Development of Stepper motor based Two DOF Robotic Arm Transferring Liquid using Peristaltic Pump

Sudha Ramasamy[1], Karthikesh.R[1], Manikandan.P[1], Padma Thiagarajan[2]
[1]School of Electrical Engineering, VIT University, Vellore, India.
[2]School of Biosciences and Technology, VIT University, Vellore, India
E-Mail: ishuma@gmail.com

***Abstract* - The aim of this work is to transfer liquid contents from one micro cell to another using two stepper motors and a peristaltic pump. There are two objectives here. One is to develop a low cost robotic arm using stepper motors. The second objective is the control and calibration of the peristaltic pump. All parts are controlled and operated by their respective microcontrollers. Fulfillment of both the objectives leads to an integrated system to transfer liquids from one cell to another. The end effecter of the robotic arm is connected to the peristaltic pump. The pump has two pipes connected to it. Through one pipe it takes in the liquid and through the other pipe it delivers the liquid into the second cell. After transferring one sample of liquid, the arm moves to a cleaning module where the end effecter is cleaned to avoid cross contamination. The robotic arm is built using stepper motors and controlled using Atmega32 microcontroller whereas the peristaltic pump is controlled and calibrated using 8051 microcontroller. The pumping is done with the help of DC motors. As a result, the working of the robotic arm and the peristaltic pump is verified experimentally.**

Keywords- Liquid Transfer, Microcontroller, Peristaltic Pump, Robotic arm, Stepper Motors

I. INTRODUCTION

The liquid transfer system is used to transfer a liquid from one cell to another automatically. Simultaneously the end effecter gets cleaned and transfers the next sample of liquid in a sequence and the whole process repeats. The main core components of this liquid transfer system are two stepper motors and one peristaltic pump. The main purpose of using stepper motors for the robotic arm is to get precise and accurate angular movement of the arm. Here, transfer of liquid from a cell takes place and in order to place the end effecter into the cell, step wise angular movement is required. Another advantage of using stepper motors is that they do not require any feedback signal.

Peristaltic pumps are the most common type of propelling units used in fluid injection systems (FIS). A propelling system is one which affects the transport of a liquid through a dynamic mode either by impulsion or aspiration. This function can be performed by a large variety of devices, the functioning of which is the key for achieving proper analytical performance of the overall approach. Another type of system known before uses a dispenser with a motor controlled metering valve to dispense liquids, but it has few performance limitations like slow response time [1].

D.C. servo motors can also be used to control the robotic arms but they have limitations as well. They require closed loop feedback control which absorbs large amount of the operational time. Additionally, the analog nature of the D.C. servo motor systems complicates their interfacing with computers of the preferred digital type [2].An alternate system for transfer mechanism was invented where the speed of the motor can be controlled by the microcontroller based upon a calculation stored in the microcontroller. But, it is not possible to get the accurate speed of the motor each time as there may be some defects with the motor and also the power supply at times [3].The tubing material and the tube diameter will vary according to the type of liquid that needs to be pumped [4]. There is an increasing interest for embedded systems capable of handling small and precise volumes of liquids, such as in applications like drug delivery systems or micro total analysis system (TAS) implementations. Adding sensing and controlling aspects in propelling devices through electronics and microcontrollers are becoming very popular as it has many more applications [5]. Due to their importance and applications, lot of research is focusing around reducing the hardware complexities in integrating propelling devices. The integration and multiplexing of complex peristaltic pump propelling networks on a single microchip based on VLSI concept of pull ups and pull downs seems to have lot of scope in future [6]. Integrating peristaltic pump to a microcontroller and thus making a sensor and actuation part to automatically control the liquid flow rate or for monitoring and adjusting the flow rate can be easily done [7]. In this study, the robotic arm is controlled using Atmega32 microcontroller. This microcontroller is used because of its ease of programming and ability to interface directly with the stepper motors. A calibrating and controlling system using 8051microcontroller for peristaltic pump is also designed and by integrating the pump to the robotic arm, the implementation of a automated liquid separating and filling system was made possible.

II. METHODOLOGY

A. Basic Approach

The first step in this design was to divide the whole system into two parts which were design of robotic arm using a stepper motor and the control and calibration of the peristaltic pump. Finally, the inlet of the peristaltic pump was connected to the end effecter of the robotic arm and it formed a complete integrated system. It was decided to use Atmega32 microcontroller for the control of robotic arm and 8051 microcontroller for the peristaltic pump. This was because the AVR development board supported the operation of stepper motors very well without the use of any external drivers. To operate the peristaltic pump by driving DC Motors, the assembly language of 8051 was simple. Further it was also easy to interface.

