Nigerian Journal of Technology (NIJOTECH)
Vol. 31, No. 3, November, 2012, pp. 261–267.

DIGITAL CONTROL OF EXTERNAL DEVICES THROUGH THE PARALLEL PORT OF A COMPUTER USING VISUAL BASIC

P.E. Orukpe[a], A. Adesemowo[b]

[a]Department of Electrical/Electronic Engineering, University of Benin, Nigeria.
Email: *patience.orukpe01@imperial.ac.uk*
[b]Total Support Ltd, Nigeria.

Abstract

In this paper we carry out the digital control of external devices using the parallel port of a computer. The PC parallel port adapter that is specifically designed to attach printers has been found to be useful as a general input/output port for any device or application that matches its input/output capabilities. This paper gives details and involves the design and construction of the parallel port power switching device and its associated control program (Device Controller). The Unit is capable of simultaneously and sequentially powering ON/OFF eight external devices depending on the number of bits sent to the port. However, with the presence of overflow signal to four additional ports, the unit can power up to twelve external devices. The control program causes the computer port to generate an actuating signal (the controlling voltage of 5V) which drives an electrically isolated circuit with a relay. A model of the system was built and the test result was satisfactory.

Keywords: device controller, digital switching, digital interfacing, visual basic, computer parallel port

1. Introduction

The computer and its diverse applications have in recent times witnessed a revolution. Its enormous success is due largely to the flexibility and reliability that computer systems offer to potential users [1, 2, 3].

Domestic lifestyles have also been affected by interfacing digital processors into household appliances, thereby making them semi or fully automatic [4]. The appliances can be programmed to switch ON or OFF at appropriate time automatically. The computer acts as the controller and provides the enabling technology that allows one to design and implement the overall system so that satisfactory performance is obtained. Applications of computer control are varied; examples include Energy management, Industrial and process control and Load management as applicable to $N + 1$ philosophy in Power Station where N is the number of generating stations required at any point in time [3, 5].

However, it is worth noting that, digital control systems differ from continuous ones in that, the computer execute one operation at a time in a sequential manner [6]. Digital controllers have been used to give results as good as and even better than analogue controllers in numerous cases, with the added feature that the control strategies can be varied by simply reprogramming the computer instead of having to change the hardware [7, 8, 9]. In addition analogue controllers are susceptible to ageing and drifting, which in turn cause degradation in performance. Advantages like these have attracted many users to adopt digital technology in preference to conventional methods and consequently application areas are growing rapidly.

External devices have been controlled by various authors. In [10], experimental control language (ECL) was used and this enable researchers with tasks that require extremely short, worst-case delays but that are computational intensive to carry out. ECL forgoes the very simple and standardized syntax of BASIC in order to provide faster execution. In [11], the interface and compiler have been written in C, while the program directly controlling the counter/time board has been written in assembly language (MASM 6.0). Borland C++ builder was used in [12] for multimedia environment for teaching and learning interfaces. Position control of stepper motor using Macro Assembler MASM 86 was considered in [13]. In [14], seven segment displays and a DIP switch were controlled in the extended capabilities port mode and the hardware was programmed in Visual Basic. In this paper, our work follows closely the work in [14], however, we controlled various household devices using the standard parallel port. Also, our device controller can control up to twelve devices with the presence of signal overflow to four additional ports.

2. Basic Element of Computer Control

The basic functional block diagram of a computer control is as shown (Figure 1) and the circuit diagram of the computer control system is as shown (Figure 2). The various units are discussed in subsections 2.1 - 2.5.

2.1. Isolating unit

The isolating unit is made up of the following; switching unit (transistor, relay, a diode,) and optical isolator.

2.1.1. Switching unit

There is an imminent danger of damaging the computer if a load is connected directly to the I/O port. Therefore, a suitable interface is needed to enable computer control the power switching of loads. Such devices include, power transistor, which operates based on the output port control signal. The power transistor is used to switch a conventional electromechanical relay, which in turn switches on a load.

