Interfacing Parallel Port Using Turbo C Programming

Kansham Modar Maring[*], Kansham Angphun Maring[**]

[*] Scientific Officer, NIC, DeitY, Government of India
[**] Department of Computer Science and Engineering, NIT Jalandhar, Punjab, India

Abstract- Hardware Interfacing Using Parallel Port Programming is designed to control electrical appliances using computer printer port. The electrical appliances are controlled by using software through an interfacing circuit. The software used in this is TURBO –C. A windows based program is used to control eight LEDs, through an interfacing circuit, by using computer printer port. The base address used is LTP1 (0378 H). The program accepts input signal from the keyboard, decodes the keyboard hit and accordingly send signal to the corresponding pins of the printer port which in turn controls the LEDs. It is further extended to the automatic control i.e. in random or sequential order of the LEDs. The LEDs will be controlled automatically according to some delays provided in the program. The purpose of this work is to reduce human effort in controlling large number of switches as in stadiums, buildings, apartments etc. This effort can also be extended to controlling up to 256 electrical switches by the use of proper interfacing circuits. By proper implementation, 256 electrical switches can be controlled using only eight keys from the keyboard.

Index Terms- Parallel Port, DB25 Connector, Register, LED, Circuit, White Board, Base Address and LTP

I. INTRODUCTION

COM (Communication port) [1] is the original, yet still common, name of the serial port interface on IBM PC-compatible computers. It might refer not only to physical ports, but also to virtual ports, such as ports created by Bluetooth or USB-to-serial adapters. Most PC-compatible computers in the 1980s and 1990s had one or two COM ports. As of 2007, most computers ship with one or no physical COM ports. As of 2014, most PC-compatible computers don't include any COM ports. After the RS-232 COM port was removed from most IBM PC compatible computers in the 2000s, an external USB-to-UART serial adapter cable was used to compensate for the loss. A major supplier of these chips is FTDI.[2] . In computers, ports are used mainly for two reasons: Device control and communication. We can program PC's Parallel ports for both. Parallel ports are mainly meant for connecting the printer to the PC. But we can program this port for many more applications beyond that Parallel ports are easy to program and faster compared to the serial ports. But main disadvantage is it needs more number of transmission lines. Because of this reason parallel ports are not used in long distance communications. In serial ports, there will be two data lines: One transmission and one receive line. To send a data in serial port, it has to be sent one bit after another with some extra bits like start bit, stop bit and parity bit to detect errors. But in parallel port, all the 8 bits of a byte will be sent to the port at a time and an indication will be sent in another line. There will be some data lines, some control and some handshaking lines in parallel port. Computer places the data in the data pins, and then it makes the strobe low. When strobe goes low, printer understands that there is a valid data in data pins. Other pins are used to send controls to the printer and get status of the printer. To use the printer port for applications other than printing, we need to know how ports are organized. There are three registers associated with LPT port: Data register, Control register and Status register. Data register will hold the data of the data pins of the port. That means, if we store a byte of data to the data register, that data will be sent to the data pins of the port.

II. LITERATURE SURVEY

The first parallel interface for printers was introduced with the Centronics Model 101 printer in 1970 [3] . The early calculators, which they used to create the Centronics interface on their computers, had 36-pin micro ribbon connectors Wang etal. The connector has become so closely associated with Centronics that it is now popularly known as the "Centronics connector". The Centronics parallel interface quickly became an industry *de facto* standard; manufacturers of the time tended to use various connectors on the system side, so a variety of cables were required. For example, early VAX systems used a DC-37 connector, NCR used the 36-pin micro ribbon connector, Texas Instruments used a 25-pin card edge connector and Data General used a 50-pin micro ribbon connector. Before the advent of USB, the parallel interface was adapted to access a number of peripheral devices other than printers. Probably one of the earliest devices to use parallel was dongles used as a hardware key form of software copy protection. Zip drives and scanners were early implementations followed by external modems, sound cards, webcams, gamepads, joysticks, external hard disk drives and CD-ROM drives. Some of the earliest portable MP3 players required a parallel port connection for transferring songs to the device [4]. There are also some print servers that provide interface to parallel port through network. USB-to-EPP chips can also allow other non-printer device to continue to work on modern computers without a parallel port [5]. On almost all the PCs only one parallel port is present, but we can add more by buying and inserting ISA/PCI parallel port cards. The Pin outs of DB25 connector is as shown in figure 1 below.

