Advanced Research in Electrical and Electronic Engineering
p-ISSN: 2349-5804; e-ISSN: 2349-5812 Volume 3, Issue 5, October-December, 2016, pp. 375-380
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/Publication.html

Remotely Accessible Robotic Arm: Optimized for Laboratory Demonstration

Jyoti Prasad Nath[1], Hidam Kumarjit Singh[2] and Tulshi Bezboruah[3]

[1,2,3]Department of Electronics and Communication Technology Gauhati University, Assam
E-mail: [1]nathjyoti1993@gmail.com, [2]kumarjit_hidam@yahoo.com, [3]zbt_gu@yahoo.co.in

Abstract—*This paper describes the design and implementation of prototype model for controlling of robotic arm over web in local area network zone. A robotic arm with three degrees of freedom is designed and interfaced to printer port of a local web server computer. An interactive web application is developed to control operation of the robotic arm. Live video of the robotic arm is captured with a commercially available web camera and the video is fed to the interactive web application as visual information for the users. This system can be used as prototype model for demonstration to undergraduate electronic students in Instrumentation laboratory.*

1. INTRODUCTION

Tele-operation and control of robots and robotic systems had been started almost 50 years back for industrial and scientific applications. Both wired and wireless communication techniques were employed in them. Later on, control of robotic systems over Internet and World Wide Web came into being in 1990s. Goldberg *et. al* demonstrated Internet based control of SCARA robot, as in [1]. B.Dalton *et. al* reported a communication framework for access and control of robots over internet, as in [2]. Multi-tier server based system was purposed for remote operation of mobile robots in [3]. Mitsubishi PA10-6CE robotic manipulator was accessed and controlled through internet, as in [4]. In many of the reported works, vendor specific robots and their supervisory application programming interface (API) are used for implementing the systems. In this paper, we describe the design and implementation of a web based system to operate a customized robotic arm remotely in local area network (LAN). Design principle of the proposed system is described in section 2 of this paper. Hardware and software modules of the system are described in section 3 and 4 respectively. Finally, the experimental details, results and discussion are described in sections 5 and 6.

2. PROPOSED SYSTEM

The purposed instrumentation system is based on client-server application, where a robotic arm is interfaced to the server and clients access the robotic arm through a web application running in the server. A simplified functional block diagram of the web based Instrumentation system is shown in Fig.1. Here, a robotic arm is interfaced to printer port of a web server computer through a control circuit. The robotic arm is made accessible to remote client machines of a Local Area Network (LAN) through a web application developed in Microsoft's ASP.NET technology. A user can log-in to a designated website of the system to control operation of the robotic arm. A web camera captures physical movement of the robotic arm in real time and this video is fed back to a web application as visual feedback for the clients to take up control actions.

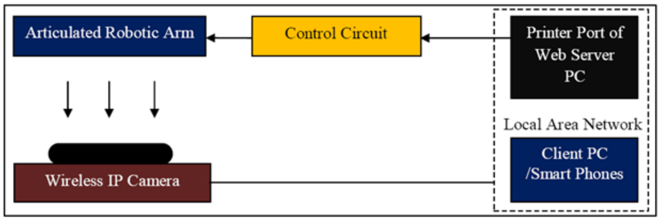


Fig. 1: Functional block diagram of the DAQ system for monitoring and control of robotic arm.

3. HARDWARE MODULES OF THE SYSTEM

3.1 Robotic arm

An articulated robotic arm has been designed with three revolute joints (J_0, J_1, and J_2), three links (L_0, L1, and L2) and one end-effecter as shown below in Fig.2. DC gear motors are used to make the joints, while the links and end-effectors are made of locally available Aluminum frames. The DC motors (M_1, M_2 & M_3) employed at J_0, J_1 and J_2 can move in bidirectional ways and they have torque capacities of 5Kg-force-cm, 2Kg- force-cm and 1 Kg- force-cm respectively and revolution per minute of about 10. The end-effecter of the robotic arm is provided with a hook so that it can be used to lift certain objects. The joint J_0 is attached to a heavier platform so that the robotic arm does not fall during operation.

