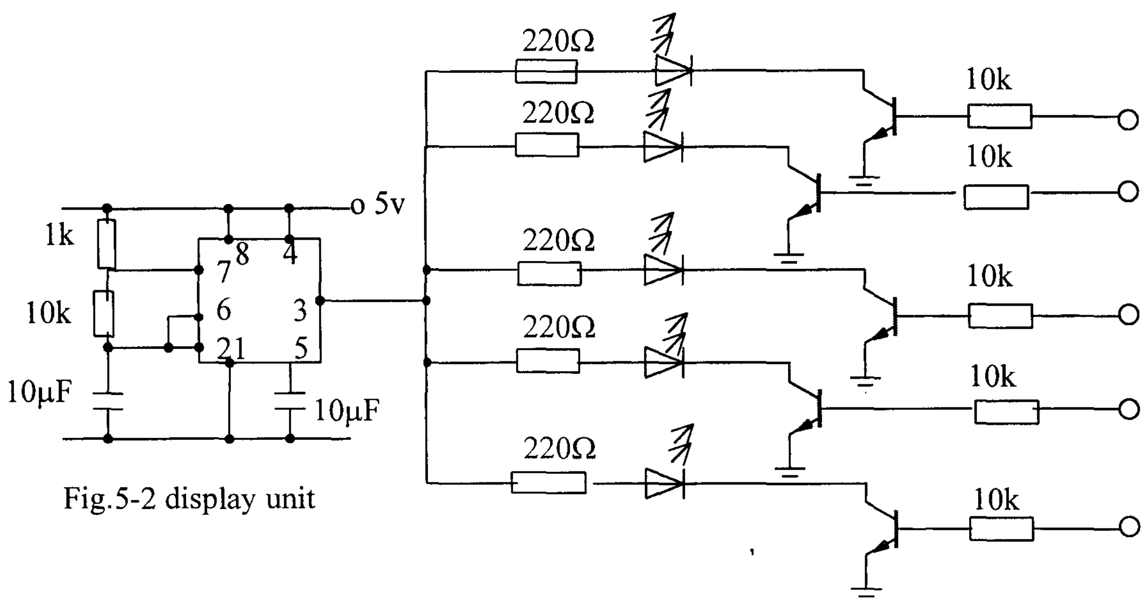


Fig.5-2 display unit

5.3 CONSTRUCTION OF THE ALARM UNIT

The monostable vibrator is brought together with a filtering capacitor and a loudspeaker to produce the sound.

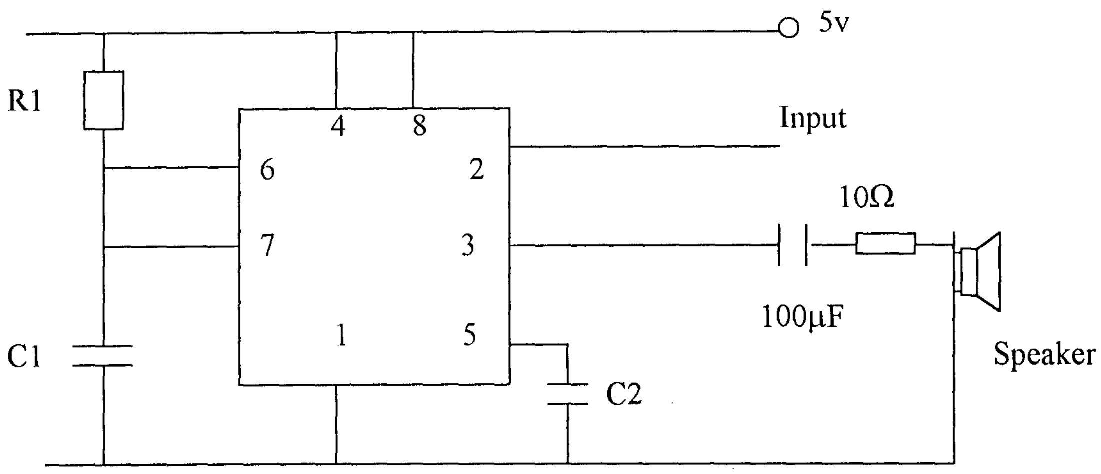


Fig 5-3 the alarm unit

5.4 TESTING THE CONSTRUCTION DEVICE

5.4.1 STAGE OF TESTING

After soldering the component in the PCB, to make the power unit, I measure the output of my transformer, then the voltage across the indicator. This was done after some troubleshooting. I made the display unit, here I used my 5 V supply to activate the transistors through the buffer. The Alarm unit is then set. I used my astable multivibrator to test it. All these work properly individually after some troubleshooting.