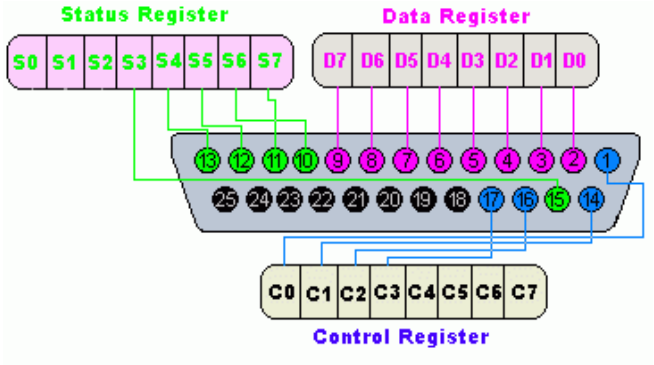

Figure 1: Pin-Out diagram of DB25 connector

The LEDs can be controlled by using simple interfacing circuit. The only component needed is one LED and one 470 ohm registers. In this circuit one end of the circuit is connected to the data pins from where the signal is to be sent to the LEDs and another end goes to the circuit as shown in the figure 2 below:

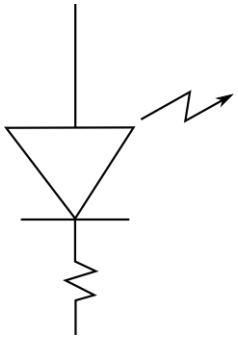

Figure 2: A Simple LED

In this hardware interfacing, we are using only first parallel port (LPT1); the following table lists the details of how the software interfaces to hardware port.

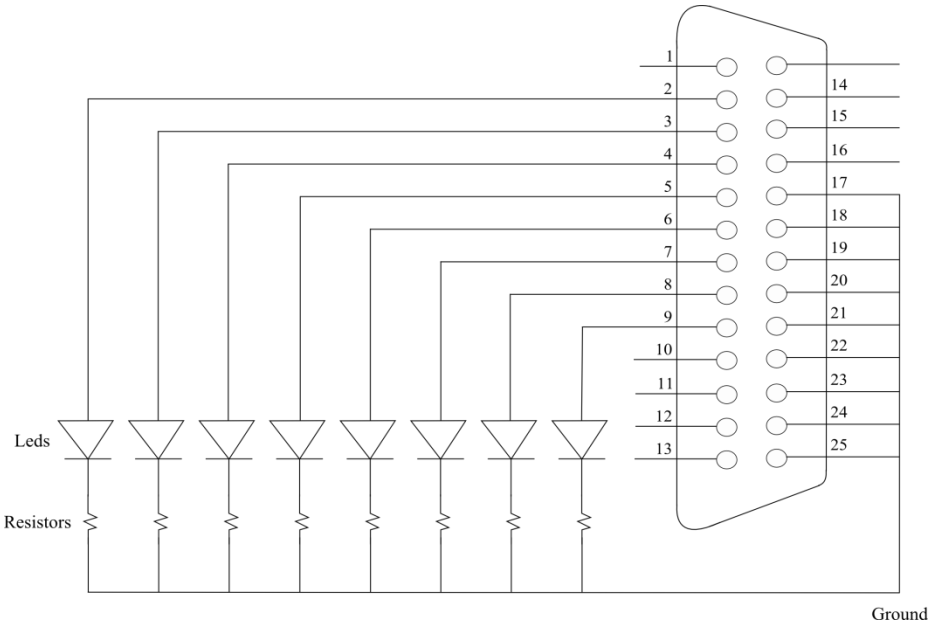

Figure 3: Circuit diagram used for controlling the LEDs

Table I: Printer Port and Address

Printer Port Addresses	
Printer Port	Base Address
LPT1	0x0378 or 0x03BC
LPT2	0x0278 or 0x378
LPT3	0x0278

Table II: Printer Port Registers

Printer Port Registers	
Register Name	Address
Data Register	Base+0x00
Status Register	Base+0x01
Control Register	Base+0x02

Table III: Registers bit definitions

Register Bit Definition									
Bit	Data Register	Status Register	Control Register	Data Low	Data High	Status Low	Status High	Control Low	Control High
7	D7	Busy	Not Used	0	1	Busy	Not Busy	-	-
6	D6	Acknowledge	Not Used	0	1	Nack	Ack	-	-
5	D5	Paper Status	Not Used	0	1	No Paper	Paper	-	-
4	D4	Selection Status	Interrupt	0	1	Not Selected	Selected	Interrupt Disabled	Interrupt Enabled
3	D3	Error Status	Select	0	1	No Error	Error	Selected	Not Selected
2	D2	Not Used	Initialize	0	1	-	-	False	True
1	D1	Not Used	Auto Feed	0	1	-	-	True	False
0	D0	Not Used	Strobe	0	1	-	-	True	False

