



## Design and Construction of a Computer Controlled Clothes Washing Machine

**\*AKPOBI, J A; OKONTA, J A**

*Production Engineering Department, University of Benin, Benin City, Nigeria*

**ABSTRACT:** In this work we develop a computer software for controlling the operations of a domestic clothes washing machine. The valves, pump, electric motor and other sensitive parts were computer-controlled. In the design, the mechanical timer's function were replaced by a software driven emulator which then controls components of the machine to carry out specific tasks at pre-determined intervals. The software was written using Microsoft Visual C++ for the IO.DLL File, while the control codes were developed using Microsoft Visual Basic Language. The software performed accurately and efficiently. @JASEM

For a very long time, efforts have been made to overcome the difficulties associated with washing clothes manually. It is painstaking, wearisome and time consuming to constantly wash bulky clothes, especially the very dirty ones. The search for easier and highly efficient ways of washing clothes has evolved over the years. One of such methods was the use of scrub boards in 1797. In the 1800s, hand-powered clothes dryers came into focus (Ament, 2005). Also a hand-powered, washing machine with a drum was invented by James King in 1851. While in 1908, Alva J. Fisher invented the first ever electric-powered washing machine, (Procter and Gamble, 2005). The developmental trend continued and a time (time switch) controlled washing machine evolved, which was powered by electricity. One of the authors and coworker had earlier designed and manufactured a horizontal axis automatic washing machine, ( Akpobi and Ayeke, 2003).

Present day washing machines are usually incorporated with Timers. The Timer is a device which is subject to the effects of wear and tear, and high cost of maintenance. It is usually a combination of mechanical and electrical components. In this work we develop a computer controlled Washing Machine that is highly efficient, reliable and durable. This would help to reduce the running cost. With this system, other computer-controlled Washing Machines can be networked with a single software program and a single computer. This will be suitable for industrial application. The controller was designed and constructed to control: 1) the pumping of water in and out of the machine, Aand 2) the washing and the rinsing of the clothes. This was achieved through the use of a personal computer. The developed controller, is very flexible and user friendly. It is supported by robust, yet simple software, which controls precisely. A blend of electronics and computer technology makes the controller very elegant. It can operate completely on its own for different programmed timings. This can be easily changed by, clicking on the appropriate command button. Thus, the proper washing of different fabrics relative to the degree of dirtiness, can be carried out.

### *Schematic Structure of the Controller*

The controller consists of different sections. These sections include the regulated power supply section; the relay section; the parallel port section and the software section. The relays are powered by the regulated direct power supply, which converts the alternating current supply to direct current supply. This conversion is possible through the appropriate use of diodes, step-down transformers,

capacitors, voltage regulators, etc. the relays are diode-protected. Some of the relays are connected to the terminals of the electric motor: some are connected to the terminals of the electric pump while some of the relays are connected to the terminals of the valves, which allow intake of water. The parallel port section relies on the signals to and from the developed software. The software relies on the software timer to achieve its objectives. The relays are also connected to the pins of the parallel port. To safeguard the motherboard of the personal computer from the danger of excessive voltage intake, an appropriate buffer is employed between the relays and the pins of the parallel port. The buffer acts as an opto-isolator, while the diode prevents excessive voltage spikes from the relay coil from damaging the computer and the relay, (Engdahl, 2005). The schematic diagram of the controller is shown in Fig.1, while the flow chart of the process is shown in Fig. 2.

The relay uses direct current (d.c.) of 12V. This comes from appropriately regulating the alternating current (a.c.) supply. The relay can switch alternating current supply on and off through signals sent from the parallel port. In the construction the relay was interfaced to the parallel port through the wiring in the printer cord. The developed software was then used to configure the relay's responses by sending signals, which are effected as appropriate in the Input/Output (I/O) mode.

To keep the parallel port from being damaged by excessive voltage values, a PC123 optocoupler was employed. This optocoupler optically isolates excessive input voltages from the circuit. When signals, called from the main program, get to the optocoupler, the circuit is completed. The circuit will be open when there are no receptions of signals. This prevents the dangerous back Electromotive force from being developed in the circuit due to loss of signals.