A load can be turned on or off by a bipolar junction transistor which acts as a switch. This is because the power level of the control signal provided by the parallel port of the computer is very small [15, 16].

2.1.2. Optical isolation

The external devices and the microprocessor-based system require a high degree of isolation, for this reason optical isolation or optocoupler [17] was used. To provide total isolation from high voltages, signals are coupled through an optoisolator between the output port of the computer and the power control device. The optoisolator negates the need for hardwired electrical connections and the signals are transmitted by means of a beam of infrared radiation emanating from an emitting device and sensed at a detector. The beam, in fact simply switches on a transistor at the output pin (4 and 5). It is purely a protective device which combats spurious noise signals, which can corrupt the digital logical values being transmitted on the busses [18].

2.2. Control Program

These are software (written program) used for controlling the communication between the microcomputer and other external devices. All computers have instructions concerned with the input and output of information from peripherals and output to the peripherals. An interface unit is needed to ensure compatibility between the bus and the peripherals. The present day PCs run under Windows (event-driven) operating system in an environment of multiple tasks. Visual Basic (VB) is a simple and powerful high-level language that is easy to learn with good resources such as books, web sites, news group etc. Also answers to programming problem can be easily found more than other programming languages. However, VB do not interact with system hardware directly but is useful in writing Windows based application. Thus, based on the advantages VB has been used in the implementation of the control program [19].

2.3. Input/Output Unit

The input/output interface is necessary for the computer to communicate with the outside world, that is, all of its peripheral equipment. Interface units are usually built into the computer itself and points on the interface unit where the peripheral devices are connected are often referred to as I/O ports. Several interface are used in practice today, however, in this development a specific type of port called general purpose parallel port or simply parallel port [20] was used. Typical uses of ports are, communicating with printers, modems, keyboards, and displays, or any component or device except system memory. It is found on every personal computer along with the RS-232 serial port [2].

2.3.1. Parallel port

Port is a set of signal lines (most computer ports are digital) that the microprocessor, or CPU, uses to exchange data with other components, this enables digital representation of a physical variable or the output of a control signal to switch a device on and off to be accomplished through the computer port.

The input/output signals of the parallel port are made available at the back of the adapter through a right-angled, PCB-mounted, 25-pin, D-type female connector, the connector protrudes through the rear panel of the system, where a cable may be attached.

The parallel port, as implemented on the PC, consists of a connector with 17 signal lines and 8 ground lines. The signal lines are divided into three groups: Control (4 lines) which interface control and handshaking signals, Status (5 lines), responsible for status indication such as paper empty, busy indication and interface or peripheral error and Data (8 lines), allowed for data to be driven from the peripheral to the PC. Pins 2 through 9 are used to carry data [2], [21].

2.3.2. Port addresses

Each printer port consists of three port addresses; data, status and control port. The data register re-