We coupled all the parts together to form the device and we tested it on bench by supplying current to the input lines and output lines too.'Then we went on computer for the last testing. Some connections have being made to the program before it works.

5.4.2 RESULTS

The output of the overall system is shown in the picture of the following pages. It shows the flashing LEDS that represent a door while green or a window while red then the computer screen, which shows the states of the doors and windows.

The results are in accordance with the design and construction required. The above are recorded while testing by opening some windows and doors.

So as soon as an intruder opens a door or a window the system alarms and the LEDs flash, the computer continues to scan the input to see if another door or window is not opened and this is done five times before the program stops. If any of the door and windows is opened the program is stopped by pressing the Escape key on the keyboard.

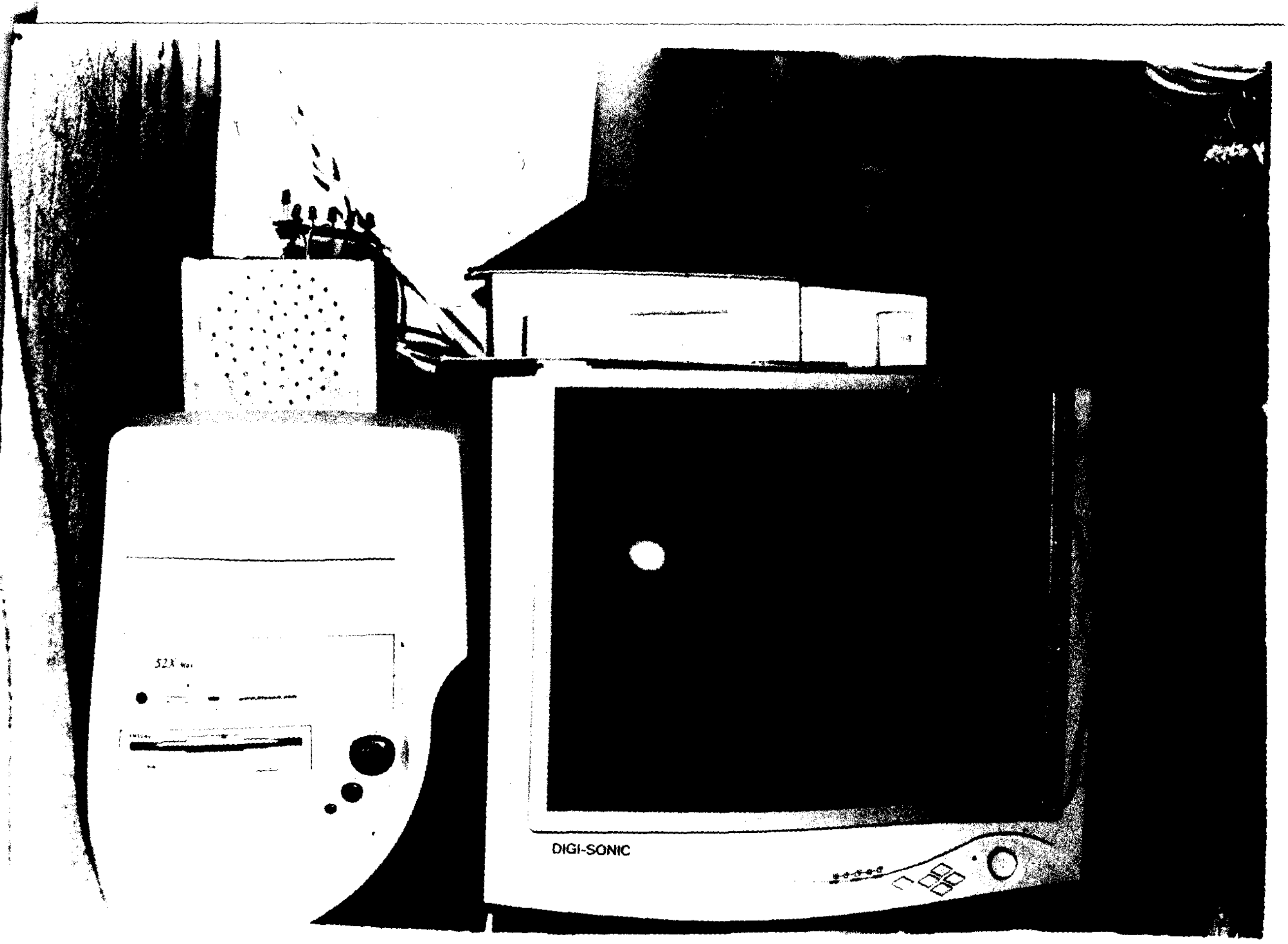
5.4.3 RESULTS ANALYSIS.