*The Controller's Software Design:* The design of the controller involved the following: 1) Defining the function of the controller 2) Developing the software 3) Testing the software on the built Controller.

In carrying out the design, the software's methods and functions were written as separate modules. Each written module was debugged and tested in assembly language. This was achieved by incorporating the software with an Input/Output dynamic link library (IO.DLL). This was written in C++ programming language. During testing it was observed that the modular coupling was quite minimal while the overall efficiency of the software was high

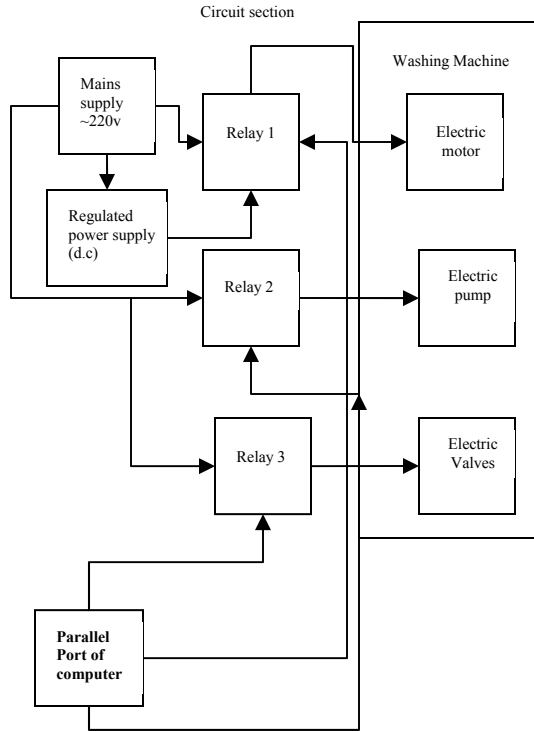


Fig. 1: Block diagram of the controller.

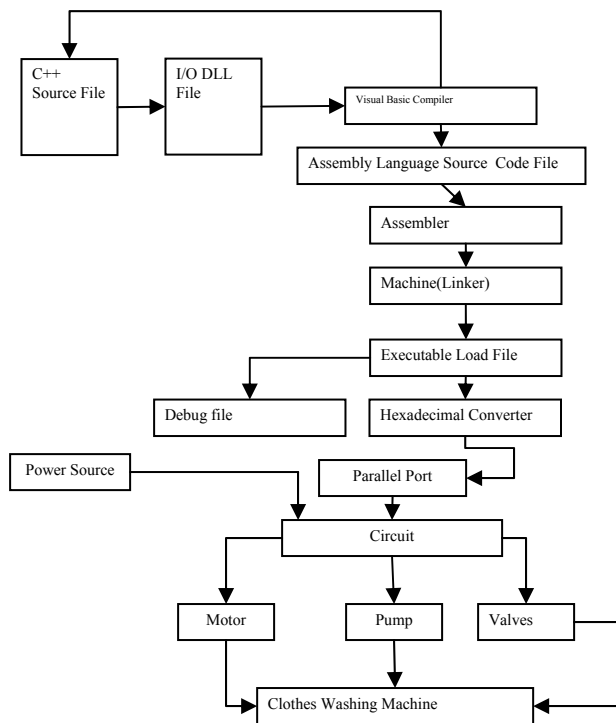


Fig. 2. Flow chart of the operational sequence of the operational software

The Software Development: The very important part of the software was the input-output dynamic link library file (IO.DLL file). The IO.DLL file was written in C++ Language, since Microsoft Visual does not communicate directly with the parallel port. However, Since Microsoft Visual Basic can communicate with C++ Language, and C++ can communicate with the parallel port, the IO.DLL files were written in C++. The main program was then developed using Microsoft Visual Basic. In order to protect

the copyright of the software we have omitted its source codes in this work. We then used a printer cable to link the Washing Machine with the parallel port of the computer. The Parallel Printer Port used for this work, had a total of 12 digital outputs and 5 digital inputs accessed via 3 consecutive 8-bit ports in the processor's Input/Output space. It was only necessary to connect to the 8 data lines and the ground pins. The schematic of the parallel port used is shown in Fig. 4, while the data for the pin assignments of the port are shown in the Table 1.