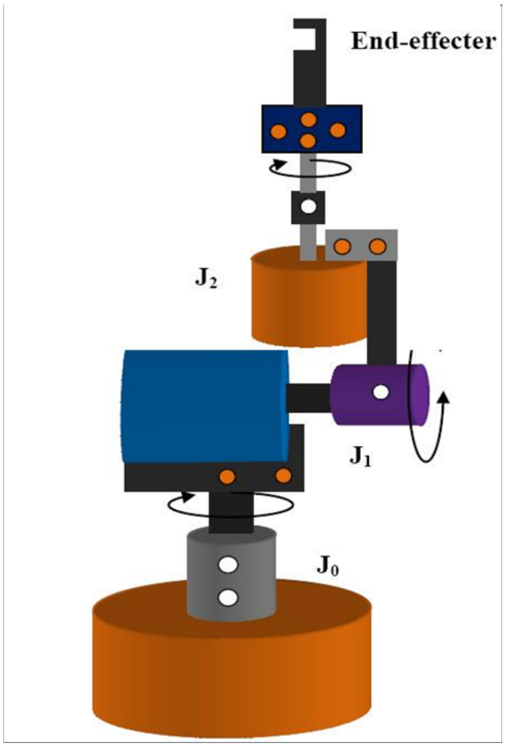


Fig.2: Schematic diagram of an articulated robotic arm.

3.2. Driving circuit the robotic arm

The DC motors used in the robotic arm are controlled with the help of a driver circuit. The driver circuit is designed with two motor driver chips (L293D). The operations of the chips are controlled by a server-side script running in the web server computer. Interfacing of the control circuit with printer port of the server is shown in Fig.3, while its assembled circuit is shown in Fig.4.

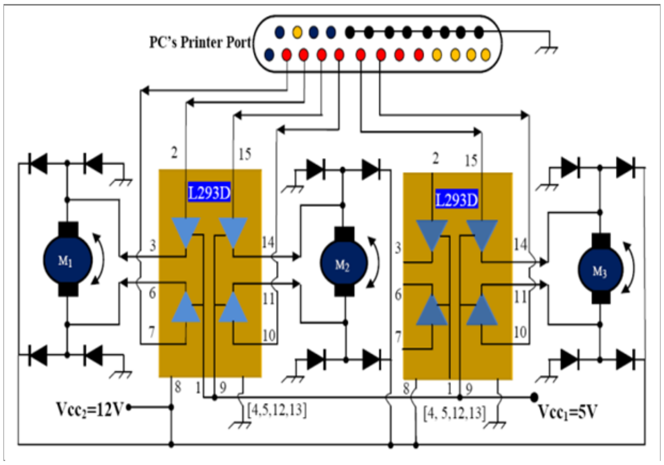


Fig. 3: Schematic circuit for controlling robotic arm; the L293D chips get their control signals from data lines of the printer port.

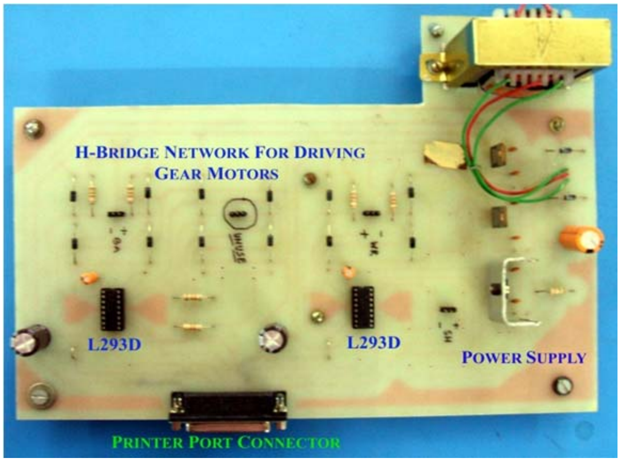


Fig. 4: Snapshot of the assembled circuit board.

A single L293D chip has four on-chip tri-state active high buffers. Two buffers are needed to drive a motor for bidirectional rotation [4]-[5]. Hence, two L293D chips are used for driving three gear motors in the control circuit. When the circuit is switched on, all the tri-state buffers are kept in enabled state, and then they are driven by pulse signals transmitted from the server computer's printer port. In order to rotate a motor in a given direction, its corresponding buffers are driven by two logic pulses. The direction of rotation can be altered any time by reversing the logic levels of the inputs. Server side program can alter the logic level of the control input signals. Pulse width modulation technique is used for controlling the speed of the motors. However, no additional circuit for pulse width modulation is used, because it is taken care of by the server-side script of the web application. Fig.5 shows the interfacing of the circuit to printer port of server computer. A web camera (D-link make: DCS-931L) is used for streaming real time video of robot actions on the web page.

