NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

VISUAL BASED TTL GATE TESTER

Graduation Project
COM- 400

Student: Yunus KURTOGLU

Supervisor: Assoc. Prof. Dr. Dogan IBRAHIM

Nicosia - 2001

ACKNOWLEDGMENT

Firstly, I have to send my thanks to my supervisor Assoc.Prof.Dr.Dogan IBRAHIM for his helps.

Secondly, I would like to express my gratitude to Near East University for the scholarship that made the work possible.

Finally, I have to say thanks to my friend Latif NIZIPLI. I designed my hardware with him. And also like to thanks all other my friends for their advice and support.

ABSTRACT

Standard TTL logic gates are very important in logic design circuits. This project is a personal computer based TTL gate tester project. The project consists of a hardware where the gate to be tested is mounted on and connected to the parallel port of a standard personal computer. A Visual Basic program, developed by the author, enables the user to select the gate type and then the gate is tested and user is informed whether or not the gate is faulty. The project can be used to test simple 14-pin 74 series logic gates.

TABLE OF CONTENTS

ACKNOWLEDGMENT i
ABSTRACT ii
TABLE OF CONTENTS iii
LIST OF ABBREVIATIONS v
INTRODUCTION 1
Chapter 1 USING THE PARALLELPORTS 3
1.1. Defining the Port 3
1.2. Port Types 4
1.2.1.Original (SPP) 4
1.2.2. PS/2-type (Simple Bidirectional) 5
1.2.3. Epp 5
1.2.4. Ecp 5
1.2.5. Multi-mode Ports 6
1.3. System Resources 6
1.4. Interrupts 7
1.5. DMA Channels 8
1.6. Finding Existing Ports 8
1.7. Configuring 8
1.8. Port Options 8
1.9. Multi-mode Ports 9
1.10. Adding a Port 12
1.11. Port Hardware 13
1.12. Basic Concepts of Computer Cabling 13
1.13. IBM Parallel port Pin Descriptions 13
1.14. Parallel Port Laplink Cable Pinout 14
1.15. IBM DB-25 numbering scheme. (Pin numbers) 16
Chapter 2 INTRODUCTION TO TTL GATES 17
2.1. 7400 Devices 17
2.2. Propagation Delay Time and Power Dissipation 17
2.3. 5400 Series 17
2.4. High Speed TTL 18
2.5. Low Power TTL 18
2.6. Schottky TTL 18
2.7. Low Power Schottky TTL 19
2.8. TTL Characteristics 19
2.8.1. Worst-Case input Voltage 19
2.8.2. Worst-Case Output Voltages 20
2.8.3. Standard Loading 20
2.9. Definitions of Device Terminology 21
2.10. TTL Logic Gate Devices 24
2.11. Comparison of Characteristics 25
2.12. SN74LS00, Quad 2-input positive-NAND gates 25
2.13. SN74LS03, Quad 2-input positive-NAND gates with open collector outputs 26
2.14. SN74LS08, Quadruple 2-Input Positive-AND Gates 27
2.15. SN74LS09, Quad 2-input positive-AND gates with open collector outputs 28
2.16. SN74LS26, Quad 2-input high-voltage interface positive-NAND gates 28
2.17. SN74LS32, Quad 2-input positive-OR gates 29
2.18. SN74LS37, Quad 2-input positive-NAND buffers 30

2.19. SN74LS38, Quad 2-input positive-NAND buffers with open collector 30
2.20. Structure of Some Common TTL Gates 32
Chapter 3 DESCRIPTION OF THE PROGRAM 33
3.1. Main Program 33
3.2. How to use the program 35
3.3. About the program 35
3.3.1. Details the program 36
3.4. Exit the program 37
3.5. Port tester 37
3.6. Modules 38
Chapter 4 THE HARDWARE 39
4.1. Hardware Description 39
4.2. Connection of Hardware 40
Chapter 5 THE SOFTWARE 42

CONCLUSION 43
REFERENCES 44
APPENDIX A 45
APPENDIX B 98
APPENDIX C 99
APPENDIX D 101
APPENDIX E 102
APPENDIX F 106

LIST OF ABBREVIATIONS

None: Standard TTL device

Abt: Advanced bus interface logic

Ac: Advanced CMOS

Act: Advanced CMOS with TTL compatible inputs

Actq: Advanced CMOS with TTL compatible inputs

Als: Advanced low-power Schottky

As: Advanced Schottky TTL

Bct: Octal buffers/drivers

C: CMOS version of a TTL device

Cbt: Crossbar technology

F: 'Fast' ± a high speed version of the device

Fct: High speed high output

Gtl: Gunning transceiver logic

H: High speed version

Hc: High speed CMOS version (CMOS compatible inputs)

Hct: High speed CMOS version (TTL compatible inputs)

H.v.: high voltage

Ls: Low-power Schottky

Lv: Low voltage high speed CMOS

Lvc: Low voltage CMOS

Lvt: Low voltage technology

O.c.: open collector

S: Schottky input configuration (improved speed and noise immunity)

Sch: Schmitt

Chapter two describes the types of TTL gates, their properties, description of all the information of the TTL gates used in the project.

Chapter three describes the usage of the program, how to start, how to test the TTL gates using the program given in this project. Also it gives the important information about the cautions necessary in order to test the TTL gates while using this program.

Chapter four gives description of the hardware which we used in our project. Also, it describes the things essential for the using this hardware, how to the interface the hardware and software.

Chapter five describes the processes used for testing of the TTL gates in terms of the flow charts.

Chapter 1 USING THE PARALLELPORTS

1.1. Defining the Port

What is the "parallel port"? In the computer world, a port is a set of signal lines that the microprocessor, or CPU, uses to exchange data with other components. Typical uses for ports are communicating with printers, modems, keyboards, and displays, or just about any component or device except system memory. Most computer ports are digital, where each signal, or bit, is 0 or 1. A parallel port transfers multiple bits at once, while a serial port transfers a bit at a time (though it may transfer in both directions at once). There are other ports such as SCSI, USB, and IrDA, but the parallel port remains popular because it's capable, flexible, and every PC has one. The term PC-compatible, or PC for short, refers to the IBM PC and any of the many, many personal computers derived from it. From another angle, a PC is any computer that can run Microsoft's MS-DOS or WINDOWS operating system and whose expansion bus is compatible with the ISA or PCI bus in the original IBM PC. The category includes the PC, XT, AT, PS/2, and most computers with 80x86, Pentium, and compatible CPUs. It does not include the Macintosh, Amiga, or IBM mainframes, though these and other computer types may have ports that are similar to the parallel port on the PC.

The original PC's parallel port had eight outputs, five inputs, and four bidirectional lines. These are enough for communicating with many types of peripherals. On many newer PCs, the eight outputs can also serve as inputs, for faster communications with scanners, drives, and other devices that send data to the PC. The parallel port was designed as a printer port, and many of the original names for the port's signals (PaperEnd, AutoLineFeed) reflect that use. But these days, you can find all kinds of things besides printers connected to the port. The term peripheral, or peripheral device is a catch all category that includes printers, scanners, modems, and other devices that connect to a PC.

1.2. Port Types

As the design of the PC evolved, several manufacturers introduced improved versions of the parallel port. The new port types are compatible with the original design, but add new abilities, mainly for increased speed. Speed is important because as computers and peripherals have gotten faster, the jobs they do have become more complicated, and the amount of information they need to exchange has increased. The original parallel port was plenty fast enough for sending bytes representing ASCII text characters to a dot matrix or daisy wheel printer. But modern printers need to receive much more information to print a page with multiple fonts and detailed graphics, often in color. The faster the computer can transmit the information, the faster the printer can begin processing and printing the result.

A fast interface also makes it feasible to use portable, external versions of peripherals that you would otherwise have to install inside the computer. A parallel port tape or disk drive is easy to move from system to system, and for occasional use, such as making backups, you can use one unit for several systems. Because a backup may involve copying hundreds of Megabytes, the interface has to be fast to be worthwhile;

1.2.1.Original (SPP)

The parallel port in the original IBM PC, and any port that emulates the original port's design, is sometimes called the SPP, for standard parallel port, even though the original port had no written standard beyond the schematic diagrams and documentation for the IBM PC. Other names used are AT-type or ISA-compatible. The port in the original PC was based on an existing Centronics printer interface.

However, the PC introduced a few differences, which other systems have continued. SPPs can transfer eight bits at once to a peripheral, using a protocol similar to that used by the original Centronics interface. The SPP doesn't have a byte wide input port, but for PC-to-peripheral transfers, SPPs can use a Nibble mode that transfers each byte 4 bits at a time. Nibble mode is slow, but has become popular as a way to use the parallel port for input.

1.2.5. Multi-mode Ports

Many newer ports are multi-mode ports that can emulate some or all of the above types. They often include configuration options that can make all of the port types available, or allow certain modes while locking out the others.

1.3. System Resources

The parallel port uses a variety of the computer's resources. Every port uses a range of addresses, though the number and location of addresses varies. Many ports have an assigned IRQ (interrupt request) level, and ECPs may have an assigned DMA channel. The resources assigned to a port can't conflict with those used by other system components, including other parallel ports.

Addressing ;

The standard parallel port uses three contiguous addresses, usually in one of these ranges:

Table 1.1 Addresses of Parallel port

Base0	Base1	Base2
378h	278h	3BCh
379h	279h	3BDh
37Ah	27Ah	3Beh

The first address in the range is the port's base address, also called the Data register or just the port address. The second address is the port's Status register, and the third is the Control register. EPPs and ECPs reserve additional addresses for each port. An EPP adds five registers at base address + 3 through base address + 7, and an ECP adds three registers at base address + 400h through base address + 402h. For a base

address of 378h, the EPP registers are at 37Bh through 37Fh, and the ECP registers are at 778h through 77Fh.

On early PCs, the parallel port had a base address of 3BCh. On newer systems, the parallel port is most often at 378h. But all three addresses are reserved for parallel ports, and if the port's hardware allows it, you can configure a port at any of the addresses. However, we normally can't have an EPP at base address 3BCh, because the added EPP registers at this address may be used by the video display. IBM's Type 3 PS/2 port also had three additional registers, at base address +3 through base address + 5, and allowed a base address of 1278h or 1378h. Most often, DOS and Windows refer to the first port in numerical order as LPT1, the second, LPT2, and the third, LPT3. So on bootup, LPT1 is most often at 378h, but it may be at any of the three addresses. LPT2, if it exists, may be at 378h or 278h, and LPT3 can only be at 278h. Various configuration techniques can change these assignments, however, so not all systems will follow this convention. LPT stands for line printer, reflecting the port's original intended use. If your port's hardware allows it, you can add a port at any unused port address in the system. Not all software will recognise these non-standard ports as LPT ports, but you can access them with software that writes directly to the port registers.

1.4. Interrupts

Most parallel ports are capable of detecting interrupt signals from a peripheral. The peripheral may use an interrupt to announce that it's ready to receive a byte, or that it has a byte to send. To use interrupts, a parallel port must have an assigned interrupt request level (IRQ).

Conventionally, LPT1 uses IRQ7 and LPT2 uses IRQ5. But IRQ5 is used by many sound cards, and because free IRQ levels can be scarce on a system, even IRQ7 may be reserved by another device. Some ports allow choosing other IRQ levels besides these two.

Many printer drivers and many other applications and drivers that access the parallel port don't require parallel-port interrupts. If you select no IRQ level for a port,

the port will still work in most cases, though sometimes not as efficiently, and you can use the IRQ level for something else.

1.5. DMA Channels

ECPs can use direct memory access (DMA) for data transfers at the parallel port. During the DMA transfers, the CPU is free to do other things, so DMA transfers can result in faster performance overall. In order to use DMA, the port must have an assigned DMA channel, in the range 0 to 3.

1.6. Finding Existing Ports

DOS and Windows include utilities for finding existing ports and examining other system resources. In Windows 95, click on Control Panel, System, Devices, Ports, and click on a port to see its assigned address and (optional) IRQ level and DMA channel. In Windows 3.1 or DOS, you can use Microsoft's Diagnostic (msd.exe) to view ports, assigned IRQ levels, and other system details.

1.7. Configuring

The parallel port that comes with a PC will have an assigned address and possibly an IRQ level and DMA channel. Multi-mode ports may also be configured with specific modes enabled. You can change some or all of these assignments to match your needs. If you're adding a new port, you need to configure it, making sure that it doesn't conflict with existing ports and other resources.