B. Components Used

Two stepper motors were used for the robotic arm. One was employed for the body movement and the other was used for the shoulder movement. The robotic arm was built using thin aluminium sheets and a pulley was used to mount the shaft of the stepper motors. The stepper motors used were 6 wire unipolar ones. The end effectors used while verifying the experiment was a straw. The peristaltic pump used consisted of a motor-driven wheel with peripherally placed rollers and an adjustable compression cam which is squeezed against the rollers. A 12V, 200 RPM motor D.C. motor was used to pump the liquid by fixing its shaft to the peristaltic pump. It was not possible to drive the D.C. motor directly through the 8051 microcontroller development board; hence a driver IC L293D was used. Apart from these, bread board was used for connection purposes. Connecting wires and two way connectors were used to interface the stepper motor and the development board. Figure 1 shows the block diagram for the control and calibration of peristaltic pump. Figure 2 shows the block diagram for the control of robotic arm using the Atmega32 microcontroller. Port D is connected to one stepper motor and Port C is connected to another one. 12V DC power supply was given to both the development board and the two stepper motors through a 12V DC adaptor. The AVR studio codes were loaded into the microcontroller using an USB AVR programmer with the help of software called AVR burner.

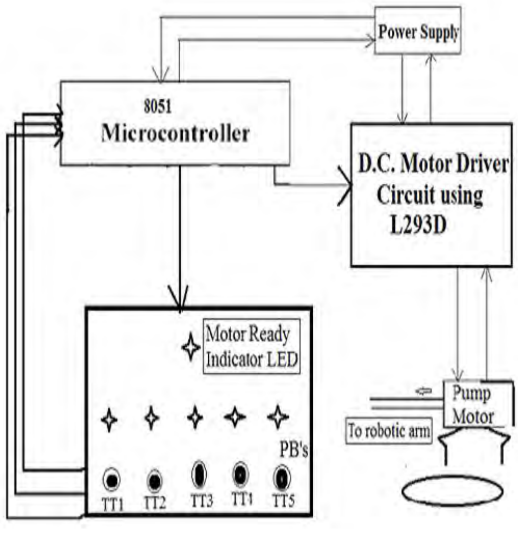

Figure 1: Control and calibration of peristaltic pump

C. Stepper Motor Driver Circuit

The stepper motors used in this were 6 pin unipolar ones. Two pins were connected to the 12 V DC power supply and the other four pins were connected to the microcontroller. Initially there was a difficulty in connecting the power supply wires. The other four pins belonged to the two coils of the stepper motor. That is each coil contains two wires. Two wires of the same coil should not be given the same pulse. That is if wire 1a of coil is given high pulse, 1b wire should be given low pulse and vice versa. The order in which the pulses are given to the wires determines the direction of rotation of the motor shaft. L293D was used to drive the DC motor from the microcontroller. This is because the DC motor requires more current to start the rotation and if this is directly interfaced with the development board, the latter might get burnt. The pin diagram of L293D driver is shown in Figure 3. The Vcc pins are pin numbers 8 and 16 with one driver circuit; this can be operated for two output sources. Also, two input sources can be connected to this driver circuit. Programming for the stepper motors was done in AVR Studio-4 software and for the peristaltic pump it was done in Keil µvision-3 software. Coding in AVR was done in C format and it was burnt to the microcontroller using the AVR programmer. Two programmers were used to burn the assembly languages into the microcontrollers.