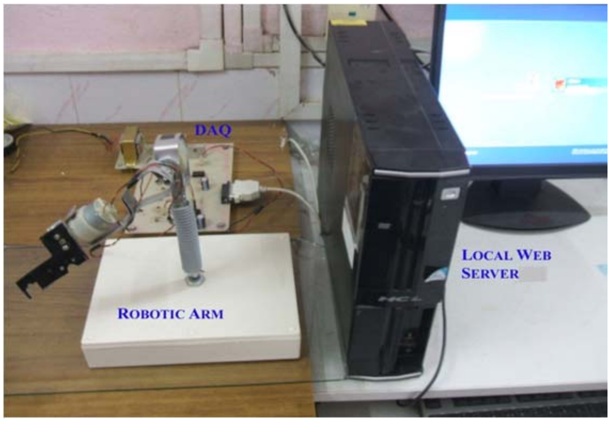


Fig. 5: Interfacing of the control circuit to a web server PC (personal computer).

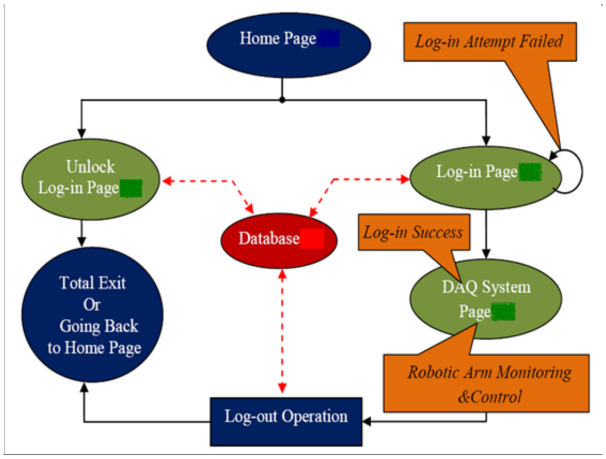
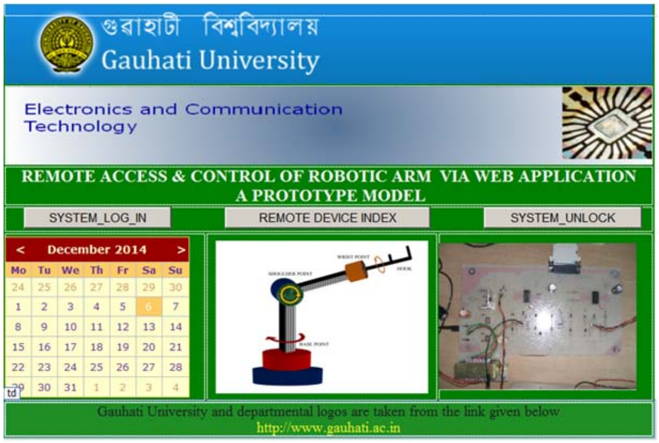


Fig. 6: Architecture of the web application for controlling the robotic arm.

4. SOFTWARE MODULE OF THE SYSTEM

Microsoft's Visual Studio 2010 (Professional Version) has been used to design web application of the present work and server side scripting is done in Visual Basic.NET has been used for developing server side scripts. Fig.6 shows the architecture of the web application developed for controlling the robotic arm over Internet in LAN.


গুৱাহাটী বিশ্ববিদ্যালয়
Gauhati University

Electronics and Communication Technology

REMOTE ACCESS & CONTROL OF ROBOTIC ARM VIA WEB APPLICATION
A PROTOTYPE MODEL

SYSTEM_LOG_IN | REMOTE DEVICE INDEX | SYSTEM_UNLOCK

< December 2014 >

Mo	Tu	We	Th	Fr	Sa	Su
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Gauhati University and departmental logos are taken from the link given below
http://www.gauhati.ac.in

Fig. 7: Snapshot of the Home page for the web application.

Fig.7 shows the snapshot for home page designed. This web page provides links for accessing various web pages included in the architecture of the web application. The web application has a Home page, a Log-in page, an Unlock page and a DAQ system page. This application is designed in such a way that it disallows concurrent access of the robotic arms. When concurrent access is allowed, operational conflict can occur in the robotic arm (for instance one user may try to rotate a motor in one direction while another user may do the opposite). A database is employed for user verification and unlocking of the web application. This web application makes use of a Java applet program to view the live video of the robotic arm in real time.

