We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter 13

Remote Process Control and Monitoring Using Matlab

Vedran Vajnberger, Semir Silajdžić and Nedim Osmić

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/46464

1. Introduction

Remote control is one of the best solutions for managing inaccessible systems. Many researches have been made in the field of the remote control [1-4]. These researches have improved certain aspects of industry, medicine, military, etc. The benefits of remote control are numerous such as: operating in hazard environment, telemedicine, missile guidance, etc. The most important characteristic of remote control is operating in real-time as shown in papers form references [5-7].

Many applications in the field of medicine and industry use different kind of motor-based systems especially stepper motors because of their wide-range of sufficient characteristics like the fact that they can be used as constant power devices with accurate positioning and fast response [8-12].

In today's society, robots are used in various areas especially in those where high precision is required. Some of the examples where robotic arms found their appliance are: in vehicle construction where efficiency and reliability are required, in chemical industry where environment is not suitable for human, in medicine where robotic arm precision is used in operations, etc [13]. Robots have improved life standards and we are upgrading their performances in order to make our lives easier and more comfortable.

This chapter describes implementation of the proposed remote control of the stepper motor and robotic arm with five DOF via web server and VNC server [14, 15]. VNC server was used to receive visual feedback from robotic arm. The quality of image was important and it couldn't be sent via MATLAB server because real time characteristic would be lost. The decision to use microcontroller was based on its characteristics. Developing such system is cheaper than developing on other platforms such as PLC or FPGA.

The realized system exhibits the following:

1. implementation based on PIC16F877a microcontroller,
2. adjustment
   a. of the velocity, number of steps and rotation direction of the stepper motor,
   b. of the rotation direction of the DC motors inside each joint of the robotic arm
3. precise control over RS-232 serial communication,
4. extension to remote control the whole system,
5. feedback
   a. data acquisition and transmission to confirm the proper operation of the stepper motor
   b. visual feedback and contact sensors to confirm the proper operation of the robotic arm
6. realized GUI.

Both systems consist of all previously mentioned characteristics, where character 'a)' refers to stepper motor and character 'b)' to robotic arm.

This chapter is structured as follows: In section 2. whole data analysis of the realization of the system in accordance with previously specified requirements is described. Section 3. provides hardware description that was realized manually. Section 4. explains hardware programming.

2. Problem analysis

The problem of realizied systems that remotely control stepper motor and robotic arm can be divided into several interconnected units, as it will be described in the following text and shown on Figure 1. and Figure 2. To understand the problem and its solution, a brief description of each part will be provided in the rest of this section.

a. Control via microcontroller

The Microchip microcontroller PIC16F877a © generates impulses on four pins that are used to stimulate the movement of the stepper motor gear as shown in Figure 2. Stepper motor consists of multiple "toothed" electromagnets ranged around a coil of iron. Those electromagnets need to be stimulated by some kind of external control unit, in our case a microcontroller. The microcontroller provides electrical impulses which stimulate the gear's teeth to magnetically be attracted to electromagnet's teeth. When the gear's teeth are thus aligned to the first electromagnet, they are slightly offset from the next electromagnet. When the next electromagnet is turned on and the first is turned off, the gear rotates slightly to align with the next one, and from there the process is repeated. Each of those slight rotations is called a "step," with an integer number of steps making a full rotation. In that way, the motor can be turned by a precise angle [16].

The impulses (U1÷U4) and response on Figure 2. show the movement of the stepper in one direction by four steps. Each step corresponds with one triggering impulse.