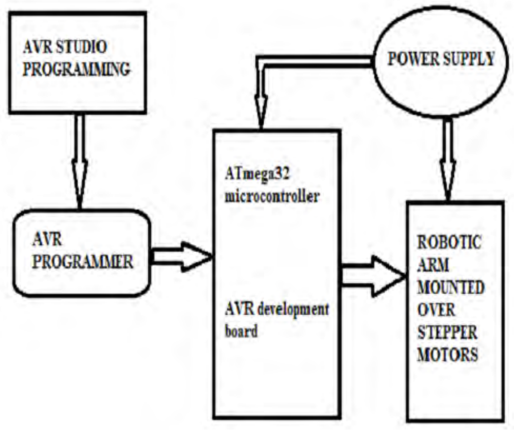

Figure 2: Control of Robotic Arm

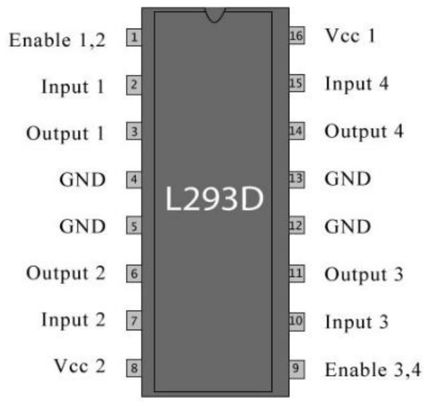

Figure 3: L293D pin diagram

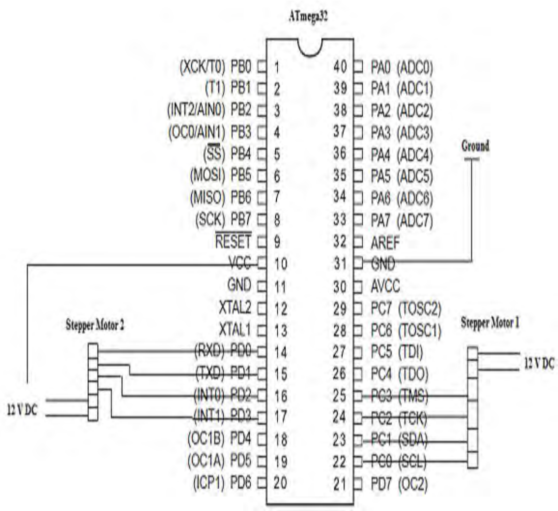

Figure 4: Circuit diagram for stepper motor interfacing

For 8051 microcontroller, flash magic burner was used and for Atmega32 microcontroller, AVR programmer was used. The flash magic programmer was connected with the computer and the programme was loaded into it using the flash magic software. Then the program is stored into the 8051 microcontroller using the flash burner. For ATmega32 microcontroller, AVR programmer was used and it was connected to the computer. The program was loaded into the programmer using the AVR burner software and from the programmer it was loaded into the microcontroller. The whole process starts with the planning of design for this system. In the next stage, the hardware parts were built *i.e.*, the robotic arm was built using aluminium sheets and the pipe was fixed inside the peristaltic pump. Later, the two parts were connected together and programming was done and loaded into the microcontrollers. Finally, power supply was given and the system was operated. Fig 5 shows the circuit of interfacing motors with microcontroller and LEDs.