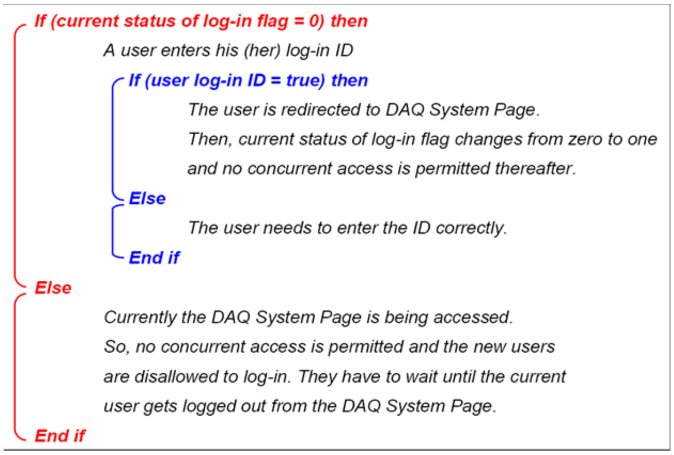

```
If (current status of log-in flag = 0) then
        A user enters his (her) log-in ID
        If (user log-in ID = true) then
                The user is redirected to DAQ System Page.
                Then, current status of log-in flag changes from zero to one
                and no concurrent access is permitted thereafter.
        Else
                The user needs to enter the ID correctly.
        End if
Else
        Currently the DAQ System Page is being accessed.
        So, no concurrent access is permitted and the new users
        are disallowed to log-in. They have to wait until the current
        user gets logged out from the DAQ System Page.
End if
```

Fig. 8: Algorithm of the program for log-in process.

The algorithm of the log-in program is shown above in Fig.8. This algorithm is developed in such a way that concurrency is not allowed in the log-in process. This means that new users are not allowed to access the robotic until he (she) gets logged out from the web application. This condition is implemented by using a log-in flag, as illustrated in Fig.8.

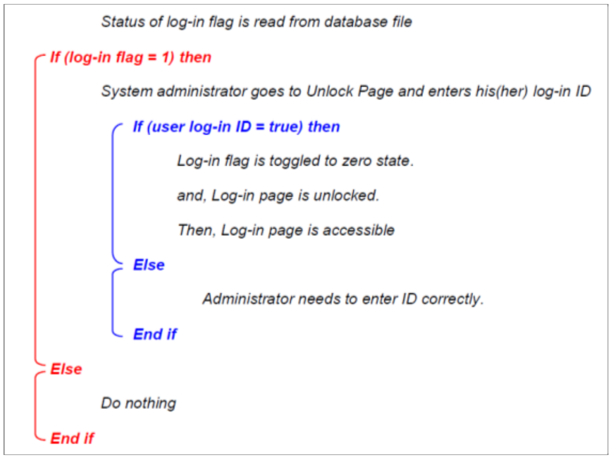
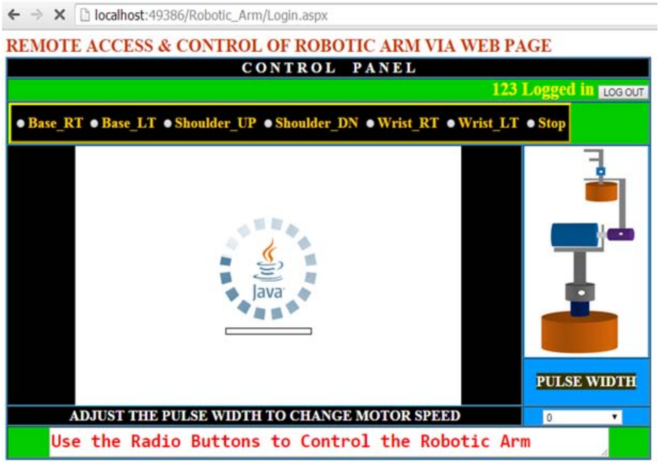

```
        Status of log-in flag is read from database file
If (log-in flag = 1) then
        System administrator goes to Unlock Page and enters his(her) log-in ID
        If (user log-in ID = true) then
                Log-in flag is toggled to zero state.
                and, Log-in page is unlocked.
                Then, Log-in page is accessible
        Else
                Administrator needs to enter ID correctly.
        End if
Else
        Do nothing
End if
```

Fig. 9: Algorithm of the program code in Unlock page.