1.8. Port Options

There is no standard method for configuring a port. Some ports, especially older ones, use jumper blocks or switches to select different options. Others allow configuring in software, using a utility provided on disk. A port on a system motherboard may have configuration options in the system setup screens (the CMOS setup) that you can access on bootup. On ports that meet Microsoft's Plug and Play standard, Windows 95 can automatically assign an available port address and IRQ level to a port.

Check your system or port's documentation for specifics on how to configure a port. Some ports allow a choice of just one or two of the three conventional base addresses. A few allow you to choose any uncommitted address, including non-standard ones. On some boards, the jumpers or switches are labeled, which is extremely handy when you don't have other documentation (or can't find it). If your port supports ECP transfers, assign it an IRQ level and DMA channel if possible. Most ECP drivers do use these, and if they're not available, the driver will revert to a slower mode.

1.9. Multi-mode Ports

Configuring a multi-mode port needs special consideration. A multi-mode port's controller chip supports a variety of modes that emulate different port types. In addition to the configuration options described above, on most multi-mode ports, you also have to select a port type to emulate.

The problem is that there is no single standard for the basic setup on the controller chips, and there are many different chips! Usually the setup involves writing to configuration registers in the chip, but the location and means of accessing the registers varies.

For this reason, every port should come with a simple way to configure the port. If the port is on the motherboard, look in the CMOS setup screens that you can access on bootup. Other ports may use jumpers to enable the modes, or have configuration software on disk.

The provided setup routines don't always offer all of the available options or explain the meaning of each option clearly. For example, one CMOS setup I've seen allows only the choice of AT or PS/2-type port. The PS/2 option actually configures the port as an ECP, with the ECP's PS/2 mode selected, but there is no documentation explaining this. The only way to find out what mode is actually selected is to read the chip's configuration registers. And although the port also supports EPP, the CMOS setup includes no way to enable it, so again, accessing the configuration registers is the only option.

If your port is EPP or ECP capable but the setup utility doesn't offer these as choices, a last resort is to identify the controller chip, obtain and study its data sheet, and write your own program to configure the port. The exact terminology and the number of available options can vary, but these are typical configuration options:

Multi-Mode Port

SPP : Emulates the original port. Also called AT-type or ISA-compatible.

PS/2 : Complex or simple bidirectional. Like an SPP, except that the data port is bidirectional.

EPP : Can do EPP transfers. Also emulates an SPP. Some EPPs can emulate a PS/2-type port.

ECP : Can do ECP transfers. The ECP's internal modes enable the port to emulate an SPP or PS/2-type port. An additional internal mode, Fast Centronics, or Parallel Port FIFO, uses the ECP's buffer for faster data transfers with many old style (SPP) peripherals.

ECP + EPP : An ECP that supports the ECP's internal mode 100, which emulates an EPP. The most flexible port type, because it can emulate all of the others.

Drivers :

After setting up the port's hardware, you may need to configure your operating system and applications to use the new port.

For DOS and Windows 3.1 systems, on bootup the operating system looks for ports at the three conventional addresses and assigns each an LPT number. In Windows 3.1, to assign a printer to an LPT port, click on Control Panel, then If the printer model isn't displayed, click Add and follow the prompts.

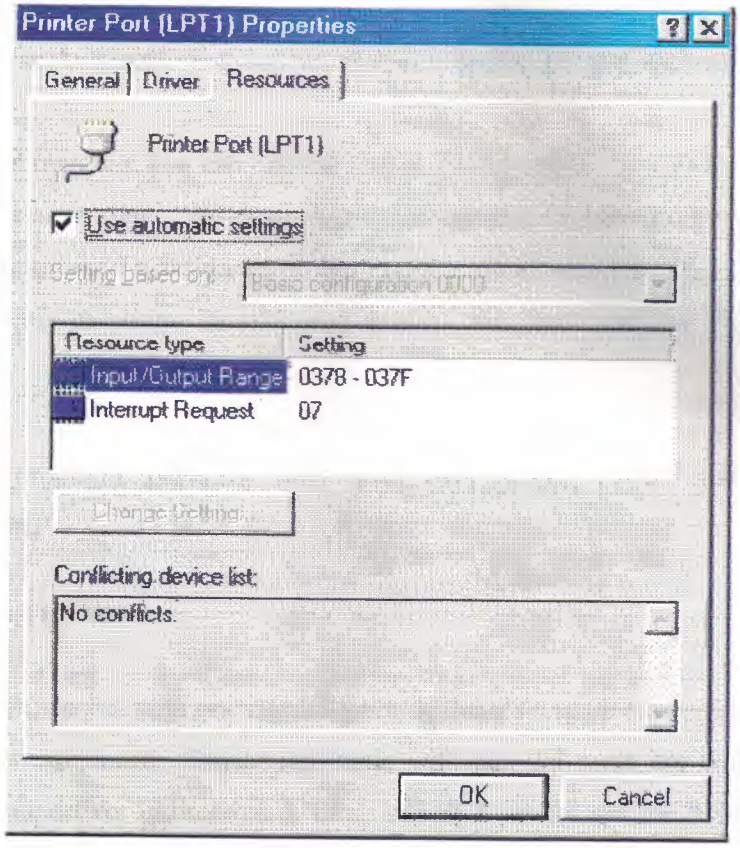


Figure 1.1 In Windows 95-98 you can select a port configuration in the Device Manager's Resources Window. A message warns if Windows detects any system conflicts with the selected configuration.

Select the desired printer model, then click Connect to view the available ports. Select a port and click OK, or Cancel to make no changes. In Windows 95, the Control Panel lists available ports under System Properties, Device Manager, Ports. There's also a brief description of the port. Printer Port means that Windows treats the port as an ordinary SPP, while ECP Printer Port means that Windows will use the abilities of an ECP if possible. To change the driver, select the port, then Properties, Driver, and Show All Drivers. Select the driver and click OK. If an ECP doesn't have an IRQ and DMA channel, the Windows 95 printer driver will use the ECP's Fast Centronics mode, which transfers data faster than an SPP, but not as fast as ECP.

The Device Manager also shows the port's configuration. Select the port, then click Resources. Figure 1.1 shows an example. Windows attempts to detect these

settings automatically. If the configuration shown doesn't match your hardware setup, deselect the Use Automatic Settings check box and select a different configuration.

If none matches, you can change a setting by double clicking on the source type and entering a new value. Windows displays a message if it detects any conflicts with the selected settings. To assign a printer to a port, click on Control Panel, Printers, and select the printer to assign Parallel port devices that don't use the Windows printer drivers.

DOS programs generally have their own printer drivers and methods for selecting a port.

1.10. Adding a Port

Most PCs come with one parallel port. If there's a spare expansion slot, it's easy to add one or two more. Expansion cards with parallel ports are widely available. Cards with support for bidirectional, EPP, and ECP modes are the best choice unless you're sure that you won't need the new modes, or you want to spend as little as possible. Cards with just an SPP are available for as little as $15. A card salvaged from an old computer may cost nothing at all.

We can get more use from a slot by buying a card with more than a parallel port. Because the port circuits are quite simple, many multi-function cards include a parallel port. Some have serial and game ports, while others combine a disk controller or other circuits with the parallel port. On older systems, the parallel port is on an expansion card with the video adapter. These should include a way to disable the video adapter, so you can use the parallel port in any system. When buying a multi-mode port, it's especially important to be sure the port comes with utilities or documentation that shows you how to configure the port in all of its modes. Some multi-mode ports default to an SPP configuration, where all of the advanced modes are locked out. Before you can use the advanced modes, you have to enable them. Because the configuration methods vary from port to port, you need documentation. Also, because the configuration procedures and other port details vary from chip to chip, manufacturers of ECP and EPP devices may guarantee compatibility with specific chips, computers, or

expansion cards. If you're in the market for a new parallel port or peripheral, it's worth trying to find out if the peripheral supports using EPP or ECP mode with your port.

1.11. Port Hardware

The parallel port's hardware includes the back panel connector and the circuits and cabling between the connector and the system's expansion bus. The PC's microprocessor uses the expansion bus's data, address, and control lines to transfer information between the parallel port and the CPU, memory, and other system components.

Figure 1.2 In the Figure on the left shows the back panel of an expansion card, with a parallel port's 25-pin female D-sub connector on the left side of the panel. (The other connector is for a video monitor.) The figure on the right shows the 36-pin Female Centronics connector used on most printers.

1.12. Basic Concepts of Computer Cabling

A parallel port consist of a 25 pin port adapter called a DB-25. Each adapter can be a male type connector with pins or a female type adapter with tiny holes. Generally a printer port (called LPT1) on the back of a computer is female type adapter and we need to use male DB-25 pin cable on it for printer connection or for parallel data transfers.

1.13. IBM Parallel port Pin Descriptions

It is difficult to make our own printer cables and they are cheap and easy to find in electronic shops. Following is a table, which shows the pin assignments of the parallel port.

Table 1.2 Pin Assignment of Parallelport (Lpt1) Female DB-25 on PC.

<= in or => out	DB25 Pin	Cent Pin	Name of Signal	Reg Bit	Function Notes
=> out	1	1	-Strobe	C0-	Set Low pulse >0.5 us to send
=> out	2	2	Data 0	D0	Set to least significant data
=> out	3	3	Data 1	D1	...
=> out	4	4	Data 2	D2	...
=> out	5	5	Data 3	D3	...
=> out	6	6	Data 4	D4	...
=> out	7	7	Data 5	D5	...
=> out	8	8	Data 6	D6	...
=> out	9	9	Data 7	D7	Set to most significant data
<= in	10	10	-Ack	S6+	IRQ; Low Pulse ~ 5 US, after accept
<= in	11	11	+Busy	S7-	High for Busy/Offline/Error
<= in	12	12	+PaperEnd	S5+	High for out of paper
<= in	13	13	+SelectIn	S4+	High for printer selected
=> out	14	14	-AutoFeed	C1-	Set Low to autofeed one line
<= in	15	32	-Error	S3+	Low for Error/Offline/PaperEnd
=> out	16	31	-Init	C2+	Set Low pulse > 50 US to init
=> out	17	36	-Select	C3-	Set Low to select printer
==	18-25	19-30, 33,17,16	Ground	-	Do not connect any of these grounds to a shield

Note: "<= In" and "=> Out" are defined from the viewpoint of the PC, not the printer. The IRQ line (-Ack/S6+) is positive edge triggered, but only enabled if C4 is 1.

1.14. Parallel Port Laplink Cable Pinout

If you are seeking to buy a Parallel port Laplink cable, or trying to make your own cable, you should know what pins need to be switched in order to make it. Below is a table showing pin connections. Only 18 pins are used in a Laplink Cable, therefore only these pins are shown here.

To make this cable we need :

1. Two DB-25 type male sockets.
2. Shielded cable with 18 cores (lines of wires).

Table 1.3 Parallel Laplink Cable Pinouts

Male DB-25	==>>	Male DB-25
1	Both Not used	
2	To	15
3	To	13
4	To	12
5	To	10
6	To	11
7	Both Not used	
8	Both Not used	
9	Both Not used	
10	To	5
11	To	6
12	To	4
13	To	3
14	Both Not used	
15	To	2
16	Both Not used	
17	To	19
18	To	18
19	To	17
20	Both Not used	
21	To	21
22	To	22
23	To	23
24	Both Not used	
25	To	25
Pinbody*	To	Pinbody

*In this project, one wire was attached to the metal body of the Male pins on both sides. Total 18 wired cable is necessary for this cable including one wire for body of the pin too.

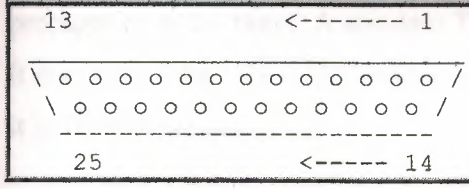
IMP: DCC users to troubleshoot and test DCC connection and cable on both the computers. It also provides detailed information about the connection, the cable being used for the connection, the I/O mode (4-bit, 8-bit, ECP), the parallel port types, I/O address, and IRQ. The company, Parallel Technologies, is associated with the Microsoft for DCC and can be seen listed in Win9x Help (Index+Direct Cable Connection + ordering cables).

SPEED: Parallel port Laplink cable is always faster than Serial port cable because of more numbers of cores of wires used in Parallel port cable (25 pin) than Serial port cable (9 pins). The expected speed is 300kb/second but it is extremely dependent on the different quality chipset structure of Parallel Ports on different makes of the Motherboards. Some even reported the lowest speed of 60kb/sec even though all other settings are correct. It is recommended that you setup LPT1 mode as only "ECP/EPP" or "ECP" mode in bios to get better speed, and not the "Normal" (4bit/8bit) modes.