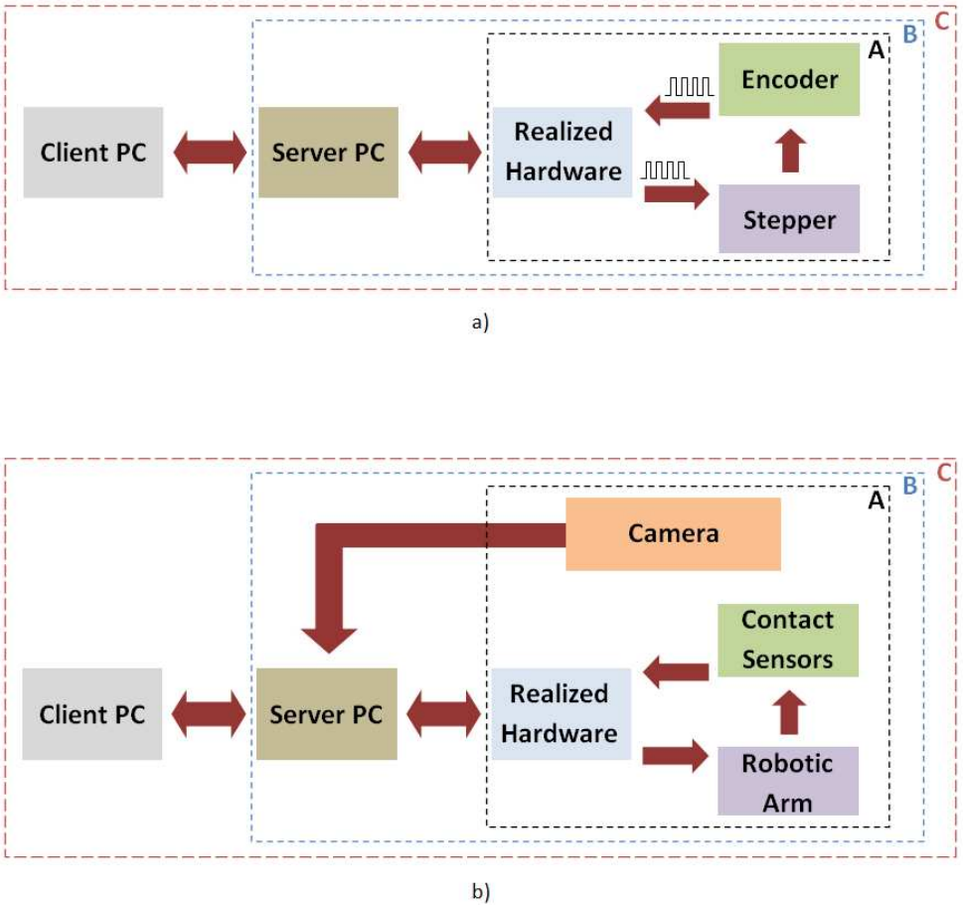


Figure 1. a) Interconnected units of stepper motor system; b) Interconnected units of robotic arm system

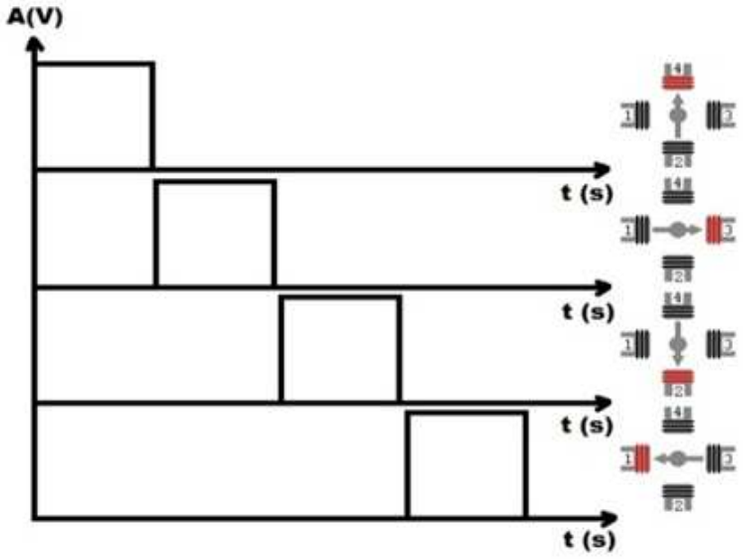


Figure 2. Triggering impulses and gear response

If triggering impulses are reversed (U4÷U1) movement of the motor is in the opposite direction. The length of the triggering impulses depends on desired velocity. The width

between those impulses is infinitesimal. The shorter the length of the triggering impulses is, the higher velocity becomes; i.e. to reach 60 [rpm], the length of each triggering impulse needs to be 5 [ms], to reach 120 [rpm] length should be 2.5 [ms]. As earlier mentioned, besides velocity control, there was the urge for exact positioning. This task was realized by sending a certain amount of triggering impulses in the correct order.