Fig.9 shows the algorithm for the Unlock page of the web application. Unlock page enables a system administrator to reset the logic status of log-in flag in database. Successful log-out event changes the status of log-in flag from 1 to 0. When a log-out process gets failed or the user leaves the DAQ system page by closing the web browser, the status of log-in flag remains unchanged at logic 1. Then, the web application will enter into locked state; hence it will, later on, will prohibit users from accessing the DAQ system page.



localhost:49386/Robotic_Arm/Login.aspx

REMOTE ACCESS & CONTROL OF ROBOTIC ARM VIA WEB PAGE

CONTROL PANEL

123 Logged in

LOG OUT
● Base_RT ● Base_LT ● Shoulder_UP ● Shoulder_DN ● Wrist_RT ● Wrist_LT ● Stop
Java
PULSE WIDTH

ADJUST THE PULSE WIDTH TO CHANGE MOTOR SPEED

0

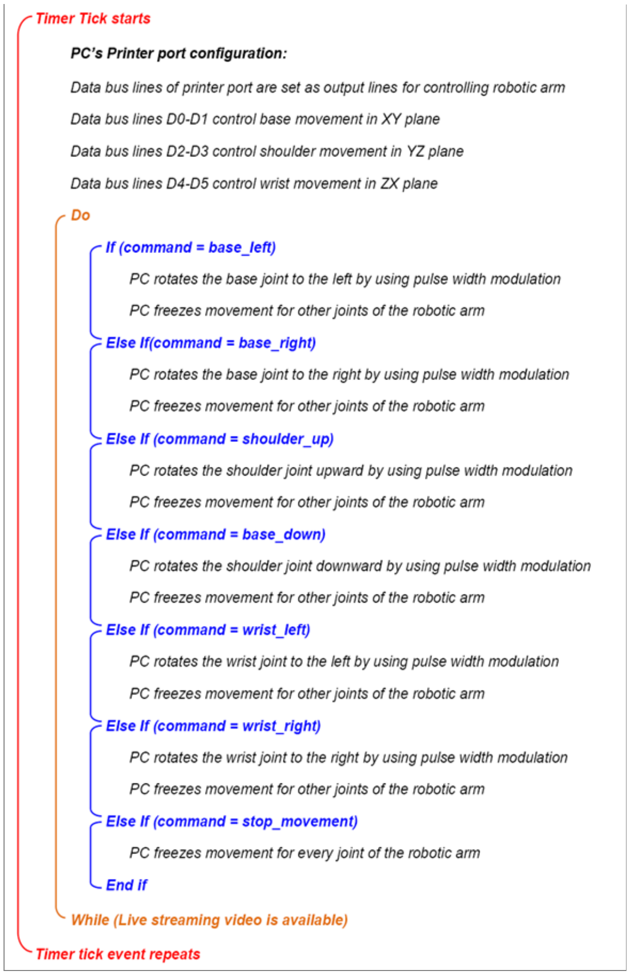
Use the Radio Buttons to Control the Robotic Arm

Fig. 10: Snapshot of the DAQ system page.

Fig.10 shows the snapshot of DAQ system page, while its algorithm is shown in Fig.11. This web page consists of command options for moving a robotic arm in three different planes. A Java applet is used for displaying live video of robot actions on the web page. By looking at the real time video, a user can control a robotic arm to carry out some of the tasks. In this web page, a user can rotate the base, arm and wrist of the robotic arm in a controlled manner. The radio buttons *Base_RT* and *Base_LT* cause the base joint of the robotic arm to rotate towards right and left directions respectively.The *Shoulder_UP* and *Shoulder_DN* rotate shoulder joint to rotate in upward and downward directions respectively.The *Wrist_RT* and *Wrist_LT* buttons rotate wrist joint to right and left directions respectively. The Stop button freezes joint activities of the robotic arm. The drop down text box named as *PULSE WIDTH* is used for varying duty cycle of control signals that are needed to drive the robotic arm. Fig.12 shows some of the Visual Basic.NET statements used for accessing of printer port in the web application. In this web application, printer port of the server computer has been accessed by using a DLL file (named as inpout32.dll), as in [12]-[14]. A LOG-OUT button is used for closing a control and monitoring session.