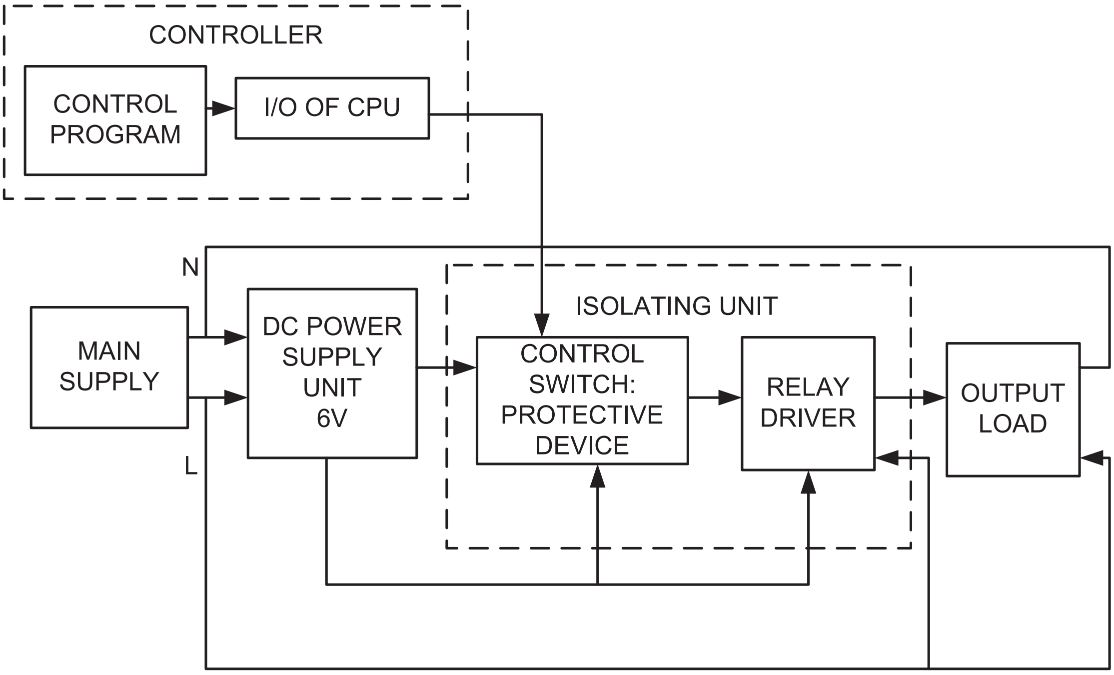

Figure 1: Block diagram of a computer control system.