Even though stepper motors are used in open-looped systems because of their characteristics, in this case an additional encoder was installed on the system. Its purpose is to check if the stepper responses accurately to the given commands i.e. combination of impulses.

Similar to the robotic arm (Figure 1.b), Microchip microcontroller PIC16F877A© generates impulses on ten pins (PORT D and two pins from PORT C) which are triggering relays. Signals from relays are used to stimulate the movement of the DC motors implemented inside joins of the robotic arm.

For every degree of freedom (DOF) two pins of microcontroller and two relays are assigned (pins RD0 and RD1 for base, RD2 and RD3 for shoulder, RD4 and RD5 for elbow, RD6 and RD7 for wrist and RC0 and RC1 for fist).

Depending on the state of two pins, there are four situations:

- if both pins are low, two relays controlled by them are open and motor of the appropriate DOF is not running,
- if one pin is high and another is low, current flows in one direction and motor is running in appropriate direction,
- for opposite state of pins, motor is running in opposite direction,
- 'forbidden combination' is when both pins are high, because then both relays are active and source is short circuited.

Figure 3. explains the movement of robotic arm.

b. Control via RS-232

In this chapter, the advantages of a microcontroller were used to establish a communication with the server PC. The used communication is the serial communication RS-232. Serial communication is the most common low-level protocol for communicating between two or more devices [17, 18]. Texas Instruments' MAX232 is used to make adjustment between microcontroller's TTL logic level (5V÷0V) and logic level for RS232 standard (-12V÷12V). An example of the conversion is shown in Figure 4.

The communication was established through MATLAB using three m-files. One m-file was written to create serial port object and to configure its properties. The communication between the server PC and the microcontroller is realized by using second m-function called "send". It has five input arguments: *send(s1, message1, message2, message3, message4)*. The first input argument is the name of a created serial port object. Other arguments are one-byte values that are sent to the microcontroller. Those values represent desired mode of operation (velocity or positional mode), direction of rotation, velocity and number of steps (available only in positional mode). The first bit of message1 is start bit (1 for start of

rotation and 0 for stop). The second bit represents desired mode of operation (1 for velocity and 0 for positional mode). The third bit determines direction of rotation. The last five bits of message1, together with message2, represent desired velocity (13 bits for velocity). Message3 and message4 are low and high bytes of desired number of steps. The last m-file ends the serial port session.