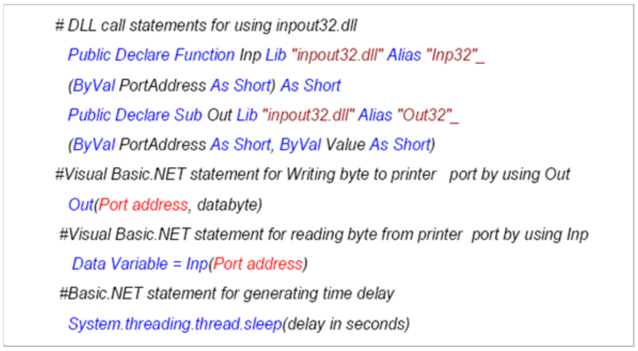
5. FUNCTIONAL TESTING OF THE SYSTEM

The web application is configured to run as local website of IIS (Internet Information Service) Express 7.5 in the program development machine. The Home page is browsed by entering its local website address (http://localhost:49386/Home.aspx) in the address bar of web browser. A web enable camera (having both day and night vision capability) is connected to LAN zone by assigning a unique TCP/IP address.



```
Timer Tick starts
    PC's Printer port configuration:
    Data bus lines of printer port are set as output lines for controlling robotic arm
    Data bus lines D0-D1 control base movement in XY plane
    Data bus lines D2-D3 control shoulder movement in YZ plane
    Data bus lines D4-D5 control wrist movement in ZX plane
    Do
        If (command = base_left)
            PC rotates the base joint to the left by using pulse width modulation
            PC freezes movement for other joints of the robotic arm
        Else If(command = base_right)
            PC rotates the base joint to the right by using pulse width modulation
            PC freezes movement for other joints of the robotic arm
        Else If (command = shoulder_up)
            PC rotates the shoulder joint upward by using pulse width modulation
            PC freezes movement for other joints of the robotic arm
        Else If (command = base_down)
            PC rotates the shoulder joint downward by using pulse width modulation
            PC freezes movement for other joints of the robotic arm
        Else If (command = wrist_left)
            PC rotates the wrist joint to the left by using pulse width modulation
            PC freezes movement for other joints of the robotic arm
        Else If (command = wrist_right)
            PC rotates the wrist joint to the right by using pulse width modulation
            PC freezes movement for other joints of the robotic arm
        Else If (command = stop_movement)
            PC freezes movement for every joint of the robotic arm
        End If
    While (Live streaming video is available)
Timer tick event repeats
```

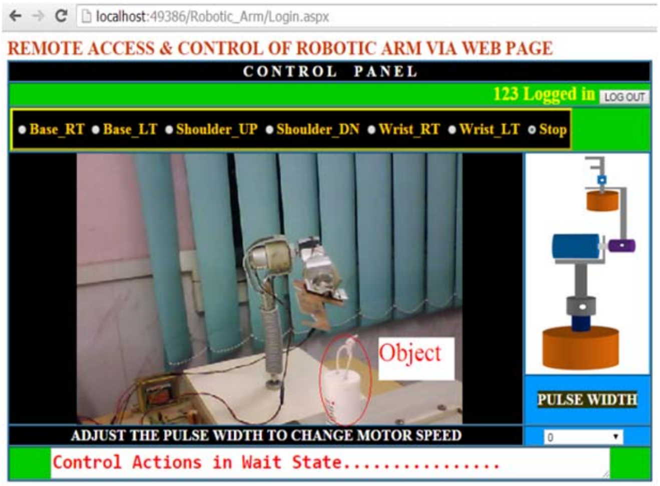
Fig. 11: Algorithm of the program for DAQ system page.



```
# DLL call statements for using inpout32.dll
  Public Declare Function Inp Lib "inpout32.dll" Alias "Inp32"_
  (ByVal PortAddress As Short) As Short
  Public Declare Sub Out Lib "inpout32.dll" Alias "Out32"_
  (ByVal PortAddress As Short, ByVal Value As Short)
#Visual Basic.NET statement for Writing byte to printer   port by using Out
  Out(Port address, databyte)
#Visual Basic.NET statement for reading byte from printer  port by using Inp
  Data Variable = Inp(Port address)
#Basic.NET statement for generating time delay
  System.threading.thread.sleep(delay in seconds)
```

Fig. 12: Visual Basic.NET pseudo-code used to access printer port from the web application.

The software setting of the camera is adjusted to serve streaming video to the DAQ system page in the form of Java applet. The hyperlink buttons provided on web page are clicked one after another in order to examine their responses. The Log-in page and Unlock page are tested by using different user names and passwords.