1.15. IBM DB-25 numbering scheme. (Pin numbers)

Each pin has a number assigned to it. When connecting null modem, for example, it is important to know these numbers in order to select the correct cables, or when making your own cables.

Table 1.4 DB-25 Connectors (Female and Male)

Female	Male
	

Chapter 2 INTRODUCTION TO TTL GATES

In this chapter some information will be given about the Transistor-Transistor Logic (TTL) family.

2.1. 7400 Devices

The 7400 series is a line of TTL circuits introduced by Texas instruments in 1964 and it has become the most widely used of all bipolar IC's. This TTL family contains a variety of SSI and MSI chips that allow one to build all kinds of digital circuits and systems.

2.2. Propagation Delay Time and Power Dissipation

Two quantities needed for our later discussion are power dissipation and propagation delay time. A standard TTL gate has a power dissipation of about lO mW. It may vary from this value because of signal levels, tolerances. etc. but on the average it is 10 mW per gate.

The propagation delay time is the amount of time it takes for the output of a gate to change after the inputs have changed. The propagation delay time of a TTL gate is in the vicinity of 10 ns.

2.3. 5400 Series

Any devices in the 7400 series works over a temperature range of O^0 to $70C^0$ and over a supply range of 4.75 to 5.25 V. This is adequate for commercial applications. The 5400 series, developed for the military applications has the same logic functions as the 7400 series, except that it works over a temperature range of -55^0 to 125 C^0 and over a supply range of 4.5 to 5.5V. Although 5400-series devices can replace 7400 series devices, they are rarely used commercially because of their much higher cost.

2.4. High Speed TTL

By decreasing the internal resistances a manufacturer can lower the internal time constants; this decreases the propagation delay time. The smaller resistances, however, increase the power dissipation. This variation is known as High Speed TTL. Devices of this type one numbered 74H00, 74H01, 74H02 etc. A High Speed TTL gate has a power dissipation of around 22mW and a propagation delay time of approximately 5 ns.

2.5. Low Power TTL

By increasing the internal resistances a manufacturer can reduce the power dissipation of TTL gates. Devices of this type are called Low Power TTL and are numbered 74L00, 74L01, 74L02 etc. These devices are slower than the standard TTL because of the larger internal time constants. A Low Power TTL gate has a power dissipation of approximately 1 mW and a propagation delay time around 35 ns.

2.6. Schottky TTL

With standard TTL, High Speed TTL and Low Power TTL the transistors go into saturation causing extra carriers to flood the base. If you try to switch this transistor from saturation to cut-off, you have to wait for the extra carriers to flow out of the base; the delay is known as the saturation delay time.

One way to reduce the saturation delay time is with Schottky TTL. The idea is to fabricate a Schottky diode along with each bipolar transistor of a TTL circuit, because the Schottky diode has a forward voltage of only 0.4 V, it prevents the transistor from saturating fully. This virtually eliminates saturation delay time, which means better switching speed. This variation is called Schottky TTL; These devices are numbered as 74S00, 74S01, 74S02 etc.

Schottky TTL devices are very fast, capable of operating reliably at 100 MHz The 74S00 has a power dissipation around 20mW per gate and a propagation delay time of approximately 3 ns.

2.7. Low Power Schottky TTL

By increasing the internal resistance as well as using Schottky diodes manufacturers have come up with the best compromise between low power and high Speed : Low Power Schottky TTL. Devices of this type are numbered 74LS00, 74LS01 ,74LS02 etc. A Low Power Schottky gate has a power dissipation of around 2 mW and propagation delay time is approximately 10 ns.

Standard TTL and low-power Schottky TTL are the main stays of the digital designer. TTL and low power Schottky TTL have emerged as the favorites of the digital designers. You will see them more than any other bipolar types.

2.8. TTL Characteristics

2.8.1. Worst-Case input Voltage

Table 2.1 shows us the TTL inverter with an input voltage of Vi and an output voltage of Vo. When Vi is 0 V (Grounded) the output voltage is high. With TTL devices, we can raise Vi to 0.8 V and still have an output. The maximum low-level input voltage is designated V_{IL} . Data sheets list this worst-case low input as;

$$V_{IL} = 0.8 \text{ V}$$

Take the other extreme and suppose Vi is 5 V in Table 2.1 this is a high input therefore the output of the inverter is low , Vi can decrease all the way down to 2 V, and the output will still be low. Data sheets list this worst-case high input as:

$$V_{IH} = 2 \text{ V}$$

In other words, any input voltage from 2 to 5 V is a high input for TTL devices.

2.8.2. Worst-Case Output Voltages

Ideally 0 V is the low output and 5 V is the high output. We cannot attain these ideal values because of internal voltage drops. Any voltage from 0 to 0.4 V is a low output. This means that the worst-case output values are:

$$V_{OL} = 0.4V \qquad V_{OH} = 2.4$$

Table 2.1 TTL states (Worst Case)

	Output Vo	**Input Vi**
Low	**0.4**	**0.8**
High	**2.4**	**2**

2.8.3. Standard Loading

A TTL device can source current (high output) or it can sink current (Low output). Data sheets of standard TTL devices indicate that any 7400-series device can sink up to 16 ms a designated as;

$$I_{OL} = 16mA$$

And can source up to 400 μA

$$I_{OH} = -400\ \mu A$$

Minus sign means that the current is out of the device and a plus sign means that it is into the device. A single TTL load has a low-level input current of 1.6 mA and a high-level input current of 400 μA since the maximum output current is 10 times greater than input so that we can connect 10 TTL emitters to any TTL output. In the Table 2.2 we can see the number of loads for each type of TTL.

Table 2.2: Fan-outs

TTL Driver	TTL Load				
	74	74H	74L	74S	74LS
74	10	8	40	8	20
74H	12	10	50	10	25
74L	2	1	20	1	10
74S	12	10	100	10	50
74LS	5	4	40	4	20

2.9. Definitions of Device Terminology

Orderable Device

The complete part number required to order this device.

Generic Device

The orderable part number minus the package nomenclature.

Device Description

The description of the designated generic or orderable device number.

Functionality

The functionality of the designated generic device number.

Package

The package designator code used by Texas Instruments to identify a specific package.

Pins

The number of external pins on the device package.

Temp

The operating temperature range for the device:

Blank = See data sheet for operating temperature.

S = Special temperature range, see data sheet.

Status

The current status of the device:

PREVIEW = Announced device not yet in production.

ACTIVE = Active device available for purchase from a TI Authorised Distributor.

NRND = Not Recommended for New Designs. Device is in production.

LIFEBUY = Lifetime buy period in effect. Device is being discontinued.

OBSOLETE = Obsolete, discontinued device.

Military Status

The availability of the device in Military Packages:

NO = Available in Commercial Packaging.

YES = Available in Military Packaging.

BOTH = Available in both Commercial and Military Packaging.

Pack Qty

This is the standard package quantity, i.e. devices per tube, box or reel.

Availability/Samples

The "Check stock or order" link will go to a dynamic web page that displays the selected device inventory availability at TI Authorised Distributors in your country and the availability of samples from TI.

If TI offers free samples for the device, sample requests can be entered online. If a distributor in your country offers online ordering, you can place an order immediately.

To protect your personal information, the www.a.ti.com server uses a secure https connection. If you have problems connecting to the server, verify the following:

Your web browser supports SSL. (Netscape Navigator 2.0 or higher and Microsoft Internet Explorer 3.0 or higher should work, however, using the latest web browser version is highly recommended.) SSL2 and SSL3 have not been disabled in your web browser.

Your Security Proxy settings are correct. (Contact your network administrator if you need assistance.)

Your company firewall permits https and SSL connections. (Contact your network administrator.)

DSCC Number

Defense Supply Center, Columbus. Texas Instruments fully supports the Standard Microcircuit Drawing (SMD) program. Devices procured to SMDs are compliant to MIL-PRF-38535 and offer customers products tested to industry standard electricals.

2.10. TTL Logic Gate Devices

TTL device codes used in project are given in table 2.3 and here the devices are preceded by '74..' (where '..' represents the infix code.). except for the 'STD ' series which are simply preceded by '74' without an infix code.

Table 2.3 TTL Device Codes

Device	Function	Ac	Act	Als	As	B	Bctc	F	Fct	H	Hc	Hct	L	Ls	S	Std
00	Quad 2-input NAND	X	X	X	X	X	X	X	-	X	X	X	X	X	X	X
03	Quad 2-input o.c. NAND	-	-	X	-	-	-	-	-	-	X	X	X	X	X	X
08	Quad 2-input AND	X	X	X	X	X	X	X	-	X	X	X	-	X	X	X
09	Quad 2-input o.c. AND	-	-	X	-	-	-	-	-	-	-	-	-	X	X	X
24	Quad Sch. NOR	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-
26	Quad 2-input o.c. h.v. NAND	-	-	-	-	-	-	-	-	-	-	-	X	X	-	X
32	Quad 2-input OR	X	X	X	X	-	X	X	-	-	X	X	X	X	X	X
37	Quad 2-input NAND buffer	-	-	X	-	-	-	X	-	-	-	-	-	X	X	X
38	Quad 2-input o.c. NAND buffer	-	-	X	-	-	-	X	-	-	-	-	-	X	X	X
86	Quad 2-input exclusive-OR	X	X	X	X	-	X	X	-	-	X	X	X	X	-	-

2.11. Comparison of Characteristics

Table 2.4 Shows a comparision of the various TTL device characteristics.

Characteristic	Logic Family 74Ls
Maximum supply voltage	**5.25 V**
Minimum supply voltage	**4.75 V**
Power dissipation (mW per gate at 100 kHz)	**2**
Dynamic power dissipation (mW per gate at 100 kHz)	**2**
Typical propagation delay (ns)	**10**
Maximum clock frequency (MHz)	**40**
Speed-power product (pj at 100 KHz)	**20**
Minimum output current (mA at V=0.4 V)	**8**
Fan-Out (Ls loads)	**20**
Maximum input current (mA at V=0.4 V)	**-0.4**

2.12. SN74LS00, Quad 2-input positive-NAND gates

Table 2.5 Properties of 74LS00

PARAMETER NAME	SN74LS00
Voltage Nodes (V)	5
Vcc range (V)	4.75 to 5.25
Input Level	TTL
Output Level	TTL
Output Drive (mA)	-0.4/8
No. of Gates	4
Static Current	3
Tpd(max) (ns)	15

Features

Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPS

Description

These devices contain four independent 2-input-NAND gates.

The SN5400, SN54LS00, and SN54S00 are characterised for operation over the full military temperature range of -55°C to 125°C. The SN7400, SN74LS00, and SN74S00 are characterised for operation from 0°C to 70°C.

2.13. SN74LS03, Quad 2-input positive-NAND gates with open collector outputs

Table 2.6 Properties of 74LS03

PARAMETER NAME	SN74LS03
Voltage Nodes (V)	5
Vcc range (V)	4.75 to 5.25
Input Level	TTL
Output Level	TTL
Output Drive (mA)	-/8
No. of Gates	4
Static Current	3
tpd(max) (ns)	32

Features

Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPs

Description

These devices contain four independent 2-input-NAND gates. The open-collector outputs require pull up resistors to perform correctly. They may be connected to other open-collector outputs to implement active-low wired-OR or active-high wired-AND functions. Open-collector devices are often used to generate higher V_{OH} levels.

The SN5403, SN54LS03 and SN54S03 are characterised for operation over the full military temperature range of -55°C to 125°C. The SN7403, SN74LS03 and SN74S03 are characterised for operation from 0°C to 70°C.

2.14. SN74LS08, Quadruple 2-Input Positive-AND Gates

Table 2.7 Properties of 74LS08

PARAMETER NAME	SN74LS08
Voltage Nodes (V)	5
Vcc range (V)	4.75 to 5.25
Input Level	TTL
Output Level	TTL
Output Drive (mA)	-0.4/8
No. of Gates	4
Static Current	6.8
tpd(max) (ns)	20

Features

Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPs

Description

These devices contain four independent 2-input AND gates.

The SN5408, SN54LS08, and SN54S08 are characterised for operation over the full military temperature range of -55°C to 125°C. The SN7408, SN74LS08 and SN74S08 are characterised for operation from 0° to 70°C.

2.15. SN74LS09, Quad 2-input positive-AND gates with open collector outputs

Table 2.8 Properties of 74LS09

PARAMETER NAME	SN74LS09
Voltage Nodes (V)	5
Vcc range (V)	4.75 to 5.25
Input Level	TTL
Output Level	TTL
Output Drive (mA)	-/8
No. of Gates	4
Static Current	6.8
tpd(max) (ns)	35

Features

Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPS.