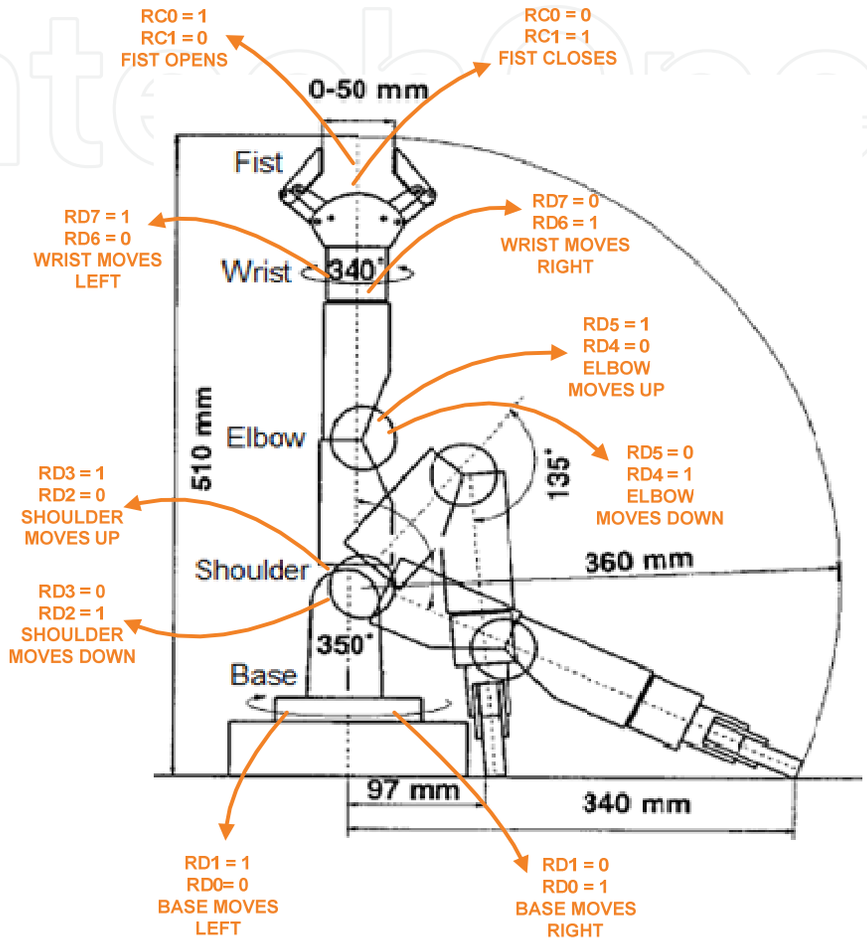


Figure 3. Movement description of robotic arm

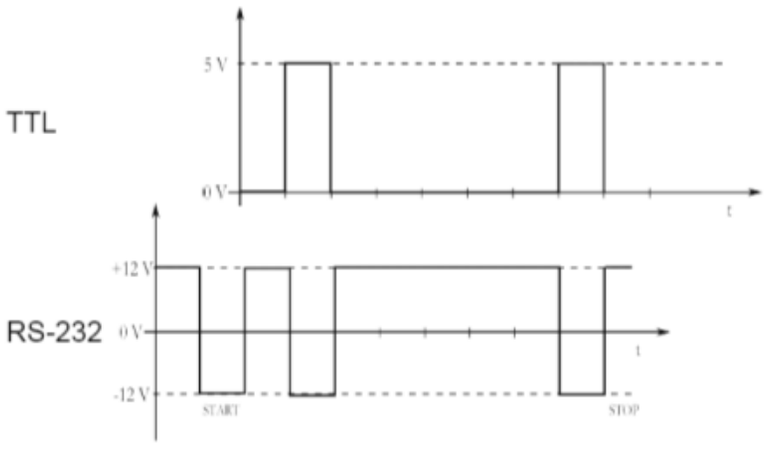


Figure 4. Conversion TTL logic level into RS232 logic level

For the robotic arm system, communication was established through MATLAB using two m-files. First m-file creates serial port and configures its properties. The communication between the server PC and the microcontroller is realized using second m-file. In this m-file, function was created to collect data set by user inside GUI.

c. Remote control via web server

Because MATLAB was used to send control commands to the microcontroller, an additional toolbox called TCP/UDP/IP was installed. This toolbox establishes a connection between two computers (server and client) using the TCP/IP protocol as shown in Figure 5.