III. SIMULATION RESULTS

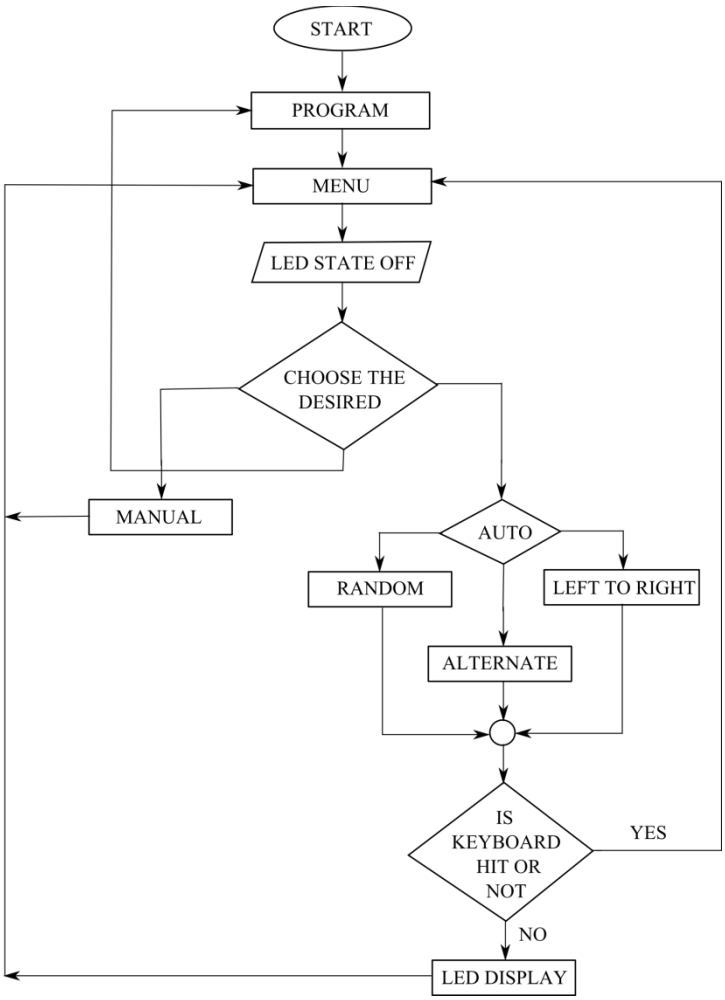

Figure 4. Flowchart showing the control flows

When the program is executed, a menu driven screen with the following options for selecting the choice for operation of LEDs is displayed.

a. Manual
b. Automatic
c. Exit

In the manual portion, the LEDs are lit as we press the corresponding key. In automatic three more options are made available i.e. Random, Left to Right and Alternate. In random mode, the LEDs are lit up in a random manner. In the left to right mode, the LEDs are lit up from left to right. In the alternate mode, the operation is made in an alternate fashion at a time. The flow chart of the program is as shown in figure 4.