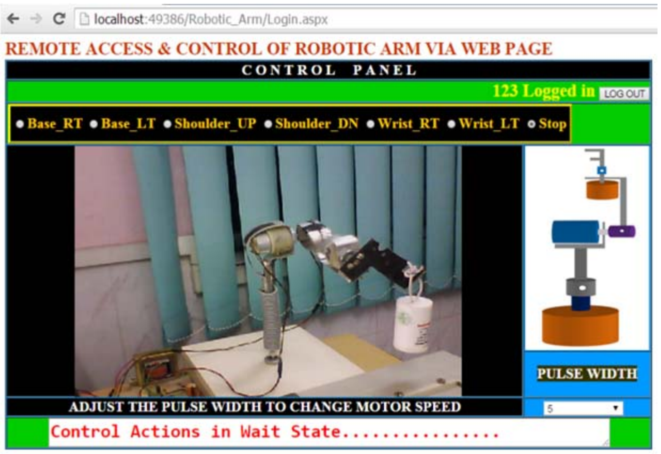

localhost:49386/Robotic_Arm/Login.aspx

REMOTE ACCESS & CONTROL OF ROBOTIC ARM VIA WEB PAGE

Control Actions in Wait State................

CONTROL PANEL
123 Logged in
LOG OUT
● Base_RT ● Base_LT ● Shoulder_UP ● Shoulder_DN ● Wrist_RT ● Wrist_LT ○ Stop
Object
PULSE WIDTH
ADJUST THE PULSE WIDTH TO CHANGE MOTOR SPEED
0

(a)

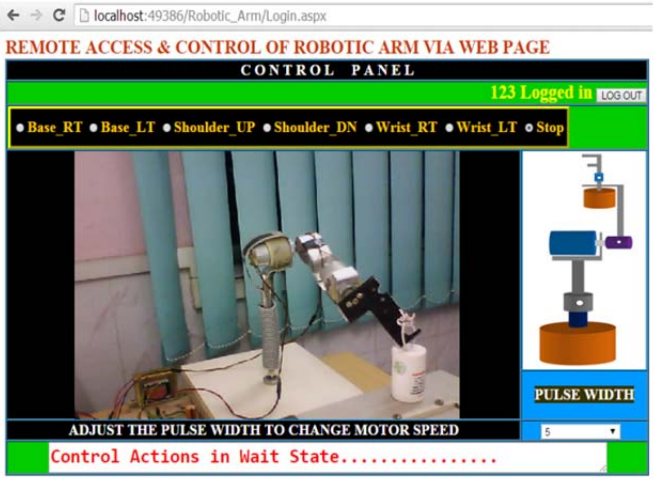

localhost:49386/Robotic_Arm/Login.aspx

REMOTE ACCESS & CONTROL OF ROBOTIC ARM VIA WEB PAGE

Control Actions in Wait State................

CONTROL PANEL
123 Logged in
LOG OUT
● Base_RT ● Base_LT ● Shoulder_UP ● Shoulder_DN ● Wrist_RT ● Wrist_LT ○ Stop
PULSE WIDTH
ADJUST THE PULSE WIDTH TO CHANGE MOTOR SPEED
5

(b)

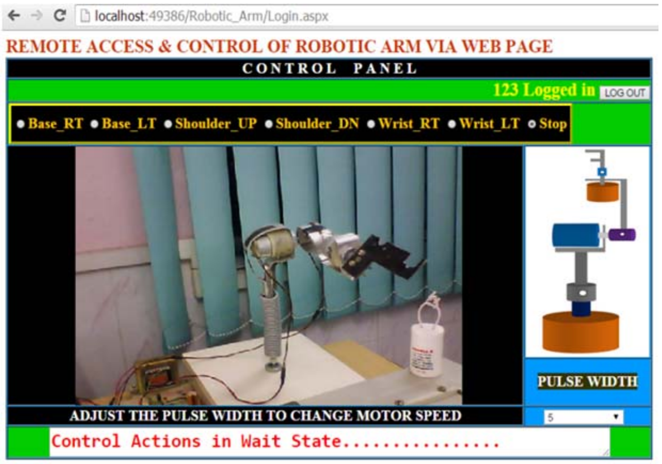

localhost:49386/Robotic_Arm/Login.aspx

REMOTE ACCESS & CONTROL OF ROBOTIC ARM VIA WEB PAGE

Control Actions in Wait State................

CONTROL PANEL
123 Logged in
LOG OUT
● Base_RT ● Base_LT ● Shoulder_UP ● Shoulder_DN ● Wrist_RT ● Wrist_LT ○ Stop
PULSE WIDTH
ADJUST THE PULSE WIDTH TO CHANGE MOTOR SPEED
5

(c)


localhost:49386/Robotic_Arm/Login.aspx

REMOTE ACCESS & CONTROL OF ROBOTIC ARM VIA WEB PAGE

Control Actions in Wait State................