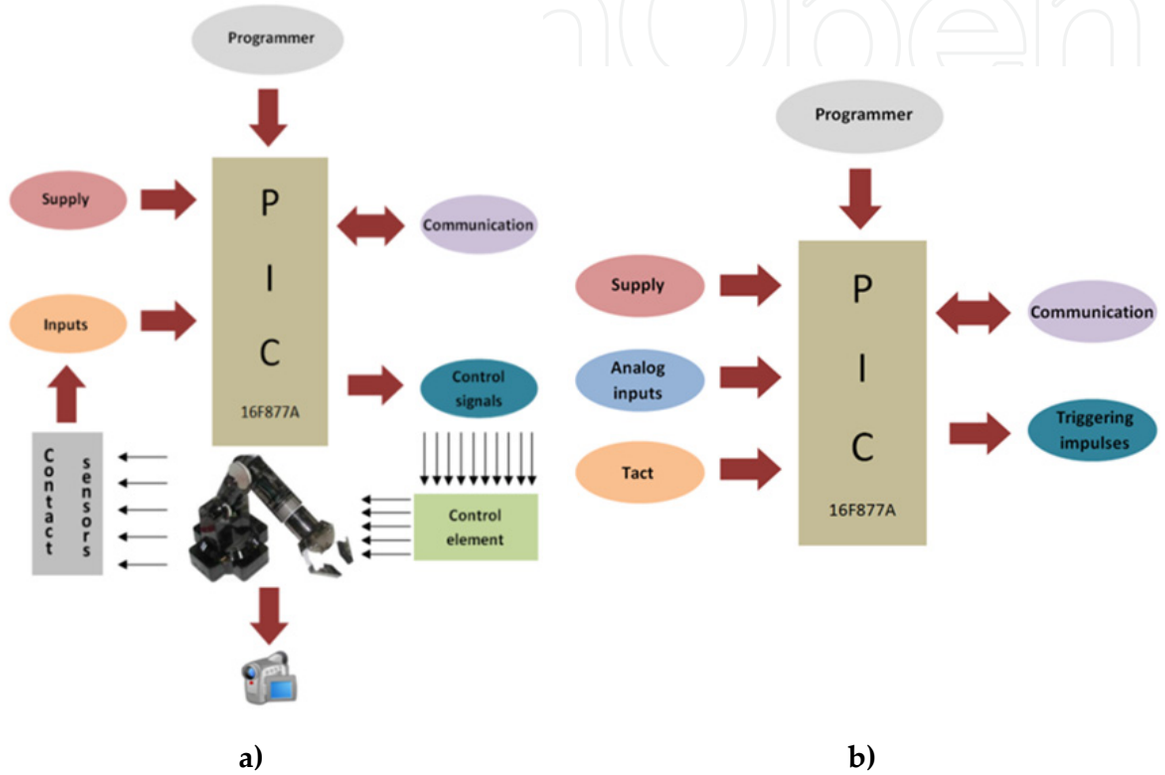
Figure 5. Established connection

One of the computers is used as the server, while the other one is used as a client. The web server is established on the server computer. All commands that control the stepper and robotic arm are sent from the client computer. On the other hand, all feedback needed for acknowledgment of the proper remote control, i.e. did the stepper reached the desired velocity or did robotic arm reached assigned position, is send from the server to the client via established connection. All that was needed to establish a connection was provided in the additional toolbox using the created m-files such as pnet, pnet_putvar and pnet_getvar. Using the pnet m-file a handler called 'con' was created. This handler was used in combination with m-files pnet_putvar and pnet_getvar to send the necessary data to the connection and to collect that data from connection respectively. To access the visual information of the robotic arm acquired by the camera, VNC server was used. The camera is connected to the server computer and by using VNC server we can gain access to the server from a remote desktop.

Two m-files were created (for host and client) using previously mentioned m-files. The required toolbox can be found in the references along with all explanations for each m-file [19].

3. Hardware structure

Figure 6. shows detailed hardware structure of the implemented distributed systems. It is a product of fully independent work.



a) **b)**

Figure 6. a) Block structure of stepper motor system; b) Block structure of robotic arm system

Production implied the implementation of the entire hardware circuit and construction of work algorithm. The structure consists of a microcontroller and controlling elements for both systems.

The stepper motor system is operated with commands from host PC which are sent through serial connection. Microcontroller interpretates those commands and generates trigger impulses on PORT D. These impulses are sent to unipolar transistors which are combined to form a power amplifier. This power amplifier is directly connected to the step motor. The step motor has an encoder disc mounted to its shaft. The encoder recognizes alterations from encoder disc and sends them as impulses to the analog input on PIC. These impulses are used to determine proper work of the stepper motor. Figure 7. shows the most important components used in this hardware realization.