At the light of the above we can say that the objective of the designed is achieved. This was done by overcoming some difficulties such as:

1. Debugging of the program
2. The fixing of the switches at suitable places to make them more sensible.
3. The difference between the status register and the input lines state
4. To establish a good compromise between the hardware and software.

Now we should ask ourselves if the technology used in this project is efficient enough to be used in our under developing countries homes?

We can easily answer by yes because in this work we consider the cost, the availability of materials used. Also the maintainability of the system is to easy and will be not time consuming. This project will also allow the use of computer more effectively towards using its huge capabilities.


DIGI-SONIC

windows 2 and doors 1-2 are opened

window 2 and door 2 are opened

window 2 and door 1 are opened

window 2 is opened

window 3 and doors 1-2 are opened

window 3 and door 2 are opened

window 3 and door 1 are opened

window 3 is opened

doors 1-2 are opened

all windows and doors are closed

door 2 is opened

windows 1-3 and door 2 are opened

windows 2 and doors 1-2 are opened

window 2 and door 2 are opened

window 2 and door 1 are opened

window 2 is opened

window 3 and doors 1-2 are opened

window 3 and door 2 are opened

window 3 and door 1 are opened

window 3 is opened

doors 1-2 are opened

all windows and doors are closed

door 2 is opened

CHAPTER SIX

6.0 CONCLUSION AND RECOMMENDATIONS

6.1 CONCLUSION

From the results obtain after construction and testing the device, we can see that the computer was able to show you which of the entrance is opened or closed. This shows how computer are used to watch your home from any intruder as stated by the project topic " computer aided security." We easily see that the result meet the aim of the project. The appropriate settings of the switches will make the system more efficient. The availability of the components used and the low cost of these should attract more Engineers that have the responsibility for bring solutions to people's problems.

6.2 RECOMMENDATIONS

The system we presents can easily be expanded to control fire, gas, temperature etc... It can also be used to switch devices such as light, fan, air conditioner just by choosing the appropriate sensing device combine with relays or timer. So I recommend that this project should be developed with the maximum of its capabilities to insure full security in our homes and offices.

REFERENCES

[1] Holmes B.J, Basic programming", ELBS edition, 3rd ed., 1989

[2] William B., PC Interfacing, communications and windows Programming, Addison – Wesley, 1999

[3] Mustafa A.M., Microcomputer interfacing and applications, Newnes, 2nd ed., 1994

[4] Stuart B., Real-time computer control, prentice Hall, 1988

[5] Schuler, Charles A., Electronics: principles and applications, international Edition, 4th ed., 1994

[6] Paul H., Winfield H., Arts of electronics, Cambridge University Press, 2nd ed., 1989.

[7] Djibrilla M., Design and construction of an automatic Door control, unpublished undergraduate Thesis, 2001

[8] Djedna N., Computer Programming and languages handouts, 1998

[9] Roger L.T., Digital Electronics, Internationals Editions, 4th Ed. 1998