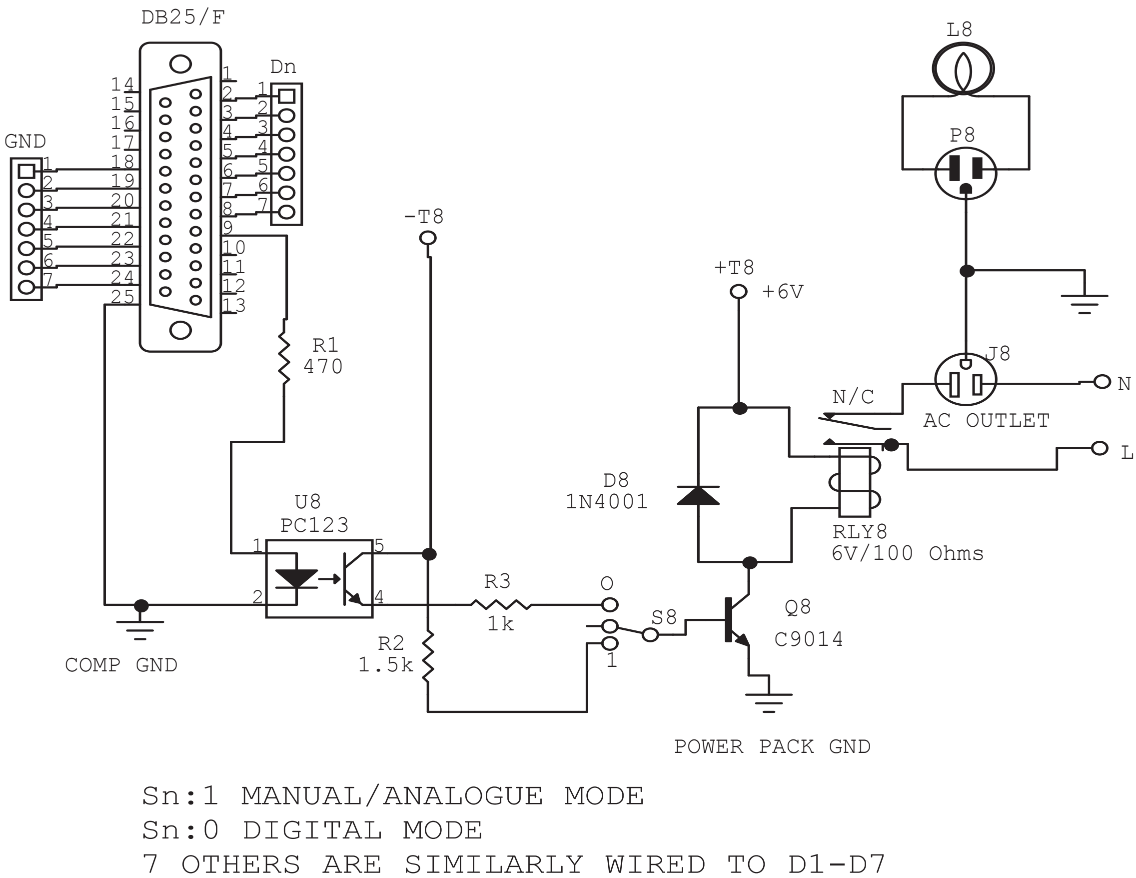

Figure 2: Circuit with optoisolation.

sides at this base address, status register at (base address + 1) and the control register is at (base address + 2). Therefore the addresses are in sequential order, that is, for a data port with address 0×0378, the corresponding status port is 0x0379 while the control port is at 0×037a address [2, 21, 22].

2.4. Digital interfacing

Data pins are Transistor Transistor Logic (TTL) level output pins. This means that they put out ideally 0V when they are in low logic level and +5V when they are in high logic level. Therefore a simple logical output is displayed using LED. To indicate that a bit has a value of 1, a charge of 5 volt is sent through the correct pin. No charge on a pin indicates a value of 0V. This is a simple but highly effective way to transmit digital information over an analogue cable in real-time. However it should be noted that the output current capacity of the parallel port is limited to only few milliamperes (2.6-20mA) [22]. The idea of the interface shown was expanded to control some external electronics [23] by simply adding a buffer circuit to the parallel port. Two concepts was given a consideration in the designs

i) Driving of controlling voltage from the computer
ii) The switching power of a transistor or current amplification [18].

2.5. Power supply

The power supply consists of a rectifier, which is a power converter for providing ac to dc conversion directly. The non-linear characteristic of a diode was used to convert alternating current into unidirectional but pulsating current. The pulsating current is removed in a frequency selective circuit called filter before it was regulated to the required voltage 6V, which is used to drive the relay [15].

The circuit is powered from external power supply, which is not connected to PC. This arrangement prevents any currents on the external circuits from damaging the parallel port. The power supply unit provides 8 outputs to power the 8-relay control by the 8 data port of the PC parallel port, which provides a controlling voltage to the relay driver is as shown (Figure 3).

3. Circuit Description and Mode of Operation

The optoisolator's input is a light emitting diode. The resistor R1 is used to limit the current when the output from the port is on. The 470 Ohms resistor limits the current to around 10.64 mA, which is sufficient to drive the output transistor.

The output side of the optoisolator U8 (pin 4 and pin 5) acts like a transistor, with the collector (pin 5) at the top of the circuit and the emitter (pin 4) at the bottom. When the output is turned on (by the input light from the internal LED in the optocoupler), current flows through the resistor and into the transistor, turning it on which now allow the flow of current into the relay, RLY8.

Turning the input (Dn) from the parallel port off (this is possible through the control program already loaded or installed into the system) causes the output of the optoisolator to turn off, so no current flows through it into the transistor, and the transistor turns off. When the transistor is off, no current flows into the relay RLY8 and it switches off the load. The diode D8 provides an outlet for the energy stored in the coil, preventing the relay from back feeding the circuit in an undesired manner.

The circuit can be used to control output loads to maximum of around 60mA when the external power supply is 6V.

Seven other similar circuits are wired and similarly connected in parallel to make a complete 8 software controllable external interface.

4. Design Analysis of Discrete Components

In this section we will look at how the values of the discrete components used were calculated and selected.