The robotic arm, shown in Figure 3., has five degrees of freedom modeled after the human arm. The controlling element consists of ten relays (Figure 8.), all assembled in accordance with scheme on Figure 9.

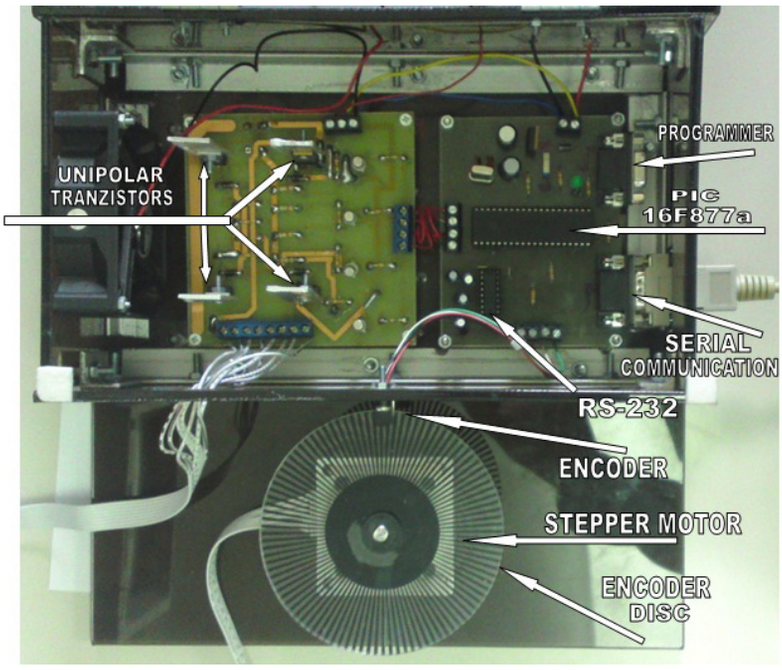


Figure 7. Realized system for stepper motor control

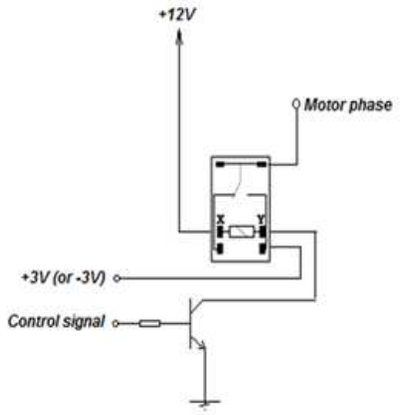


Figure 8. Relay scheme

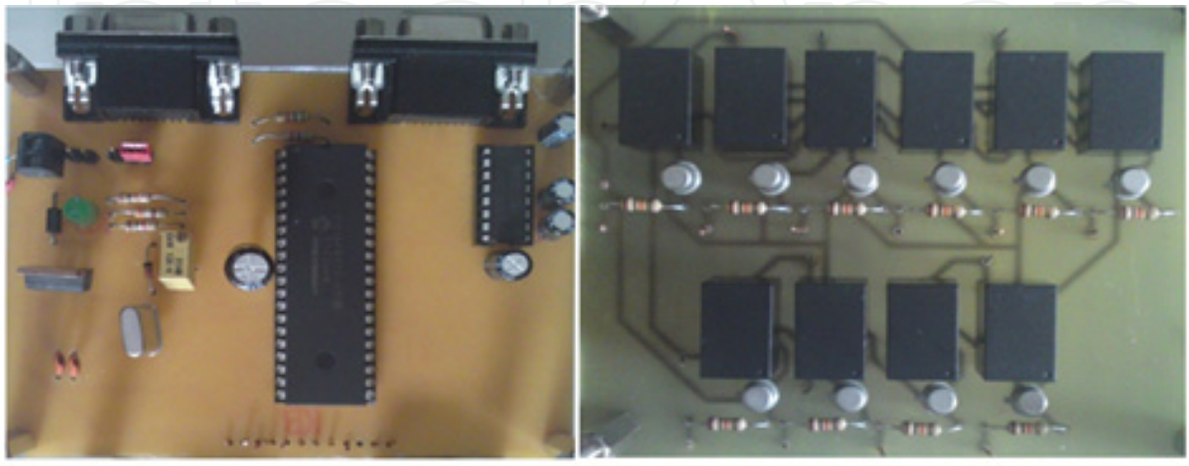


Figure 9. Microcontroller sheme and controlling elements – relays

The operating princip of the relay is simple. When current flows between pins X and Y, it leads to the creation of a magentic field. The influence of the magnetic field causes a change in the position of the switch inside the relay. Control signals from microcontroller are sent to the controlling element. Those control signals need to enable the flow of current which is necessary to trigger the DC motors placed inside the joints of the robotic arm. The system is operated from the client's computer GUI. The commands from server are sent through serial connection. Microcontroller interpretates those commands and generates the control signals. These signals are sent to the controlling element made of relays. The controlling element is directly connected to the robotic arm. The proper work of the robotic arm is monitored via camera. There are also contact sensors on each joint of the arm used to prevent movements that could cause a malfunction. Those signals are transmitted back to the microcontroller and, in the case of a possible malfunction, the movement of the affected joint will be stopped immediately. Figure 9. shows the individual parts of the realized structure and Figure 10. shows the final structure.