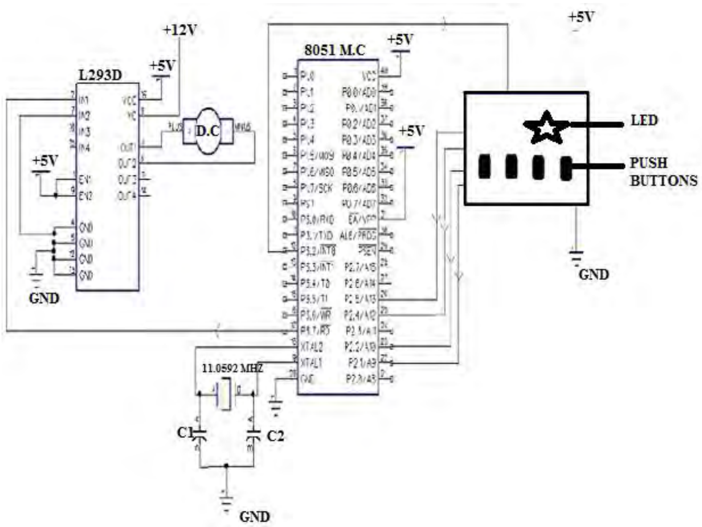

Figure 5: Circuit diagram for D.C. motor interfacing

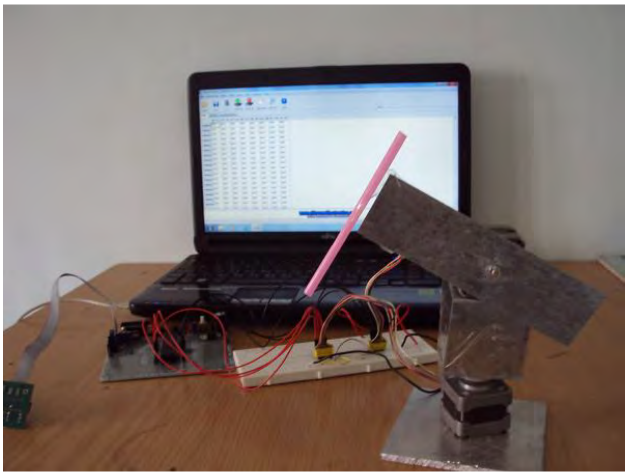

Figure 6: Experimental setup for 2 degrees of freedom robotic arm

III. EXPERIMENT

Thin aluminium sheets were cut and were bent to form the body of the robotic arm. This piece was mounted over the shaft of the stepper motor with the help of a pulley. Next, another stepper motor was sealed into this body and its shaft was fixed into another small aluminium plate which is the shoulder of the robotic arm. To this shoulder plate, a thin straw was attached to verify the working of the robotic arm experimentally. The robotic arm system was fixed on a thick aluminium plate. The developed robotic arm with the experimental setup is shown in Figure 6. To fix the robotic arm into the thick aluminium platform firmly, a strong cellophane tape was used. The same material was used to fix the second stepper motor into the body of the arm. The peristaltic pump was affixed with one pump tube and the pump was mounted over the shaft of the D.C. motor. One end of the pipe is connected to robotic arm and the other end to the liquid delivery system. Pump tube affixed with the peristaltic pump is shown in Figure 7. The two stepper motors of the robotic arm were interfaced with two ports of the Atmega32 microcontroller while the D.C. motor was interfaced with the 8051 microcontroller with the help of a driver L293d. To interface the whole system with the software programming, two external burners were required. For Atmega32 microcontroller AVR programmer was required and for the 8051 microcontroller a flash burner was needed. The operating time of the pump is given by the user through the push button switches on the right hand side of the Figure 5. To operate the motor for 5 seconds, the first push button was pressed. To operate it for 10 seconds the second push button was pressed. For 15 seconds and 20 seconds duration, the third and fourth push buttons should be respectively pressed. For our experimental analysis, the first push button was used. That is, during the operation of the embedded system, the peristaltic pump sucked liquid for 5 seconds. The AVR burner was used to write the assembly language coding to the microcontroller. An USB AVR programmer was connected to one of the USB port of the computer system, and it was connected to the microcontroller using a two way connector. Using the extreme burner AVR software, the programming codes were loaded into the programmer and later it was loaded into the microcontroller. If a change in the programming is required, it has to written in to the programmer each time. After the program is loaded into the microcontroller, the microcontroller works according to the loaded program provided that a 12V DC power supply is given.

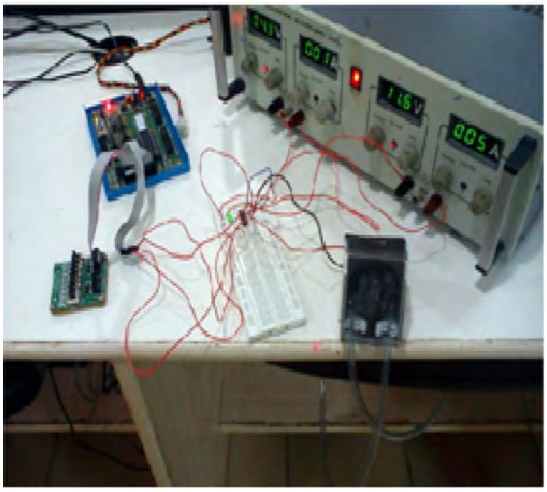

Figure 7: Experimental setup for 2 degrees of freedom robotic arm

The programming algorithm for both the modules (robotic arm and peristaltic pump) was described in the following steps:

Step 1: Opened AVR Studio-4 and create AVR GCC project file.

Step 2: Selected simulator and choose Atmega32.

Step 3: Declared ports C and D as the output ports and typed the coding for both horizontal and vertical movement of the arm.

Step 4: Compiled the program and save it.

Step 5: Opened AVR burner and read the saved file.

Step 6: Burned the program to the microcontroller using AVR programmer.

Step 7: Opened Keil µvision-3.

Step 8: Declared port 2 as the input port and port 3 as the output port.

Step 9: Wrote the coding for the operation and peristaltic pump and saved it.

Step 10: Burnt the program to the microcontroller using flash burner.

Finally when the power was switched on, the whole system was controlled by both the microcontrollers. The power supplied should be 12 V DC.

IV. RESULT AND DISCUSSION

The whole system takes about 17 seconds to transfer one sample of liquid and wash the end effecter. The robotic arm alone takes 12 seconds to make a movement of 90 degrees whereas the peristaltic pump sucks the liquid sample for 5 seconds. After the end effecter is cleaned, the arm moves back to its initial position within 6 seconds. The time taken by the whole system to perform its tasks is shown in a graph in Figure 8. This graph shows the time taken by the system when the project was verified experimentally.