Description

These devices contain four independent 2-input AND gates. The open-collector outputs require pull-up resistors to perform correctly. They may be connected to other open-collector outputs to implement active-low wired-OR or active-high wired-AND functions. Open-collector devices are often used to generate higher V_{OH} levels.

The SN5409, SN54LS09, and SN54S09 are characterised for operation over the full military temperature range of -55°C to 125°C, The SN7409, SN74LS09, and SN74S09 are characterised for operation from 0°C to 70°C.

2.16. SN74LS26, Quad 2-input high-voltage interface positive-NAND gates

Table 2.9 Properties of 74LS26

PARAMETER NAME	SN74LS26
Voltage Nodes (V)	5
Vcc range (V)	4.75 to 5.25
Input Level	TTL
Output Level	TTL
Output Drive (mA)	-/8
No. of Gates	4
Static Current	3
tpd(max) (ns)	32

Features

For Driving Low-Threshold-Voltage MOS Inputs

Description

These 2-input open-collector NAND gates feature high-output voltage ratings for interfacing with low-threshold-voltage MOS logic circuits or other 12-volt systems. Although the output is rated to withstand 15 volts, the V_{CC} terminal is connected to the standard 5-volt source.

The SN5426 and SN54LS26 are characterised for operation over the full military temperature range of -55°C to 125°C. The SN7426 and SN74LS26 are characterised for operation from 0°C to 70°C.

2.17. SN74LS32, Quad 2-input positive-OR gates

Table 2.10 Properties of 74LS32

PARAMETER NAME	SN74LS32
Voltage Nodes (V)	5
Output Level	TTL
Static Current	8

Features

Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPs

Description

These devices contain four independent 2-input OR gates.

The SN5432, SN54LS32 and SN54S32 are characterised for operation over the full military range of -55°C to 125°C. The SN7432, SN74LS32 and SN74S32 are characterised for operation from 0°C to 70°C.

2.18. SN74LS37, Quad 2-input positive-NAND buffers

Table 2.11 Properties of 74LS37

PARAMETER NAME	SN74LS37
Voltage Nodes (V)	5
Vcc range (V)	4.75 to 5.25
Input Level	TTL
Output Level	TTL
Output Drive (mA)	-1.2/24
No. of Gates	4
Static Current	7
tpd(max) (ns)	24

Features

Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPs

Description

These devices contain four independent 2-input NAND buffer gates.

The SN5437, SN54LS37 and SN54S37 are characterised for operation over the full military range of -55°C to 125°C. The SN7437, SN74LS37 and SN74S37 are characterised for operation from 0°C to 70°C.

2.19. SN74LS38, Quad 2-input positive-NAND buffers with open collector

Table 2.12 Properties of 74LS38

PARAMETER NAME	SN74LS38
Voltage Nodes (V)	5
Vcc range (V)	4.75 to 5.25
Input Level	TTL
Output Level	TTL
Output Drive (mA)	- /24
No. of Gates	4
Static Current	7
tpd(max) (ns)	32

Features

Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPs

Description

These devices contain four independent 2-input NAND buffer gates with open-collector outputs. The open-collector outputs require pull-up resistors to perform correctly. They may be connected to other open-collector outputs to implement active-low wired-OR or active-high wired-AND functions. Open-collector devices are often used to generate high V_{OH} levels.

The SN5438, SN54LS38, and SN54S38 are characterised for operation over the full military temperature range of -55°C to 125°C. The SN7438, SN74LS38, and SN74S38 are characterised for operation from 0°C to 70°C.

2.20. Structure of Some Common TTL Gates

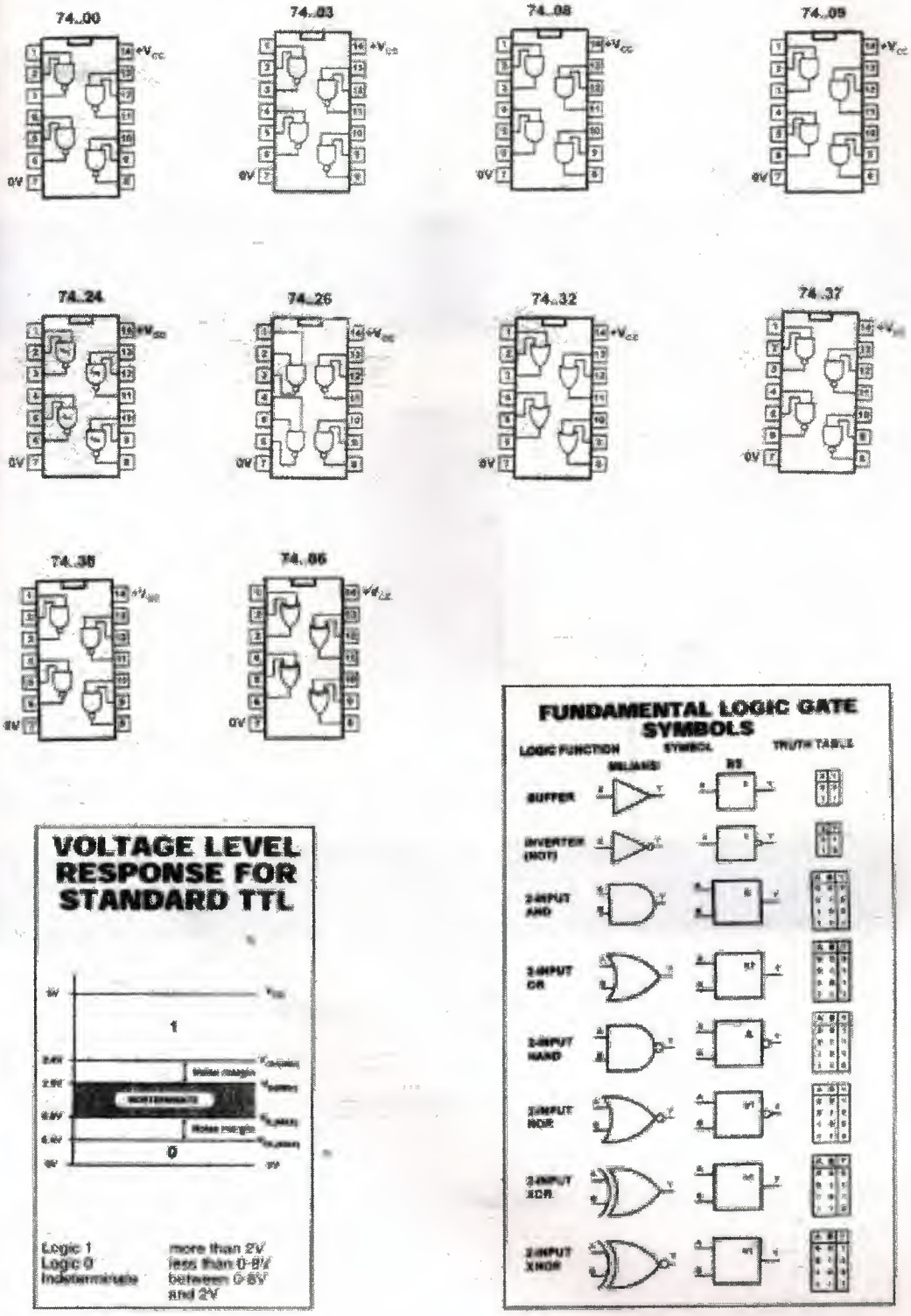


Figure 2.1 Inside of Chips and Logic Gate Symbols and Voltage Level For TTL (TTL Data Book, Texas Instruments)

Chapter 3 DESCRIPTION OF THE PROGRAM

3.1. Main Program

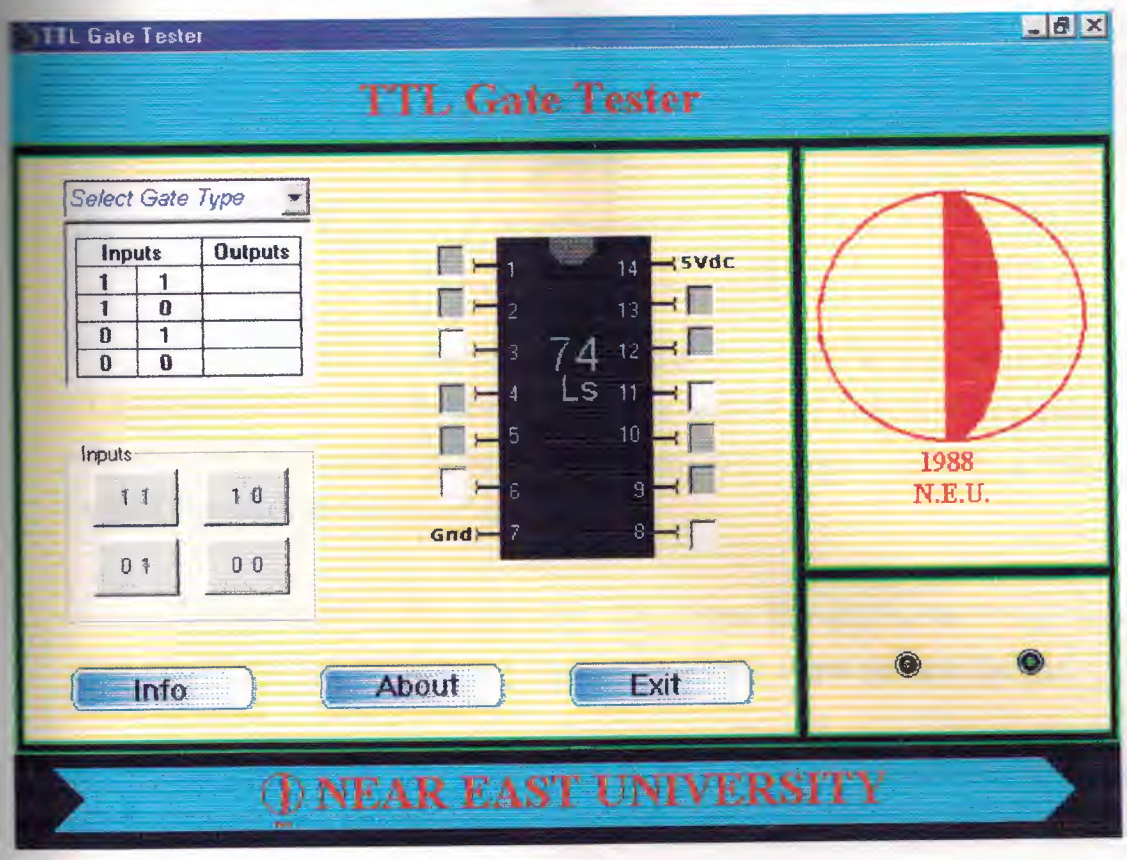


Figure 3.1 Main Menu Form

The above shown form is the starting of the program.

In this program, we can test some 74xx series TTL chips. To work with this program, we have to know the following:

a) First of all, we have to select the chip which we are going to use for testing. This chip can be selected from the selection box provided on the form.

b) After we have selected the chip to be tested, we have to send data to the chip. For data to be sent we will use a frame "Input" provided on the form. This frame named "Input" has four push buttons. By pressing these buttons, we can send data to the chip that is connected to our hardware. By pressing

one button, we can send two bits as input for the chip. These bits can be categorised as:

Table 3.1 Inputs of the TTL chips

A	B
1	1
1	0
0	1
0	0

This data will be received by the input pins of the chip. These pins are: 1,2;4,5;9,10;12,13. The data being will be seen on the form from these input pins. As this data will be received by the chip connected to hardware. Chip will produce an output depending on the type of chip.

The output data will be sent to the some output pins of the chip. These output pins can be seen from the form given is program. These output pins are: 3,6,8,11.

We can check the output by looking at the hardware as well as on the form given in the program. For output, we used two LEDs, one red and one green. These LEDs can be seen on the hardware as well as in form will turn on. On the other condition, if the red LED is turned on, it indicates that one or more integrated gates in the chip are not working properly or damaged. The gates in the chip which are not working properly can be seen pin the form given by the picture of the chip.

According to these conditions, we can check the reason of the damaged gates, by looking at the output pins shown in the picture of the chip on the form and comparing these outputs with the expected outputs shown in the table on the form. The coding of this program can be seen in the Appendix A. This program also includes:

3.2. How to use the program

This part gives the information about how to run and use the program. Also, it gives information about the things which are needed to be cared of while we can see the appearance of this part shown below in figure.