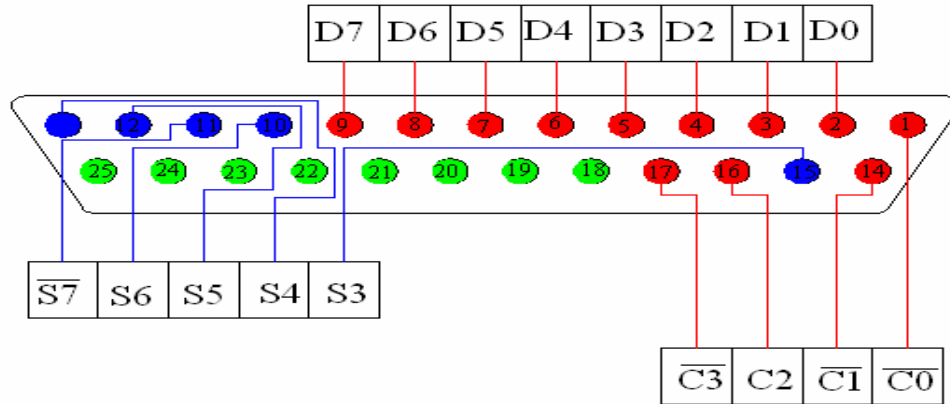


Fig.. 4: Schematic of the 25-way Female (D-Type) parallel port Connector

Table 1: The pin assignments of the 25-way Female (D-Type) parallel port Connector

Pin Number	Function
C0 or 1	-Strobe
D0 or 2	Data 0
D1 or 3	Data 1
D2 or 4	Data 2
D3 or 5	Data 3
D4 or 6	Data 4
D5 or 7	Data 5
D6 or 8	Data 6
D7 or 9	Data 7
S6 or 10	-Ack
S7 or 11	+Busy
S5 or 12	Paper End
S4 or 13	+SelectIn
C1 or 14	-AutoFd
S3 or 15	-Error
C2 or 16	-Init
C3 or 17	-Select
18-25	Ground

The IO.DLL file links up the assembly language (computer or machine language) when the program starts running. Then, the computer (machine) passes the signal to the executable load file. The executable load file carries out the action instructed by the computer. Further more, the computer identifies the specific parallel ports through the use of hexadecimal numbers. For this to be achieved, what

was written in decimal will be converted to hexadecimal so that the computer can identify the particular parallel port to which the signals are meant for. The parallel port, then sends messages in form of data signals to the circuit. The circuit then responds by linking the particular component to which the signals are meant for. The components involved are the electric motor, the electric pump and the electrically operated valves.

### DISCUSSION

The software program was developed and tested. It was observed that it accurately achieved its design objectives as the clothes washing machine responded accurately and fast to the processes initiated by the software. On the part of the software, the program was quite simple, user-friendly and particularly very robust. Also, the delay mode was used for the timing, as using the programme timer would result in obstructions when the computer takes long to access other applications. With this, the reliability of the software was highly increased. The software was also enhanced by the incorporation of a graphic interface shown in Figure 3. This is actually a step further into complete automation of the clothes washing machine.

*Conclusion:* The machine was tested and found to be very reliable and efficient. Therefore, the object of the design was satisfactorily achieved and the logic was found to be very sound.

## **REFERENCES**

Ament, P (2005). History of Washing Machine.  
<http://www.ideafinder.com>.

Engdahl, T (2005). Parallel Port Interfacing Made Easy.  
[http://www.epanorama.net/circuits/parallel\\_output.html](http://www.epanorama.net/circuits/parallel_output.html).

Akpobi, JA; Ayeke, E O (2003). Design and Manufacture  
of A Horizontal Axis Automatic Washing Machine

with Emphasis on the Critical Parts. Journal of  
Engineering for Development, Vol. 5. pp. 9-16.

Dyson, J.(2002). Contra- Rotator Washing Machine.  
<http://www.dyson.com>

National Museum of Science and Technology (2002).

[http://www.sciencech.nmstc.ca/english/collection/washing\\_machines.cfm](http://www.sciencech.nmstc.ca/english/collection/washing_machines.cfm)