Running Time of the Stepper Motor without the pump = 12 seconds

Operating Time of the Peristaltic Pump = 5 seconds

Total operating time of the whole system for one iteration = 17 seconds

The X axis in the graph is time in seconds and the Y axis depicts the movement of the arm in degrees. The graph was plotted between degrees moved with time in seconds.

The peak point of the graph depicts the position of the robotic arm at the cleaning module. The slope indicates the movement of the integrated system. The straight dashed line shows the time taken by the shoulder to bend down and the pump to suck the liquid. That is, it takes totally around 6.5 seconds for this initial task. The use of stepper motors has given better accuracy and precision control without the necessity for a feedback signal. In the case of other motors like servo motors, etc. this experiment requires a feedback signal to maintain the position which makes the whole system more complex. There was no need to use any feedback signal here because the stepper motors made step wise angular rotation and it was very simple to control their movement and maintain the position of the arm precisely. This was verified experimentally by sucking out liquid from a single test tube. However multiple liquid transfers are also possible by placing multiple test tubes and controlling the system by sending an input signal to the microcontroller that controls the robotic arm. The user can manually instruct the system to transfer liquid from a particular test tube by using a push button. This signal from the user instructs the robotic arm to move to that particular cell or test tube and transfer the liquid. This would fasten the clinical test processes and save time and human efforts.

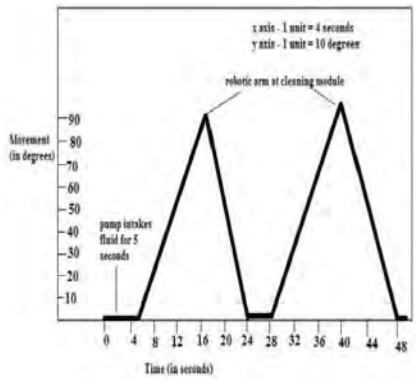

Figure 8: Response of Robotic arm movement with time

V. APPLICATIONS

This embedded system has major scope in clinical and laboratory tests where transfer of liquid samples can be done automatically. In order to carry tests of multiple liquid samples, this automatic liquid transfer system will be of major importance. This system will play a major role in pharmaceutical industries where there will be a need to transfer liquid contents in µl quantities. The pump has to be programmed to transfer µl quantities and this will be very effective system. In beverage industries many liquid ingredients are required to be mixed together. For this purpose, the contents have to be transferred between cells. This liquid transfer system can fit into that place very suitably to transfer the liquid contents from their cell to the production cell. The advantage of this system with the others is that this system is small and compact and also installation cost is very low compared to the other automated systems.

ACKNOWLEDGEMENT

The authors thank VIT University for the infrastructure provided for undertaking this research.

REFERENCES

[1] Richard. P, "*Apparatus and Method used for dispensing Fluid*", pp. 2-6, International Application Published under the Patent Cooperation Treaty, Nordson Cooperation,28601 West Lake, Clemens Road, Westlake, U.S., 1987, October 8.

[2] Saathoff, David. A. Dempski, James. L, Mosak, Joel, "*Article Transfer Mechanism*", pp. 2-12. European Patent Application, 1991, May 2.

[3] Goumas, Peter. G, Wolfe, Paul. T, "*Apparatus and Methods for handling articles in Assembly Automated Process*", pp. 2-5, European Patent Application, 1987, September 16.

[4] R.C. Prados-Rosales, J.L. Luque-Garcia, M.D. Luque de Castro, "Propelling devices: the heart of flow injection approaches", *Analytical Chemistry Division,* Annex C-3, Campus of Rabanales, University of Córdoba, E-14071, Córdoba, Spain,2002.

[5] Matthew L. Cantwell, Farid Amirouche, Johann Citerin, "*Low-cost high performance disposable micropump for fluidic delivery applications*", Biomechanics Lab, UIC, Chicago, USA, 2011.

[6] Matthew C. Cole, Amit V. Desai, Paul J.A. Kenis, "*Two-layer multiplexed peristaltic pumps for high-density integrated micro-fluidics*", University of Illinois at Urbana-Champaign, Department of Chemical & Biomolecular Engineering, 600 S. Mathews Ave., Urbana, IL 61801, USA,2010.

[7] S.Yelleti, E. Wilkins, "*Design of an automated immunosensor device for fast detection of Hantavirus in mice blood*", Department of Chemical and Nuclear Engineering, University of New Mexico, Albuquerque, NM 87106, USA, 2003.

[8] "Stepper Motor Control-AVR Tutorial", Avinash, Extreme Electronics, May 20, 2011.