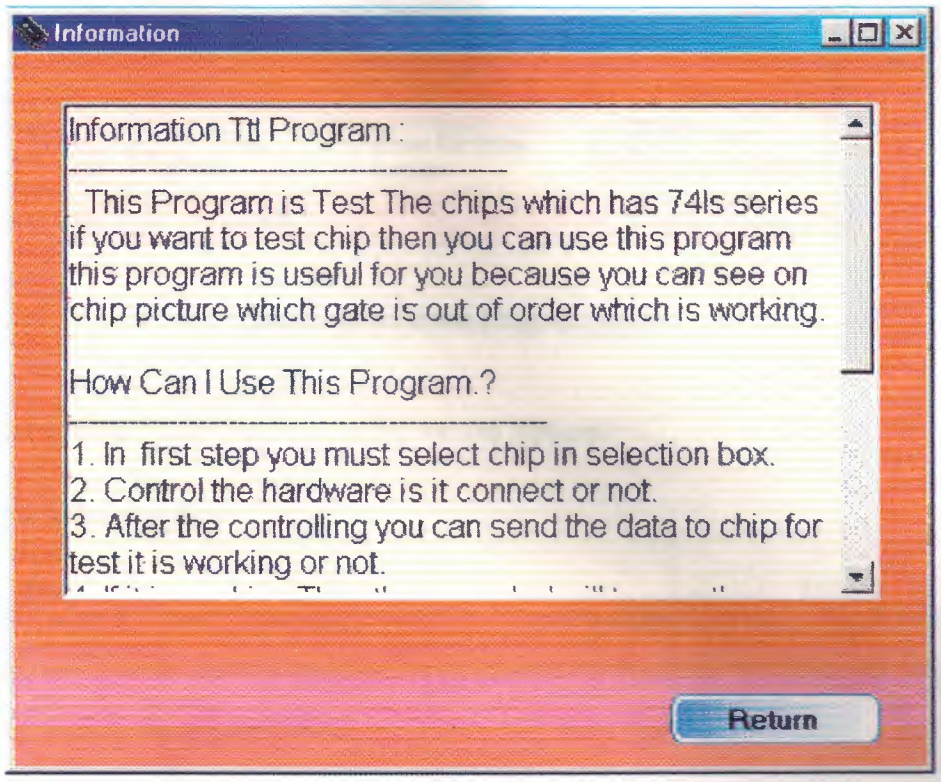


Figure 3.2 Information Form

The coding of this portion can be seen in Appendix B.

3.3. About the program

This part of the form gives details about the program version, programmer who coded the program. This part is named as "About" on the main form and can be seen below.

Figure 3.3 About Program Form

The coding of this portion can be seen in Appendix C.

3.3.1. Details the program

The above shown figure contains one more button "Details". While pressing this button, we can the see the internal configuration of the chips which are to be tested. This can be seen in the figure below.

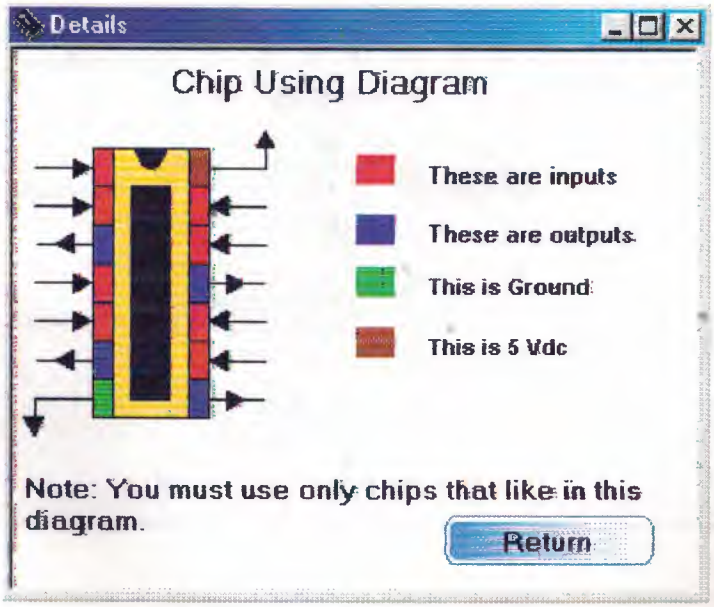


Figure 3.4 Details Form

The coding of this portion can be seen in Appendix D.

3.4. Exit the program

The last part of the form is button having a name "Exit". While pressing this button, we will exit the program. The same time, the parallel port of the computer which was used for transmission of the data will return to its default settings. If we do not use the exit button provided on the form. We will not be able to use the parallel port of the computer for other purposes and there will be a need for a reboot of the system.

3.5. Port tester

There is another program in order to see the changes of the portions of the parallel port which are;

- Status port
- Data port
- Control port

This program gives information about the changes in data sent from parallel port. We can see the appearance of this part shown below in figure.

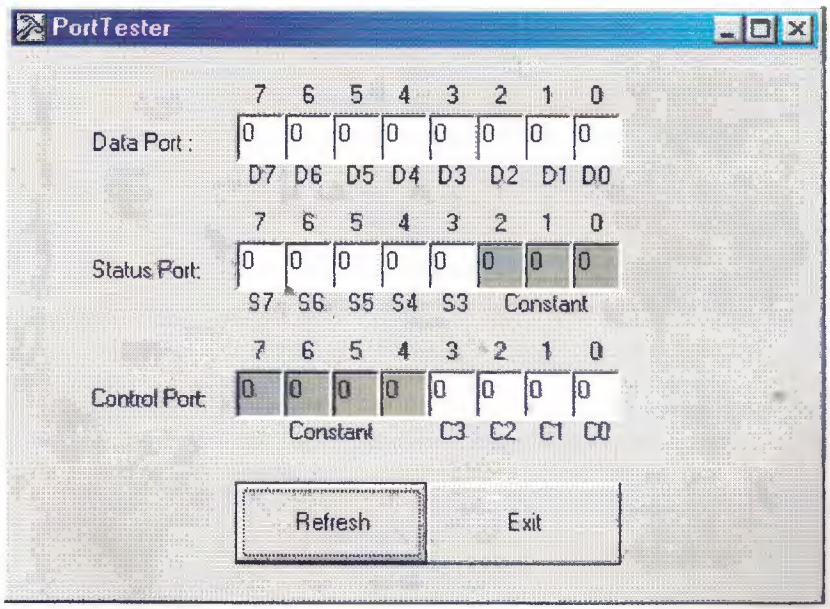


Figure 3.5 Port Tester Form

The coding of this portion can be seen in Appendix E.

3.6. Modules

In the program we have written for testing the chips, there is need for interfacing the software with the hardware. In order to make interface between the program and the hardware, we had to use a module written in Visual basic. This module is used in the program. Inside this module, we created two commands. One of the commands is INP. This command is used to accept the data that is sent from the port. Second command we used is OUT. This command sends the data to the port as output. The use of these commands can be seen below.

```
INP address;
OUT address, data;
```

Where

Address: is an integer value.

Data: is hexadecimal or integer;

The coding of this portion can be seen in Appendix F.

Chapter 4 THE HARDWARE

4.1. Hardware Description

A hardware was designed and constructed for this project. This hardware accepts the gate to be tested and is connected to the parallel port of the PC using a 25 way connector.

The following components were used in the hardware:

- Two LEDs (Green,Red)
- Two Resistors (Each 330Ω)
- One 14-pins Slot.
- One 36-pin Female Socket
- Board
- Adapter socket

A simple appearance of this hardware can be shown as below in Figure 4.1

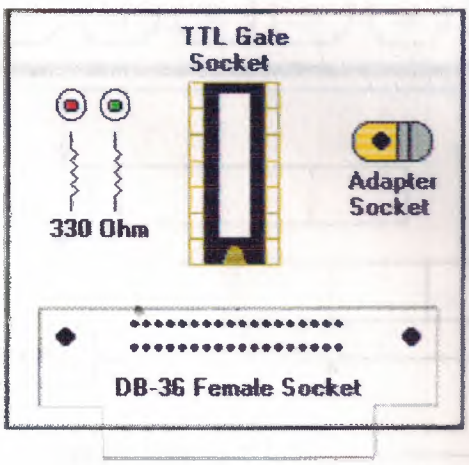


Figure 4.1 A layout of the hardware used for the program.

4.2. Connection of Hardware

In order to work with this program for the hardware we made, there must be some connections which have to be made. The diagram of these connections is shown below in Figure 4.2.

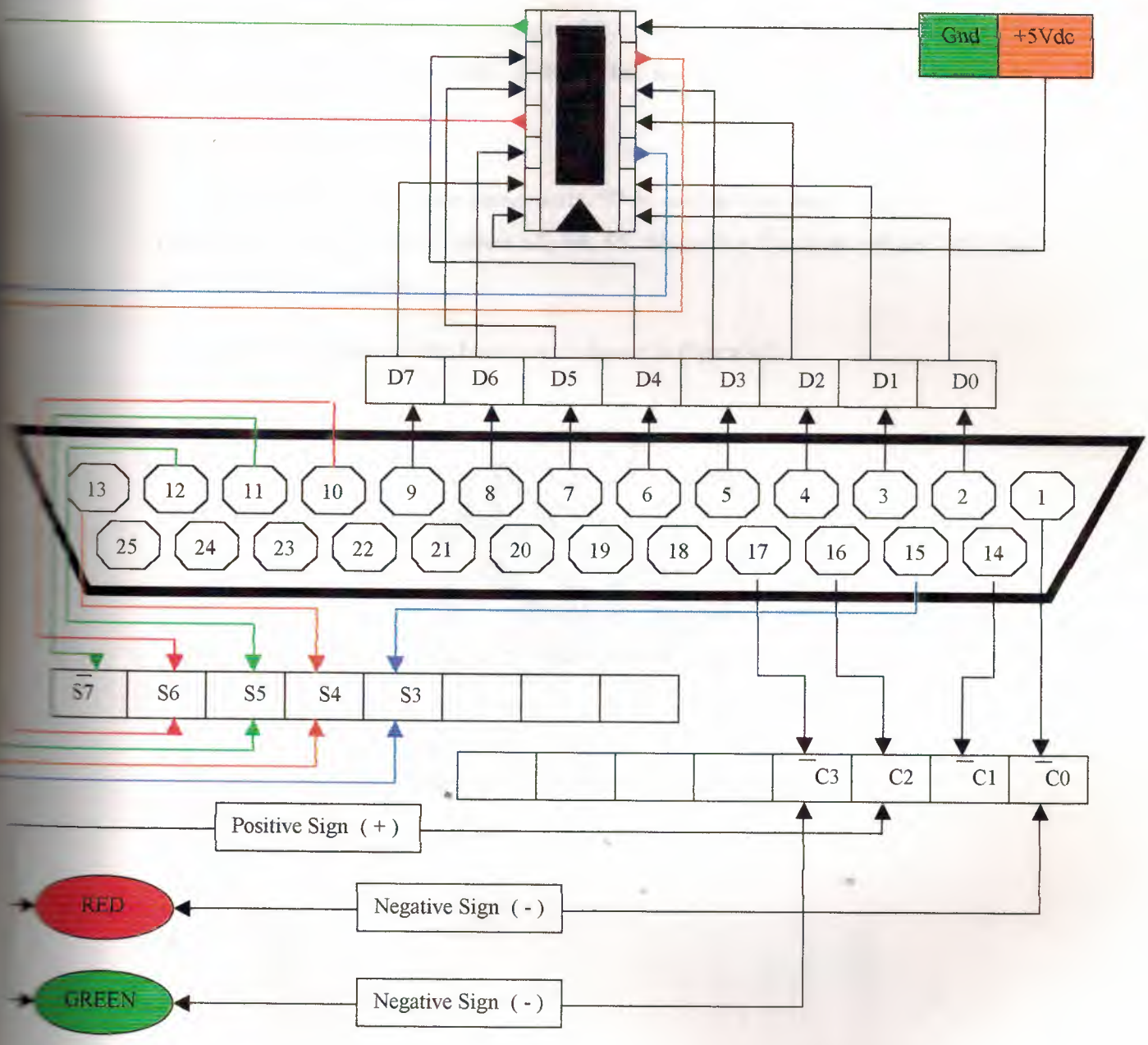


Figure 4.2 Pinouts of port Connecting to the hardware

In the above shown diagram, D0, D1, D2……D7 represent the data that has to be sent from the program.

C0, C1, C2, C3 are control ports. C0 is connected to negative pin of Red LED. C2 is connected to positive pins of Red LED and green LED. C3 is connected to negative pin of green LED.

S3, S4, S5, S6, S7 are status ports. S7 is used to test whether the hardware is connected or not. Where as, others S3, S4, S5, S6 control the output of the TTL gate when data is sent to them.