4.1. Analogue switching

Based on the rating of the relay, the transistor C9014 is selected

$$V_{RLY} = I_{RLY} R_{RLY}$$

$$I_{RLY} = \frac{V_{RLY}}{R_{RLY}} = \frac{6}{100}$$

$$I_{RLY} = 60mA$$

$$I_{RLY(MIN)} = 40mA$$
$$I_{RLY} = I_{CB(MAX)}$$

For design purpose, $I_{B8} = 10\% I_{C2(MIN)}$ design criterion. I_{B8} = 4mA. The base resistor is obtained as follows

$$R_{B8} = \frac{V_{CC} - V_{BE}}{I_{B8}} \qquad (1)$$

$$R_{B8} = \frac{6.6 - 0.6}{4} = \frac{6}{4} = 1.5\text{k}\Omega$$

Based on the design criterion a suitable transistor was obtained, while a base resistor of 1.5kΩ was used.

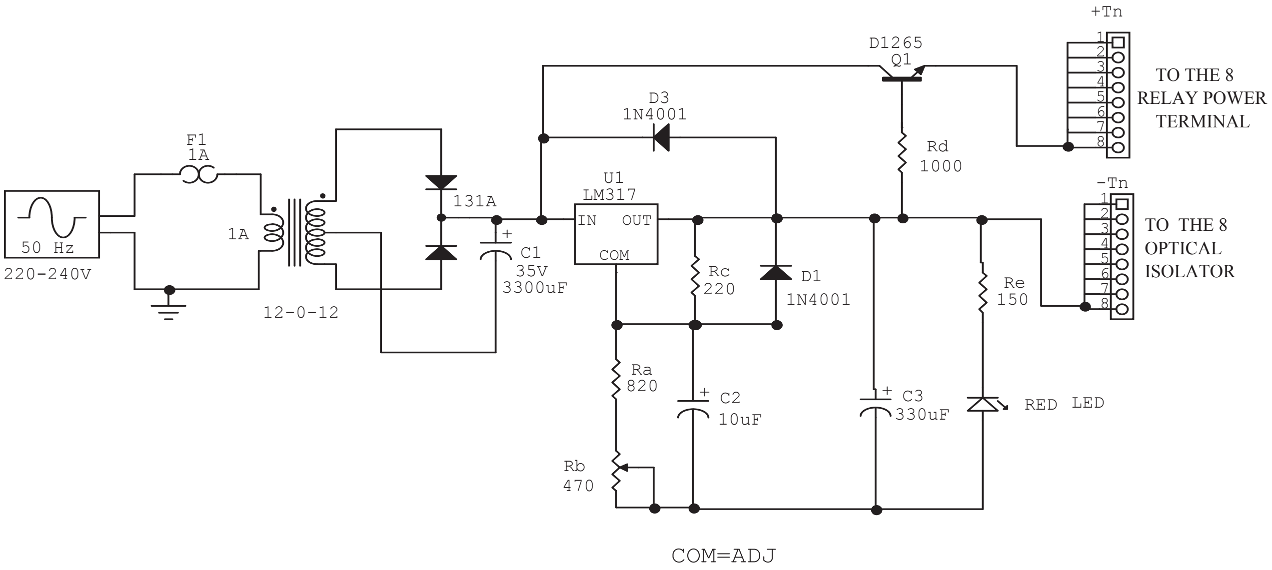

Figure 3: 6 Volt DC power supply unit.