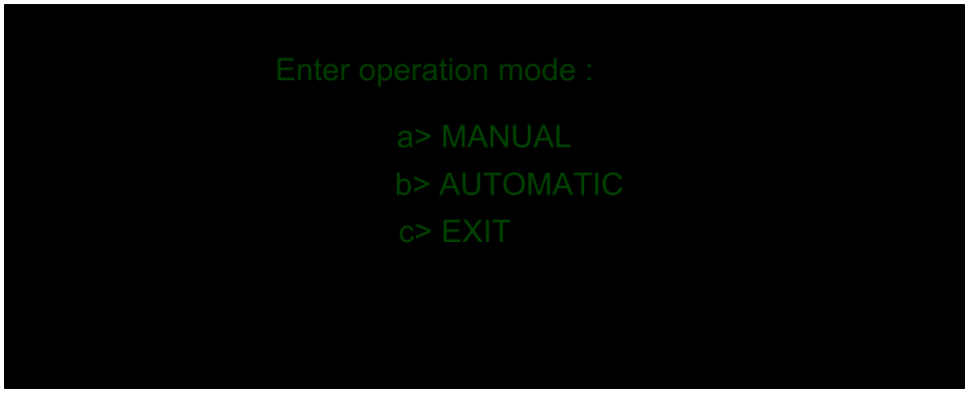

Figure 5. Snapshot of the Main Menu for Program operation

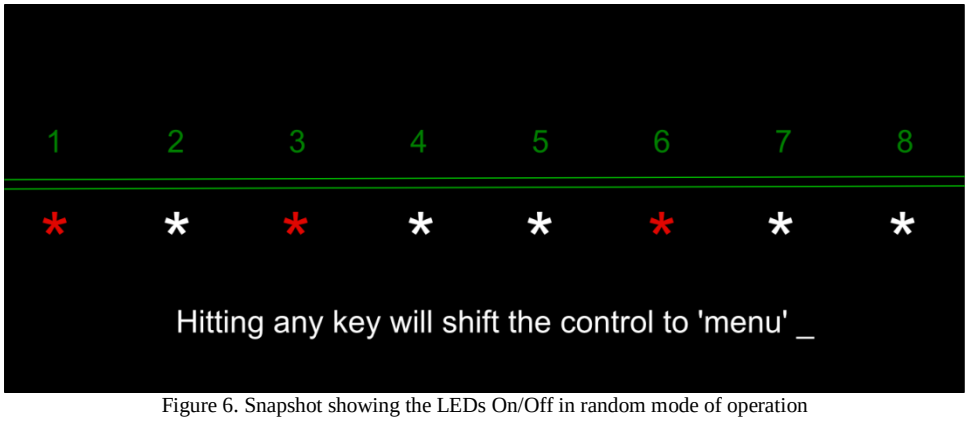

Figure 6. Snapshot showing the LEDs On/Off in random mode of operation

IV. CONCLUSIONS

The control of household equipment can be easily performed by replacing these LEDs with the household electrical appliance and also similarly in the case of big buildings like stadiums, offices etc., which will reduce effort and time and increase safety.

V. FUTURE SCOPE

The current model concerned with only controlling 8 LEDs, but it can be used to control as many as 256 electrical appliances by using decoder. It can be extended to control switching system of offices, apartments, and stadiums etc., where there is the need of controlling large number of electrical appliances.

ACKNOWLEDGMENT

We would like to show our heartfelt gratitude to **Mr. Marjit Singh**, Assistant Professor, Department of Computer Science and Engineering, **NERIST**, Nirjuli, Papum-Pare, Itanagar, Arunachal Pradesh, India-791109 for his immense help, during our entire work to complete this work..

REFERENCES

[1] IBM Lotus Domino and Notes Information Center, *Configuring a communication port*, Retrieved 17 September 2013
[2] Jan Axelson, *Lakeview Research: Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems*, 2nd Edition, 380 pages; 2007; ISBN 978-1-931-44806-2.

[3] Webster, Edward C. (2000). *Print Unchained: Fifty Years of Digital Printing: A Saga of Invention and Enterprise*. West Dover, VT: DRA of Vermont. ISBN 0-9702617-0-5.
[4] Mitskaniouk, Oleg (2000-06-19). *The D-Link DMP-100 MP3 Player*, Target PC Magazine. p. 2. Retrieved 2012-07-20.
[5] *Parallel port flatbed scanner works under USB on Win9x (Archive)*, Retrieved 2012-06-30.
[6] H. Shan and J. Pal Singh, *A comparison of MPI, SHMEM, and Cache-Coherent Shared Address Space Programming Models on a Tightly-Coupled Multiprocessor*, International Journal of Parallel Programming, 29(3), 2001.
[7] H. Shan and J. Pal Singh, *Comparison of Three Programming Models for Adaptive Applications on the Origin 2000*, Journal of Parallel and Distributed Computing, 62:241–266, 2002.
[8] Davide Pasetto and Marco Vanneschi, *About structured parallel programming: Machine independent Analytical models for cost evaluation of template--based programs*, University of Pisa, 1996
[9] J. Darlinton, M. Ghanem, H. W. To, *Structured Parallel Programming in Programming Models for Massively Parallel Computers*, IEEE Computer Society Press. 1993.
[10] Murray I. Cole. *Algorithmic Skeletons: Structured Management of Parallel Computation*
[11] Byron S Gottfried, *Programming with C*, the Mc Graw Hill Company.
[12] Magazine, *Electronics for You*.

AUTHORS

First Author – **Kansham Modar Maring** did his B. Tech (CSE) and M.Tech (IT) at NERIST, Nirjuli, Arunachal Pradesh. He is presently working as Scientific Officer / Engineer at National Informatics Centre (NIC), DeitY, Government of India. His area interests are Networking, Web Designing and Computer Troubleshooting.
Email: modar.k@nic.in

Second Author – **Kansham Angphun Maring** did his B.Tech in Computer Science and Engineering at NERIST, Nirjuli, Arunachal Pradesh. He has completed his M.Tech in Computer Science and Engineering at Dr. B R Ambedkar National Institute of Technology, Jalandhar, Punjab in May 2014. His area of interest are Digital Image Processing, Pattern Recognition, NLP, and Wireless Communication.
Email: kam10maring@gmail.com