The PCB layout of the hardware is shown in Figure 4.3.

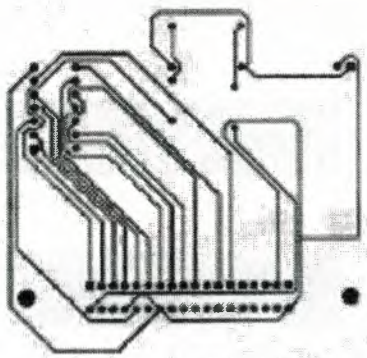


Figure 4.3. PCB layout of the hardware

Chapter 5 THE SOFTWARE

The software was developed using the Visual Basic & programming language. The software flow chart is shown in Figure 5.1.

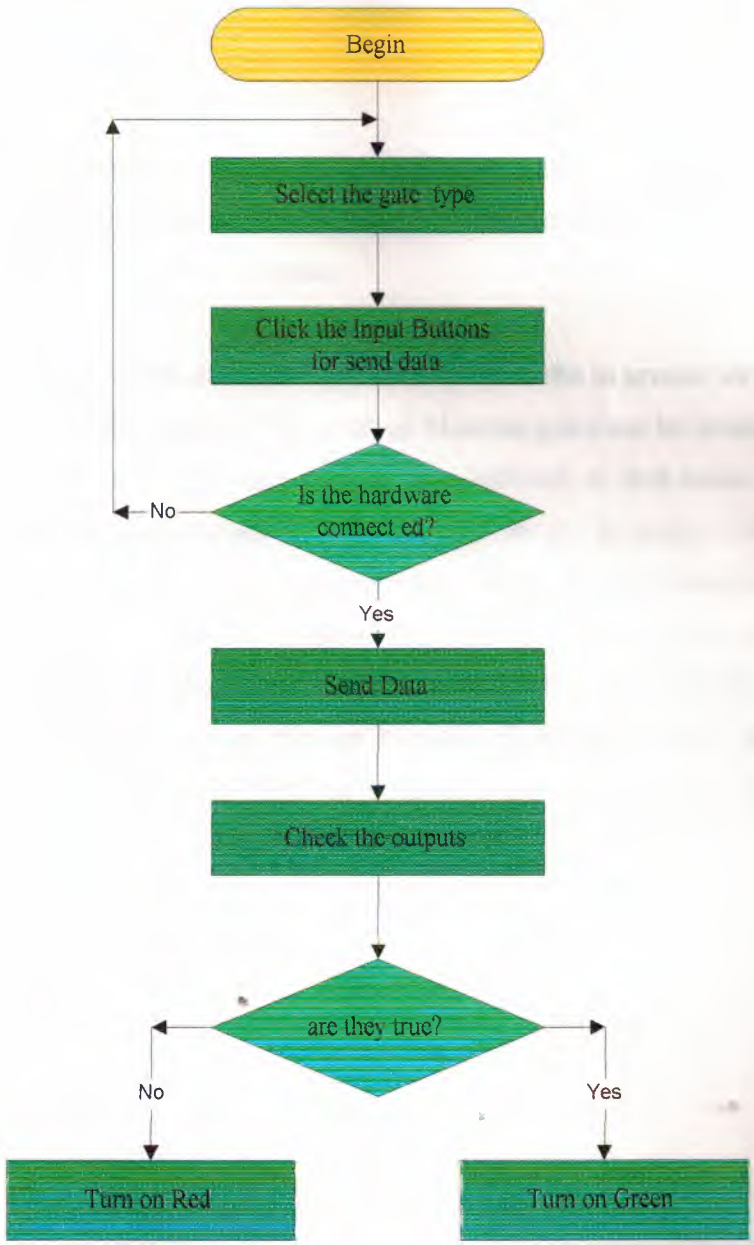


Figure 5.1 The flowchart of the TTL Gate Tester Program.

CONCLUSION

A simple personal computer based TTL gate tester project has been described. The project consists of both hardware and software. The gate to be tested is mounted on a socket and the hardware is connected to the standard parallel port of a personal computer. A Visual Basic program was developed which enables the user to select the gate type and then test the gate. The program provides visual indication as to whether or not the gate under test is faulty.

Several standard 74 series TTL gates have successfully been tested with the project. Faulty gate behaviour was simulated by connecting jumpers on the gate socket to send wrong signals to the computer.

The project can be enhanced and made more useful in several ways. First of all, currently only a small subset of the standard 74 series gates can be tested. It should be possible to modify both the hardware and the software so that more complex logic circuits such as flip-flops, counters, and shift-registers can be tested. The project tests only the static performance, i.e. the truth-table of the gates. It should also be possible to modify the software so that the dynamic features such as the timing accuracy of the gates can be tested. Another suggestion for future work is to modify the hardware and software so that more than one chip can be tested at the same time. This new feature will be very useful when it is required to test a number of identical TTL chips at the same time.

REFERENCES

Internet Addresses:

www.ti.com

www.epemag.com

www.yahoo.com

www.altavista.com

www.lvr.com

www.vbhome.cjb.net

Name of the books:

VISUAL BASIC 6.0, *Türkmen yayınevi,* **Ihsan KARAGÜLLE & Zeydin PALA**

APPENDIX A

Main Program Listing

```
Dim i, choice, x1, x2, x3, x4, x5, s3, s4, s5, s6, s7 As Integer
Private Sub Combo1_Click()
If Combo1.ListIndex = 0 Then
 Label19.Caption = "00"
 Picture3.Visible = True
 Picture9.Visible = False
 Picture10.Visible = False
 Picture12.Visible = False
 Picture13.Visible = False
 Picture14.Visible = False
End If
If Combo1.ListIndex = 1 Then
 Label19.Caption = "03"
 Picture3.Visible = True
 Picture9.Visible = False
 Picture10.Visible = False
Picture12.Visible = False
 Picture13.Visible = False
 Picture14.Visible = False
End If
If Combo1.ListIndex = 2 Then
 Label19.Caption = "08"
 Picture3.Visible = False
 Picture9.Visible = True
 Picture10.Visible = False
 Picture12.Visible = False
 Picture13.Visible = False
 Picture14.Visible = False
End If
If Combo1.ListIndex = 3 Then
```

```
 Label19.Caption = "09"
 Picture3.Visible = False
 Picture9.Visible = True
 Picture10.Visible = False
 Picture12.Visible = False
 Picture13.Visible = False
 Picture14.Visible = False
End If
If Combo1.ListIndex = 4 Then
 Label19.Caption = "24"
 Picture3.Visible = False
 Picture9.Visible = False
 Picture10.Visible = False
 Picture12.Visible = False
 Picture13.Visible = False
 Picture14.Visible = True
End If
If Combo1.ListIndex = 5 Then
 Label19.Caption = "26"
 Picture3.Visible = True
 Picture9.Visible = False
 Picture10.Visible = False
 Picture12.Visible = False
 Picture13.Visible = False
 Picture14.Visible = False
End If
If Combo1.ListIndex = 6 Then
 Label19.Caption = "32"
 Picture3.Visible = False
 Picture9.Visible = False
 Picture10.Visible = True
 Picture12.Visible = False
 Picture13.Visible = False
 Picture14.Visible = False
```

```
End If
If Combo1.ListIndex = 7 Then
 Label19.Caption = "37"
 Picture3.Visible = True
 Picture9.Visible = False
 Picture10.Visible = False
 Picture12.Visible = False
 Picture13.Visible = False
 Picture14.Visible = False
End If
If Combo1.ListIndex = 8 Then
 Label19.Caption = "38"
 Picture3.Visible = True
 Picture9.Visible = False
 Picture10.Visible = False
 Picture12.Visible = False
 Picture13.Visible = False
 Picture14.Visible = False
End If
If Combo1.ListIndex = 9 Then
 Label19.Caption = "86"
 Picture3.Visible = False
 Picture9.Visible = False
 Picture10.Visible = False
 Picture12.Visible = False
 Picture13.Visible = True
 Picture14.Visible = False
End If
End Sub

Private Sub Form_Load()
 Out 888, 0
 Out 890, 4
 Label2.Visible = False
```

```
Label3.Visible = False
Picture3.Visible = False
Picture4.Visible = True
Picture5.Visible = False
Picture6.Visible = True
Picture7.Visible = False
Picture9.Visible = False
Picture10.Visible = False
Picture12.Visible = True
Picture13.Visible = False
Picture14.Visible = False
End Sub

Private Sub DecToBin(X As Integer)
Dim Value, i, y0, y1, y2, y3, y4, y5, y6, y7 As Integer
 Out 888, X
 For i = 1 To 10
 i = i + 1
 Next
 Value = Inp(889)
If Value >= 128 Then
 y7 = 1
 Value = Value - 128
Else
 y7 = 0
End If
If Value >= 64 Then
 y6 = 1
 Value = Value - 64
Else
 y6 = 0
End If
If Value >= 32 Then
 y5 = 1
```

```
 Value = Value - 32
Else
 y5 = 0
End If
If Value >= 16 Then
 y4 = 1
 Value = Value - 16
Else
 y4 = 0
End If
If Value >= 8 Then
 y3 = 1
 Value = Value - 8
Else
 y3 = 0
End If
If Value >= 4 Then
 y2 = 1
 Value = Value - 4
Else
 y2 = 0
End If
If Value >= 2 Then
 y1 = 1
 Value = Value - 2
Else
 y1 = 0
End If
If Value >= 1 Then
 y0 = 1
 Value = Value - 1
Else
 y0 = 0
End If
```

```
 x1 = y3
 x2 = y4
 x3 = y5
 x4 = y6
 x5 = y7
End Sub

Private Sub Send11_Click()
 Call DecToBin(1)
 s7 = x5
 If s7 = 0 Then
 Show
 choice = MsgBox("Hardware Not Found !", vbRetryCancel + vbExclamation,
"Hardware Error")
 If choice = 4 Then
  Call Form_Load
 End If
 If choice = 2 Then
  Out 888, 4
  Out 890, 12
  End
 End If
 End If
  Text1.Text = "1"
  Text2.Text = "1"
  Text3.Text = "1"
  Text4.Text = "1"
  Text5.Text = "1"
  Text6.Text = "1"
  Text7.Text = "1"
  Text8.Text = "1"
If Combo1.ListIndex = 0 Then
 Call DecToBin(&H3)
  s3 = x1
```

```
Call DecToBin(&HC)
 s4 = x2
Call DecToBin(&H30)
 s5 = x3
Call DecToBin(&HC0)
 s6 = x4
If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 1 Then
 Call DecToBin(&H3)
```

```
s3 = x1
Call DecToBin(&HC)
s4 = x2
Call DecToBin(&H30)
s5 = x3
Call DecToBin(&HC0)
s6 = x4
If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
Out 890, 12
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = False
Label3.Visible = True
Picture4.Visible = True
Picture5.Visible = False
Picture6.Visible = False
Picture7.Visible = True
Else
Out 890, 5
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = True
Label3.Visible = False
Picture4.Visible = False
Picture5.Visible = True
Picture6.Visible = True
Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 2 Then
```

```
Call DecToBin(&H3)
 s3 = x1
Call DecToBin(&HC)
 s4 = x2
Call DecToBin(&H30)
 s5 = x3
Call DecToBin(&HC0)
 s6 = x4
If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
```

```
If Combo1.ListIndex = 3 Then
 Call DecToBin(&H3)
  s3 = x1
 Call DecToBin(&HC)
  s4 = x2
 Call DecToBin(&H30)
  s5 = x3
 Call DecToBin(&HC0)
  s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
  Out 890, 12
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = False
  Label3.Visible = True
  Picture4.Visible = True
  Picture5.Visible = False
  Picture6.Visible = False
  Picture7.Visible = True
 Else
  Out 890, 5
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = True
  Label3.Visible = False
  Picture4.Visible = False
  Picture5.Visible = True
  Picture6.Visible = True
  Picture7.Visible = False
 End If
```

```
End If
If Combo1.ListIndex = 4 Then
 Call DecToBin(&H3)
 s3 = x1
 Call DecToBin(&HC)
 s4 = x2
 Call DecToBin(&H30)
 s5 = x3
 Call DecToBin(&HC0)
 s6 = x4
 If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
```

```
 End If
End If
If Combo1.ListIndex = 5 Then
 Call DecToBin(&H3)
 s3 = x1
 Call DecToBin(&HC)
 s4 = x2
 Call DecToBin(&H30)
 s5 = x3
 Call DecToBin(&HC0)
 s6 = x4
 If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
```

```
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 6 Then
 Call DecToBin(&H3)
 s3 = x1
 Call DecToBin(&HC)
 s4 = x2
 Call DecToBin(&H30)
 s5 = x3
 Call DecToBin(&HC0)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
```

```
 Picture6.Visible = True
 Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 7 Then
 Call DecToBin(&H3)
 s3 = x1
 Call DecToBin(&HC)
 s4 = x2
 Call DecToBin(&H30)
 s5 = x3
 Call DecToBin(&HC0)
 s6 = x4
 If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
 Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
```

```
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 8 Then
 Call DecToBin(&H3)
 s3 = x1
 Call DecToBin(&HC)
 s4 = x2
 Call DecToBin(&H30)
 s5 = x3
 Call DecToBin(&HC0)
 s6 = x4
 If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
 Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
```

```
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 9 Then
 Call DecToBin(&H3)
 s3 = x1
 Call DecToBin(&HC)
 s4 = x2
 Call DecToBin(&H30)
 s5 = x3
 Call DecToBin(&HC0)
 s6 = x4
 If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
 Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
```

```
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
End Sub

Private Sub Send10_Click()
 Call DecToBin(1)
 s7 = x5
 If s7 = 0 Then
 Show
 choice = MsgBox("Hardware Not Found !", vbRetryCancel + vbExclamation,
"Hardware Error")
 If choice = 4 Then
  Call Form_Load
 End If
 If choice = 2 Then
  Out 888, 4
  Out 890, 12
  End
 End If
End If
  Text1.Text = "1"
  Text2.Text = "0"
  Text3.Text = "1"
  Text4.Text = "0"
  Text5.Text = "1"
  Text6.Text = "0"
  Text7.Text = "1"
  Text8.Text = "0"
If Combo1.ListIndex = 0 Then
```

```
Call DecToBin(&HFE)
 s3 = x1
Call DecToBin(&HFB)
 s4 = x2
Call DecToBin(&HEF)
 s5 = x3
Call DecToBin(&HBF)
 s6 = x4
If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
End If
End If
```

```
If Combo1.ListIndex = 1 Then
 Call DecToBin(&HFE)
 s3 = x1
 Call DecToBin(&HFB)
 s4 = x2
 Call DecToBin(&HEF)
 s5 = x3
 Call DecToBin(&HBF)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
 Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
```

```
End If
If Combo1.ListIndex = 2 Then
 Call DecToBin(&HFE)
  s3 = x1
 Call DecToBin(&HFB)
  s4 = x2
 Call DecToBin(&HEF)
  s5 = x3
 Call DecToBin(&HBF)
  s6 = x4
 If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
  Out 890, 12
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = False
  Label3.Visible = True
  Picture4.Visible = True
  Picture5.Visible = False
  Picture6.Visible = False
  Picture7.Visible = True
 Else
  Out 890, 5
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = True
  Label3.Visible = False
  Picture4.Visible = False
  Picture5.Visible = True
  Picture6.Visible = True
  Picture7.Visible = False
```

```
 End If
End If
If Combo1.ListIndex = 3 Then
 Call DecToBin(&HFE)
  s3 = x1
 Call DecToBin(&HFB)
  s4 = x2
 Call DecToBin(&HEF)
  s5 = x3
 Call DecToBin(&HBF)
  s6 = x4
 If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
  Out 890, 12
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = False
  Label3.Visible = True
  Picture4.Visible = True
  Picture5.Visible = False
  Picture6.Visible = False
  Picture7.Visible = True
 Else
  Out 890, 5
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = True
  Label3.Visible = False
  Picture4.Visible = False
  Picture5.Visible = True
  Picture6.Visible = True
```

```
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 4 Then
 Call DecToBin(&HFE)
 s3 = x1
 Call DecToBin(&HFB)
 s4 = x2
 Call DecToBin(&HEF)
 s5 = x3
 Call DecToBin(&HBF)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
 Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
```

```
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 5 Then
 Call DecToBin(&HFE)
 s3 = x1
 Call DecToBin(&HFB)
 s4 = x2
 Call DecToBin(&HEF)
 s5 = x3
 Call DecToBin(&HBF)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
 Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
```

```
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 6 Then
 Call DecToBin(&HFE)
 s3 = x1
 Call DecToBin(&HFB)
 s4 = x2
 Call DecToBin(&HEF)
 s5 = x3
 Call DecToBin(&HBF)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
```

```
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 7 Then
 Call DecToBin(&HFE)
 s3 = x1
 Call DecToBin(&HFB)
 s4 = x2
 Call DecToBin(&HEF)
 s5 = x3
 Call DecToBin(&HBF)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
```

```
Label3.Visible = False
Picture4.Visible = False
Picture5.Visible = True
Picture6.Visible = True
Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 8 Then
Call DecToBin(&HFE)
s3 = x1
Call DecToBin(&HFB)
s4 = x2
Call DecToBin(&HEF)
s5 = x3
Call DecToBin(&HBF)
s6 = x4
If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
Out 890, 12
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = False
Label3.Visible = True
Picture4.Visible = True
Picture5.Visible = False
Picture6.Visible = False
Picture7.Visible = True
Else
Out 890, 5
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
```

```
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 9 Then
 Call DecToBin(&HFE)
 s3 = x1
 Call DecToBin(&HFB)
 s4 = x2
 Call DecToBin(&HEF)
 s5 = x3
 Call DecToBin(&HBF)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
  Out 890, 12
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = False
  Label3.Visible = True
  Picture4.Visible = True
  Picture5.Visible = False
  Picture6.Visible = False
  Picture7.Visible = True
 Else
  Out 890, 5
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
```

```
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
End Sub

Private Sub Send01_Click()
 Call DecToBin(1)
 s7 = x5
If s7 = 0 Then
 Show
 choice = MsgBox("Hardware Not Found !", vbRetryCancel + vbExclamation,
"Hardware Error")
 If choice = 4 Then
  Call Form_Load
 End If
 If choice = 2 Then
  Out 888, 4
  Out 890, 12
  End
 End If
End If
  Text1.Text = "0"
  Text2.Text = "1"
  Text3.Text = "0"
  Text4.Text = "1"
  Text5.Text = "0"
  Text6.Text = "1"
  Text7.Text = "0"
```

```
  Text8.Text = "1"
 If Combo1.ListIndex = 0 Then
  Call DecToBin(&HFD)
  s3 = x1
  Call DecToBin(&HF7)
  s4 = x2
  Call DecToBin(&HDF)
  s5 = x3
  Call DecToBin(&H7F)
  s6 = x4
  If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
  Out 890, 12
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = False
  Label3.Visible = True
  Picture4.Visible = True
  Picture5.Visible = False
  Picture6.Visible = False
  Picture7.Visible = True
 Else
  Out 890, 5
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = True
  Label3.Visible = False
  Picture4.Visible = False
  Picture5.Visible = True
  Picture6.Visible = True
  Picture7.Visible = False
```

```
 End If
End If
If Combo1.ListIndex = 1 Then
 Call DecToBin(&HFD)
 s3 = x1
 Call DecToBin(&HF7)
 s4 = x2
 Call DecToBin(&HDF)
 s5 = x3
 Call DecToBin(&H7F)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
 Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
```

```
 Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 2 Then
 Call DecToBin(&HFD)
 s3 = x1
 Call DecToBin(&HF7)
 s4 = x2
 Call DecToBin(&HDF)
 s5 = x3
 Call DecToBin(&H7F)
 s6 = x4
 If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
  Out 890, 12
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = False
  Label3.Visible = True
  Picture4.Visible = True
  Picture5.Visible = False
  Picture6.Visible = False
  Picture7.Visible = True
 Else
  Out 890, 5
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = True
  Label3.Visible = False
  Picture4.Visible = False
  Picture5.Visible = True
```

```
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 3 Then
 Call DecToBin(&HFD)
 s3 = x1
 Call DecToBin(&HF7)
 s4 = x2
 Call DecToBin(&HDF)
 s5 = x3
 Call DecToBin(&H7F)
 s6 = x4
 If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
  Out 890, 12
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = False
  Label3.Visible = True
  Picture4.Visible = True
  Picture5.Visible = False
  Picture6.Visible = False
  Picture7.Visible = True
 Else
  Out 890, 5
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = True
  Label3.Visible = False
  Picture4.Visible = False
```

```
Picture5.Visible = True
Picture6.Visible = True
Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 4 Then
Call DecToBin(&HFD)
s3 = x1
Call DecToBin(&HF7)
s4 = x2
Call DecToBin(&HDF)
s5 = x3
Call DecToBin(&H7F)
s6 = x4
If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
Out 890, 12
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = False
Label3.Visible = True
Picture4.Visible = True
Picture5.Visible = False
Picture6.Visible = False
Picture7.Visible = True
Else
Out 890, 5
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = True
Label3.Visible = False
```

```
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 5 Then
 Call DecToBin(&HFD)
 s3 = x1
 Call DecToBin(&HF7)
 s4 = x2
 Call DecToBin(&HDF)
 s5 = x3
 Call DecToBin(&H7F)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
 Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
```

```
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 6 Then
 Call DecToBin(&HFD)
 s3 = x1
 Call DecToBin(&HF7)
 s4 = x2
 Call DecToBin(&HDF)
 s5 = x3
 Call DecToBin(&H7F)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
 Else
 Out 890, 5
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
```

```
Label2.Visible = True
Label3.Visible = False
Picture4.Visible = False
Picture5.Visible = True
Picture6.Visible = True
Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 7 Then
Call DecToBin(&HFD)
s3 = x1
Call DecToBin(&HF7)
s4 = x2
Call DecToBin(&HDF)
s5 = x3
Call DecToBin(&H7F)
s6 = x4
If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
Out 890, 12
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = False
Label3.Visible = True
Picture4.Visible = True
Picture5.Visible = False
Picture6.Visible = False
Picture7.Visible = True
Else
Out 890, 5
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
```

```
Text12.Text = s6
Label2.Visible = True
Label3.Visible = False
Picture4.Visible = False
Picture5.Visible = True
Picture6.Visible = True
Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 8 Then
Call DecToBin(&HFD)
s3 = x1
Call DecToBin(&HF7)
s4 = x2
Call DecToBin(&HDF)
s5 = x3
Call DecToBin(&H7F)
s6 = x4
If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
Out 890, 12
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = False
Label3.Visible = True
Picture4.Visible = True
Picture5.Visible = False
Picture6.Visible = False
Picture7.Visible = True
Else
Out 890, 5
Text9.Text = s3
Text10.Text = s4
```

```
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 9 Then
 Call DecToBin(&HFD)
 s3 = x1
 Call DecToBin(&HF7)
 s4 = x2
 Call DecToBin(&HDF)
 s5 = x3
 Call DecToBin(&H7F)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
Else
 Out 890, 5
 Text9.Text = s3
```

```
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
End Sub

Private Sub Send00_Click()
 Call DecToBin(1)
 s7 = x5
 If s7 = 0 Then
 Show
 choice = MsgBox("Hardware Not Found !", vbRetryCancel + vbExclamation,
"Hardware Error")
 If choice = 4 Then
  Call Form_Load
 End If
 If choice = 2 Then
  Out 888, 4
  Out 890, 12
  End
 End If
 End If
  Text1.Text = "0"
  Text2.Text = "0"
  Text3.Text = "0"
  Text4.Text = "0"
  Text5.Text = "0"
```

```
Text6.Text = "0"
Text7.Text = "0"
Text8.Text = "0"
If Combo1.ListIndex = 0 Then
Call DecToBin(&HFC)
s3 = x1
Call DecToBin(&HF3)
s4 = x2
Call DecToBin(&HCF)
s5 = x3
Call DecToBin(&H3F)
s6 = x4
If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
Out 890, 12
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = False
Label3.Visible = True
Picture4.Visible = True
Picture5.Visible = False
Picture6.Visible = False
Picture7.Visible = True
Else
Out 890, 5
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = True
Label3.Visible = False
Picture4.Visible = False
Picture5.Visible = True
```

```
  Picture6.Visible = True
  Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 1 Then
 Call DecToBin(&HFC)
 s3 = x1
 Call DecToBin(&HF3)
 s4 = x2
 Call DecToBin(&HCF)
 s5 = x3
 Call DecToBin(&H3F)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
  Out 890, 12
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = False
  Label3.Visible = True
  Picture4.Visible = True
  Picture5.Visible = False
  Picture6.Visible = False
  Picture7.Visible = True
 Else
  Out 890, 5
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = True
  Label3.Visible = False
  Picture4.Visible = False
```

```
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 2 Then
 Call DecToBin(&HFC)
 s3 = x1
 Call DecToBin(&HF3)
 s4 = x2
 Call DecToBin(&HCF)
 s5 = x3
 Call DecToBin(&H3F)
 s6 = x4
 If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
  Out 890, 12
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = False
  Label3.Visible = True
  Picture4.Visible = True
  Picture5.Visible = False
  Picture6.Visible = False
  Picture7.Visible = True
 Else
  Out 890, 5
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
  Text12.Text = s6
  Label2.Visible = True
  Label3.Visible = False
```

```
Picture4.Visible = False
Picture5.Visible = True
Picture6.Visible = True
Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 3 Then
Call DecToBin(&HFC)
s3 = x1
Call DecToBin(&HF3)
s4 = x2
Call DecToBin(&HCF)
s5 = x3
Call DecToBin(&H3F)
s6 = x4
If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
Out 890, 12
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = False
Label3.Visible = True
Picture4.Visible = True
Picture5.Visible = False
Picture6.Visible = False
Picture7.Visible = True
Else
Out 890, 5
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = True
```

```
Label3.Visible = False
Picture4.Visible = False
Picture5.Visible = True
Picture6.Visible = True
Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 4 Then
Call DecToBin(&HFC)
s3 = x1
Call DecToBin(&HF3)
s4 = x2
Call DecToBin(&HCF)
s5 = x3
Call DecToBin(&H3F)
s6 = x4
If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
Out 890, 12
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = False
Label3.Visible = True
Picture4.Visible = True
Picture5.Visible = False
Picture6.Visible = False
Picture7.Visible = True
Else
Out 890, 5
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
```

```
    Label2.Visible = True
    Label3.Visible = False
    Picture4.Visible = False
    Picture5.Visible = True
    Picture6.Visible = True
    Picture7.Visible = False
   End If
  End If
  If Combo1.ListIndex = 5 Then
   Call DecToBin(&HFC)
   s3 = x1
   Call DecToBin(&HF3)
   s4 = x2
   Call DecToBin(&HCF)
   s5 = x3
  Call DecToBin(&H3F)
   s6 = x4
  If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
   Out 890, 12
   Text9.Text = s3
   Text10.Text = s4
   Text11.Text = s5
   Text12.Text = s6
   Label2.Visible = False
   Label3.Visible = True
   Picture4.Visible = True
   Picture5.Visible = False
   Picture6.Visible = False
   Picture7.Visible = True
  Else
  Out 890, 5
  Text9.Text = s3
  Text10.Text = s4
  Text11.Text = s5
```

```
Text12.Text = s6
Label2.Visible = True
Label3.Visible = False
Picture4.Visible = False
Picture5.Visible = True
Picture6.Visible = True
Picture7.Visible = False
End If
End If
If Combo1.ListIndex = 6 Then
Call DecToBin(&HFC)
s3 = x1
Call DecToBin(&HF3)
s4 = x2
Call DecToBin(&HCF)
s5 = x3
Call DecToBin(&H3F)
s6 = x4
If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
Out 890, 12
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = False
Label3.Visible = True
Picture4.Visible = True
Picture5.Visible = False
Picture6.Visible = False
Picture7.Visible = True
Else
Out 890, 5
Text9.Text = s3
Text10.Text = s4
```

```
        Text11.Text = s5
        Text12.Text = s6
        Label2.Visible = True
        Label3.Visible = False
        Picture4.Visible = False
        Picture5.Visible = True
        Picture6.Visible = True
        Picture7.Visible = False
      End If
    End If
    If Combo1.ListIndex = 7 Then
     Call DecToBin(&HFC)
     s3 = x1
    Call DecToBin(&HF3)
     s4 = x2
    Call DecToBin(&HCF)
     s5 = x3
    Call DecToBin(&H3F)
     s6 = x4
    If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
    Out 890, 12
    Text9.Text = s3
    Text10.Text = s4
    Text11.Text = s5
    Text12.Text = s6
    Label2.Visible = False
    Label3.Visible = True
    Picture4.Visible = True
    Picture5.Visible = False
    Picture6.Visible = False
    Picture7.Visible = True
  Else
  Out 890, 5
  Text9.Text = s3
```

```
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 8 Then
 Call DecToBin(&HFC)
 s3 = x1
 Call DecToBin(&HF3)
 s4 = x2
 Call DecToBin(&HCF)
 s5 = x3
 Call DecToBin(&H3F)
 s6 = x4
 If (s3 = 1) And (s4 = 1) And (s5 = 1) And (s6 = 1) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
 Else
 Out 890, 5
```

```
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = True
 Label3.Visible = False
 Picture4.Visible = False
 Picture5.Visible = True
 Picture6.Visible = True
 Picture7.Visible = False
 End If
End If
If Combo1.ListIndex = 9 Then
 Call DecToBin(&HFC)
 s3 = x1
 Call DecToBin(&HF3)
 s4 = x2
 Call DecToBin(&HCF)
 s5 = x3
 Call DecToBin(&H3F)
 s6 = x4
 If (s3 = 0) And (s4 = 0) And (s5 = 0) And (s6 = 0) Then
 Out 890, 12
 Text9.Text = s3
 Text10.Text = s4
 Text11.Text = s5
 Text12.Text = s6
 Label2.Visible = False
 Label3.Visible = True
 Picture4.Visible = True
 Picture5.Visible = False
 Picture6.Visible = False
 Picture7.Visible = True
 Else
```

```
Out 890, 5
Text9.Text = s3
Text10.Text = s4
Text11.Text = s5
Text12.Text = s6
Label2.Visible = True
Label3.Visible = False
Picture4.Visible = False
Picture5.Visible = True
Picture6.Visible = True
Picture7.Visible = False
End If
End If
End Sub

Private Sub Send11_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
Picture1.MousePointer = 99
End Sub

Private Sub Send10_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
Picture1.MousePointer = 99
End Sub

Private Sub Send01_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
Picture1.MousePointer = 99
End Sub

Private Sub Send00_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
Picture1.MousePointer = 99
End Sub
```

```
Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture1.MousePointer = 1
End Sub

Private Sub Picture2_Click()
 Out 890, 12
 Out 888, 4
 End
End Sub

Private Sub Picture2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture2.PaintPicture Picture2.Picture, 0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight,
0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight, &H550009
End Sub

Private Sub Picture2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture1.MousePointer = 99
 Picture2.MousePointer = 99
End Sub

Private Sub Picture2_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
 Picture2.PaintPicture Picture2.Picture, 0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight,
0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight, &H550009
End Sub

Private Sub Picture8_Click()
 About.Show
End Sub
```

```
Private Sub Picture8_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture8.PaintPicture Picture8.Picture, 0, 0, Picture8.ScaleWidth, Picture8.ScaleHeight,
0, 0, Picture8.ScaleWidth, Picture8.ScaleHeight, &H550009
End Sub

Private Sub Picture8_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture1.MousePointer = 99
 Picture8.MousePointer = 99
End Sub

Private Sub Picture8_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
 Picture8.PaintPicture Picture8.Picture, 0, 0, Picture8.ScaleWidth, Picture8.ScaleHeight,
0, 0, Picture8.ScaleWidth, Picture8.ScaleHeight, &H550009
End Sub

Private Sub Picture11_Click()
 info.Show
End Sub

Private Sub Picture11_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture11.PaintPicture Picture11.Picture, 0, 0, Picture11.ScaleWidth,
Picture11.ScaleHeight, 0, 0, Picture11.ScaleWidth, Picture11.ScaleHeight, &H550009
End Sub

Private Sub Picture11_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture1.MousePointer = 99
 Picture11.MousePointer = 99
End Sub
```

```
Private Sub Picture11_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture11.PaintPicture Picture11.Picture, 0, 0, Picture11.ScaleWidth,
Picture11.ScaleHeight, 0, 0, Picture11.ScaleWidth, Picture11.ScaleHeight, &H550009
End Sub
```