4.2. Digital switching

In digital switching, the output of the parallel port is normally in TTL levels. An electrical "high" (on a pin or line) is TTL high (+2.4 to +5 volt). An electrical "low" is TTL low (0 to +0.8 volts). A data high is logic 1 while a data low is logic 0. Most port can source around 12mA and sink up to 20mA. Drawing 10mA from Port;

$$R_1 = \frac{V_{PORT}}{I_{DESIGN}} \qquad R_{MAX} = \frac{5V}{10mA} = 500\Omega \tag{2}$$

R_1 in series with diode resistance R_d then

$$R_1 = \frac{V_{PORT} - V_D}{I_{DESIGN}} \tag{3}$$

$$R_1 = \frac{5-0.6}{10mA} = 440\Omega$$
$$R_1 \cong 470\Omega$$

Power of the control signal (parallel port signal)

$$P = IV \tag{4}$$

I_{PORT} = 10mA, V_{PORT} = 5V (TTL)

$$P = 10mA \times 5V = 50mW$$

4.3. Optocoupler (pc123)

Maximum forward current, I_F = 50mA; I_{DESIGN} = 20% maximum I_F. Current transfer ratio (CRT), $I_B = 20\%\ I_{F,MAX} = 20\%\ I_{DESIGN} = (20\%)^2 I_{F,MAX}$ = 40% × 50mA. I_B = 2mA. But $I_B = 10\% I_C$, therefore $I_C = 10 \times 2$mA = 20mA. $I_E = I_C + I_B = 20 + 2 =$ 22mA.

5. Design of Control Program

In this section, the design approach of the control program will be looked into.

5.1. Design method

Prior to developing the program for any data acquisition or control exercise, it is essential to have a complete understanding and identification of the problem such that the software can be constructed based on a defined schematic framework. This involves a formal approach to the structuring of the programs.

VB was used to implement the control program, although it does not have possibilities for input/output access. Nevertheless, Dynamic Link Library (DLL) allows VB to link (before compilation) the code (developed in the form of library using another language like VC++) during run time [14]. VC++ has port input/output read/write functions which facilitates the user in accessing the system hardware. The flowchart for the control program is as shown (Figure 4).

5.2. Device controller interface

Device Controller is a self-installing program. The interface of the control program, named Device Controller is as shown (Figure 5). It is controlling software which controls the 5V from the output of the PC parallel port.

The function of the various parts of the device controller is as follows:

Key mode: Enables selection of mode.

Send mode: Writes or send data directly to the port as entry is made in the display panel.

Ignore Mode: Waits for the user to issue send command before data could be sent to the data port.

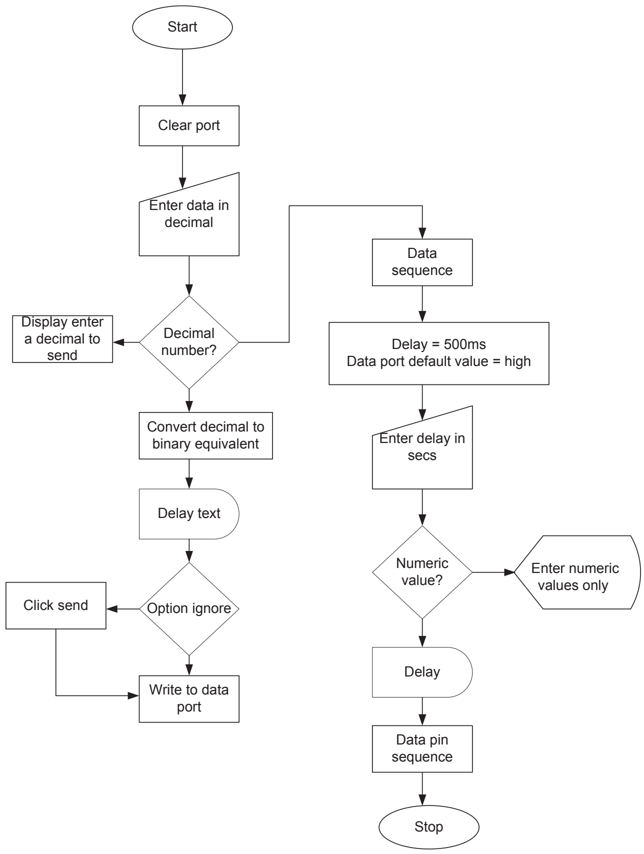

Figure 4: Flow chart for the control program.