CONTROL PANEL
123 Logged in
LOG OUT
● Base_RT ● Base_LT ● Shoulder_UP ● Shoulder_DN ● Wrist_RT ● Wrist_LT ○ Stop
PULSE WIDTH
ADJUST THE PULSE WIDTH TO CHANGE MOTOR SPEED
5

(d)

Fig. 13: Snapshot of the Robotic arm in action (a) The end-effecter ready to pick up an object (b) The end-effecter lifts up the object (c) The end-effecter about to release the object (d) The end-effecter releases the object.

6. RESULTS AND DISCUSSION

As far as the web application is concerned, the Login page is found to be capable of discriminating different user credentials supplied during the testing. Further, it also blocks concurrent access to DAQ system page. The Unlock page is found to be capable of unlocking DAQ system page from a locked state. The robotic arm is found to respond properly to the command provided by the user, and it is also able to change the position of a dummy object successfully. Maneuverability of the robotic arm in different directions has been tested by applying control pulse manually to control circuit of the robotic arm. Every joint of the robotic arm is able to move smoothly in clockwise as well as anticlockwise directions.

Fig.13 shows the snapshots the DAQ system page recorded during the rendezvous tasks. These snapshots have shown that the robotic arm can be controlled from a web page. The rendezvous tasks are found to be a bit difficult, because of the fact that position of the dummy object in three dimensional space cannot be ascertained precisely from the two dimensional image provided by web camera. So, additional camera will be needed to provide additional information on depth of the objects in the vicinity of the robotic arm. Besides, there is no attachment of position sensor in the robotic hardware to help a user while controlling the robotic arm. Future direction of the current work will be dealing increasing degree of freedom of the robot along with incorporation of position sensors and extra camera for vision assistance.

7. CONCLUSION

A simple robotic arm has been designed and controlled from an interactive web application. This system can be used as laboratory kit for practical demonstration purposes. The performance of the system can be further improved by adding suitable electromechanical hardware and sensors and these things will be take care of in future direction of the work.

REFERENCES

[1] Goldberg, K., Gentner, S., Sutter, C., and Wiegley, J. (2000). The mercury project: A feasibility study for Internet robots, IEEE Robotics and Automation Megazine, pp.35-40, March 2000 (available online at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=833573).

[2] B. Dalton and K. Taylor, Distributed robotics over the internet, IEEE robotics and automation magazine, pp.22-27, June 2000(Available online at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=848264).

[3] H. Hu, L.Yu, P. W. Tsui, Q. Zhou, "Internet-based Robotic Systems for Teleoperation", *International Journal of Assembly Automation*, Vol. 21, No. 2 (available online: http://cswww.essex.ac.uk/staff/hhu/Papers/Journal-AA-2001.pdf).

[4] C. M. Wronka & M.W. Dunnigan, "Internet remote control interface for a multipurpose robotic arm", *International Journal of Advanced Robotic Systems,* Vol. 3, No. 1, ISSN 1729-8806, pp. 179-183, 2006.

[5] Quadruple half-H drivers: L293D datasheet (available online: http://www.ti.com/lit/ds/symlink/l293d.pdf).

[6] DC motor and L293D (Available online: https://electrosome .com/dc-motor-l293d-pic-microcontroller/).

[7] R. Udayakumar, K.P. Thooyamani, V. Khanaa, "A Comparison of J2EE and .NET as Platforms for Developing E-Government Applications", International Journal of Engineering Research and Development, vol. 7, no. 1, pp. 116-121, 2003(available online: www.ijerd.com.

[8] N. Paul, D. Evans, "Comparing Java and .NET security: Lessons learned and missed", computers & security 25 (2006) 338–350, Elsevier, 2006; doi:10.1016/j.cose.2006.02.003.

[9] C. de la Torre, D.Carmona, .NET Technology Guide for Business Applications, Microsoft Press , Redmond, Washington 98052-6399 , 2013.

[10] M. Halvorson, Microsoft ® Visual Basic® 2010: Step by step, Microsoft Press , Redmond, Washington 98052-6399 , 2010.

[11] M. Gibbs, D. Wahlin, Professional ASP.NET 2.0 AJAX, Wrox Publication, 2007.

[12] Programming Tools for Port I/O and Interrupts (Available online: http://janaxelson.com/parport.htm#Programming)

[13] How Inpout32.dll works? (available at: http://logix4u.net/parallel-port/18-how-inpout32dll-works)

[14] InpOut32 and InpOutx64 (available at: http://www.highrez.co.uk/).