APPENDIX B

Information Form Listing

```
Private Sub Picture1_Click()
 Unload Me
End Sub

Private Sub Picture1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture1.PaintPicture Picture1.Picture, 0, 0, Picture1.ScaleWidth, Picture1.ScaleHeight,
0, 0, Picture1.ScaleWidth, Picture1.ScaleHeight, &H550009
End Sub

Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture1.MousePointer = 99
End Sub

Private Sub Picture1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
 Picture1.PaintPicture Picture1.Picture, 0, 0, Picture1.ScaleWidth, Picture1.ScaleHeight,
0, 0, Picture1.ScaleWidth, Picture1.ScaleHeight, &H550009
End Sub
```

APPENDIX C

About Form Listing

```
Private Sub Picture2_Click()
 Unload Me
End Sub

Private Sub Picture2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture2.PaintPicture Picture2.Picture, 0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight,
0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight, &H550009
End Sub

Private Sub Picture2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture2.MousePointer = 99
End Sub

Private Sub Picture2_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
 Picture2.PaintPicture Picture2.Picture, 0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight,
0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight, &H550009
End Sub

Private Sub Picture3_Click()
 Details.Show
End Sub

Private Sub Picture3_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture3.PaintPicture Picture3.Picture, 0, 0, Picture3.ScaleWidth, Picture3.ScaleHeight,
0, 0, Picture3.ScaleWidth, Picture3.ScaleHeight, &H550009
End Sub
```

```
Private Sub Picture3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture3.MousePointer = 99
End Sub

Private Sub Picture3_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
 Picture3.PaintPicture Picture3.Picture, 0, 0, Picture3.ScaleWidth, Picture3.ScaleHeight,
0, 0, Picture3.ScaleWidth, Picture3.ScaleHeight, &H550009
 End Sub
```

APPENDIX D

Details Form Listing

```
Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture1.MousePointer = 1
End Sub

Private Sub Picture2_Click()
 Unload Me
End Sub

Private Sub Picture2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture2.PaintPicture Picture2.Picture, 0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight,
0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight, &H550009
End Sub

Private Sub Picture2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
 Picture1.MousePointer = 99
 Picture2.MousePointer = 99
End Sub

Private Sub Picture2_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
 Picture2.PaintPicture Picture2.Picture, 0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight,
0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight, &H550009
End Sub
```

APPENDIX E

Porttester Program Listing

```
Dim x, x0, x1, x2, x3, x4, x5, x6, x7 As Integer
Private Sub Take_port(xr As Integer)
x = Inp(xr)
If x >= 128 Then
x7 = 1
x = x - 128
Else
x7 = 0
End If
If x >= 64 Then
x6 = 1
x = x - 64
Else
x6 = 0
End If
If x >= 32 Then
x5 = 1
x = x - 32
Else
x5 = 0
End If
If x >= 16 Then
x4 = 1
x = x - 16
Else
x4 = 0
End If
If x >= 8 Then
x3 = 1
x = x - 8
```

```
Else
x3 = 0
End If
If x >= 4 Then
x2 = 1
x = x - 4
Else
x2 = 0
End If
If x >= 2 Then
x1 = 1
x = x - 2
Else
x1 = 0
End If

If x >= 1 Then
x0 = 1
x = x - 1
Else
x0 = 0
End If
End Sub

Private Sub Command1_Click()
Call Take_port(888)
Text1.Text = x0
Text2.Text = x1
Text3.Text = x2
Text4.Text = x3
Text5.Text = x4
Text6.Text = x5
Text7.Text = x6
Text8.Text = x7
```

```
Call Take_port(889)
Text9.Text = x0
Text10.Text = x1
Text11.Text = x2
Text12.Text = x3
Text13.Text = x4
Text14.Text = x5
Text15.Text = x6
Text16.Text = x7
Call Take_port(890)
Text17.Text = x0
Text18.Text = x1
Text19.Text = x2
Text20.Text = x3
Text21.Text = x4
Text22.Text = x5
Text23.Text = x6
Text24.Text = x7
End Sub

Private Sub Command2_Click()
End
End Sub

Private Sub Form_Load()
Text1.Text = "0"
Text2.Text = "0"
Text3.Text = "0"
Text4.Text = "0"
Text5.Text = "0"
Text6.Text = "0"
Text7.Text = "0"
Text8.Text = "0"
Text9.Text = "0"
```

```
Text10.Text = "0"
Text11.Text = "0"
Text12.Text = "0"
Text13.Text = "0"
Text14.Text = "0"
Text15.Text = "0"
Text16.Text = "0"
Text17.Text = "0"
Text18.Text = "0"
Text19.Text = "0"
Text20.Text = "0"
Text21.Text = "0"
Text22.Text = "0"
Text23.Text = "0"
Text24.Text = "0"
End Sub
```

APPENDIX F

Module Listing

```
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```