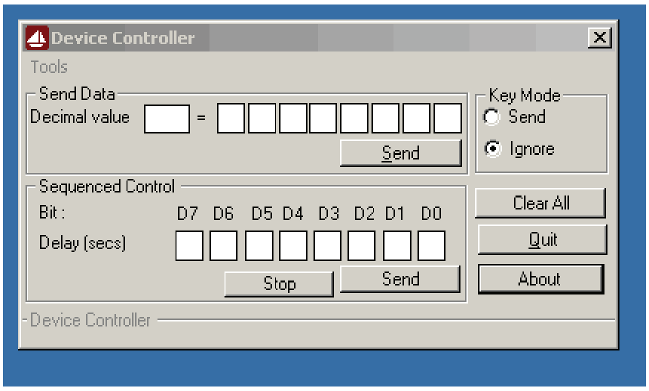

Figure 5: Device controller interface.

Clear All: Initialized the port by writing 0s to all the data pins.

Quit: Terminates, close or end the program.

About: Gives information about the device controller.

Sequenced Control: Enables the introduction of delay or timer. Time are entered in seconds to enable sequence control of data sent to the port.

Send: Write to the data pin of the parallel port.

Stop: Marks the end of the sequenced control.

5.2.1. Mode of operation

The mode of operation or concept of the *Device Controller* is as follows:

When binary 1 is sent to the data pin (D0–D1), the optical isolator invariably activate the relay which now power on the load and sending binary 0 to that same pin deactivate the relay. However, it should be noted that in the design of the interface, two modes are possible, the digital mode and the manual mode. For the digital mode, the position of the switches has to be in 1 position to enable control by the computer system and every bit of the binary number control one output bit. For example to set pins 1 and 2 i.e. D0 and D1 to logic 1, enter decimal number 3 from the keyboard in the display panel and its binary equivalent would be displayed in the panel labeled D0–D7 (the data port), then click on send.

If however the decimal number should exceed 255 there is an over flow of signals to the control port C1-C4 which gives it the additional capability of controlling 12 loads. With the timer set for each pin, device controller sequentially turns on or off loads connected to the port. Each load is held on for about T secs depending on the value of time T entered for each pins.

6. Testing and Result

Prior to assembling the complete system, the various units in the system were tested individually and they functioned satisfactorily. The design was implemented at the Department of Electrical/Electronic Engineering, University of Benin. The printer cable was connected to the computer and with the program loaded the external devices were interfaced with the computer system. Once the key on the keyboard is pressed, the corresponding devices are powered on depending on the value outputted to the data port load. The devices used included table fan, radio/tape recorder, pressing iron, 8-electric bulbs, computers and rechargeable lanterns. The result obtained was satisfactory.

7. Conclusion

The PC parallel port adapter which is specifically designed to attach printers with a parallel port interface, has been found to be useful as a general input/output port for any device or application that matches its input/output capabilities. Results of the analysis show that, it is easier and economical to make use of an existing microprocessor in the computer than to buy a new one and face the usual problem of loader or programming, which is expensive and difficult to find.

Although user friendly Visual Basic does not have any functions or support access to parallel port of a computer directly. Nevertheless, Visual Basic was used to implement the control program. This was made possible by adding a Dynamic Link Library in VC++ and calling its exported functions from VB. In this paper, we designed and constructed a digital control system and we have been able to control up to twelve devices by the inclusion of signal overflow to additional four ports.

In conclusion, a PC parallel port is an inexpensive but yet a powerful platform for implementing projects dealing with the control of real world peripherals and this can serve as a good software and hardware exercise when implemented as a laboratory experiment. Also, I believe this technique can be be explored for water level control [25] and automatic safety control in immersion water heater [26]

References

1. Fraser, C. J. and Milne, J.S. *Microcomputer Application in Measurement Systems.* Macmillan, 1990.
2. Anderson, P. H. Use of a PC Printer Port for Control and Data Acquisition, Technology Interface. *The Electronic Journal for Engineering Technology*, Fall, 1996.
3. Siemens Aktiengeselischaft. *Automatic Generation Control and Economic Load Dispatch.* Germany, 1980.
4. Groover, M. P. *Automation, Production System and Computer–Integrated Manufacturing.* Prentice Hall, Third Edition, 2007.
5. Bartee, T.C. *Computer Architecture and Logic Design.* McGraw-Hill International Edition, 1991.
6. Gurvinder, S.V. *Digital Computer Control System.* Macmillan New Electronics, 1991.
7. Ditch, W. *Microelectronic Systems: A Practical Approach.* Butterworth-Heinemann, 1995.
8. Groover, M.P. and Zimmers, E.W. Jr. *CAD/CAM: Computer Aided Design and Manufacturing.* Prentice Hall, First Edition, 1983.
9. Waterworth, G. *Workout Electronics*, Macmillan Education Ltd, London, 1988.
10. Palya, W. L. and Walter, D. E. A powerful, inexpensive experiment controller or IBM PC interface and experiment control language. *Behavior Research Methods, Instruments & Computers*, 25 (2), pp. 127 136, 1993.
11. Penttonen, M., Salmi, M., Hamalainen, P. and Merilnoto, J. A microcomputer system for controlling classical conditioning experiments. *Behavior Research Methods, Instruments & Computers*, 26 (4), pp. 447–453, 1994.
12. Junior, J. C. F., de Lucena, S. E., Pimentel, U. and Rocha, W. N. *A multimedia environment for teaching and learning the interfacing of external devices with a microcomputer.* International Conference on Engineering Education, Oslo, Norway, 2001.
13. Thimmaiah, P., Rao, K. R. and Gopal, E. R. External device control using IBM PCs centronics printer port. *Journal of the Instrument Society of India*, 32 (4), pp 239–247, 2002.
14. Rao, K. R., Laxmaiah, M. V., Gopal, E. R. and Prasad, G. R. A powerful and inexpensive technique to interface IBM PCs printer port using visual basic. *Journal of the Instrument Society of India*, 35 (3), pp 320 – 327, 2005.
15. Gary, R. *Solid State Fundamentals for Electricians.* American Technical Publishers, Inc Homewood Illinois 60430, 1985.
16. Sedra, A. S. and Smith, K. C. *Microelectronic Circuits.* Oxford University Press, USA, Sixth Edition, 2009.
17. Smith, R.J. *Circuits Devices and Systems.* John Wiley and Sons, Third Edition, 1998.
18. Covington, M.A. Port Access in Visual Basic. *Poptronics*, July 2002, pp. 26.
19. PARALLEL PORT. `www//ourworld.compuserve.com`.
20. BEYOND LOGIC. *Interfacing the Standard Parallel Port.* `http://www.beyondlogic.org/spp/parallel.htm`.
21. ENGDAHL T. *Parallel Port Output.* `IBM_Google_Parallel_Output.htm`, 1996-2000.
22. COFFEE-HOW TO. *Electronic Circuit.* `http://en.tldp.org/howto/coffee-2.html`. 2003.
23. ELETRONIC ZONE. *Control Electrical Appliances Using PC.* `http://www.eletronic-circuit-diagrams.com/`. 2002.
24. Hill, W. and Horowitz, P. *The Art of Electronics.* Cambridge University Press, Second Edition, 1995.
25. Anyanwu, C.N., Mbajiorgu, C.C. and Anoliefo E.C. Design and implementation of water level controller. *Nigerian Journal of Technology,* Vol. 31, No. 1, 2012, pp. 89–92.
26. Enokela, J.A., Eze, A. and Yakubu, J. An automatic safety control for immersion water heater. *Nigerian Journal of Technology*, Vol. 26, No. 2, 2007, pp. 74–80.