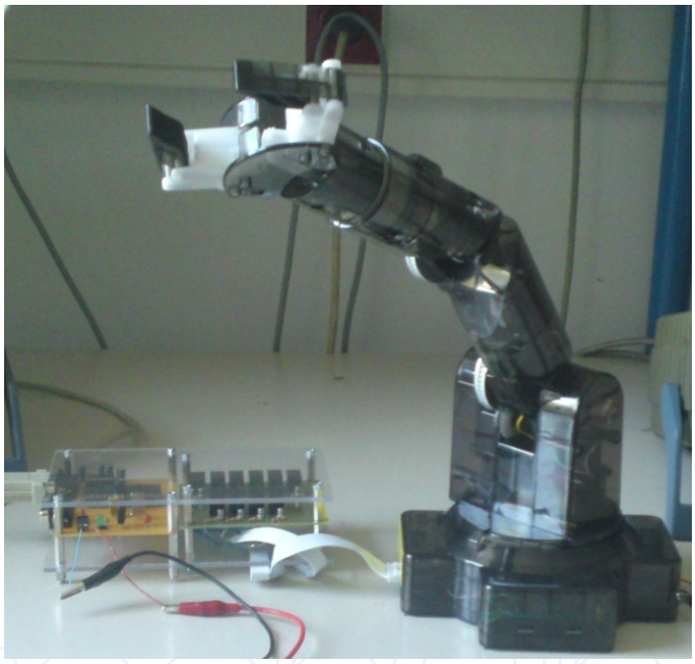


Figure 10. Final structure of the robotic arm system

4. Hardware programming

a. Stepper motor

After making the hardware, the next step was to program the circuit itself. The code is written in CCSC Compiler, version 4, from manufacturer Custom Computer Services Inc. [20]. The user chooses mode of operation, desired direction of rotation, velocity and number of steps using GUI. That information is collected into four 8-bits values and sent to server PC via ETHERNET. Server PC forwards those messages to microcontroller using RS-232. Microcontroller analyses received messages and sets/resets appropriate bits according to chosen mode, direction, velocity and number of steps. In the main program, if START bit is

set, microcontroller generates appropriate sequence of triggering impulses based on chosen direction and velocity. The width of one impulse is given by:

$$T_{width} = \frac{60\,000}{RPM \cdot 200}\,[ms] \tag{1}$$

Timer1 is set to appropriate value so it overflows every 100 ms and an interrupt is generated. In Timer1 Interrupt Service Routine microcontroller sends number of steps made in last 100 ms to server PC via RS-232. Timer0 is used as counter. Output of encoder is led to pin RA4/T0CKI. Timer0 increments on every step. Client PC uses received information about number of steps made, calculate the velocity and shows it in GUI (Figure 11.).

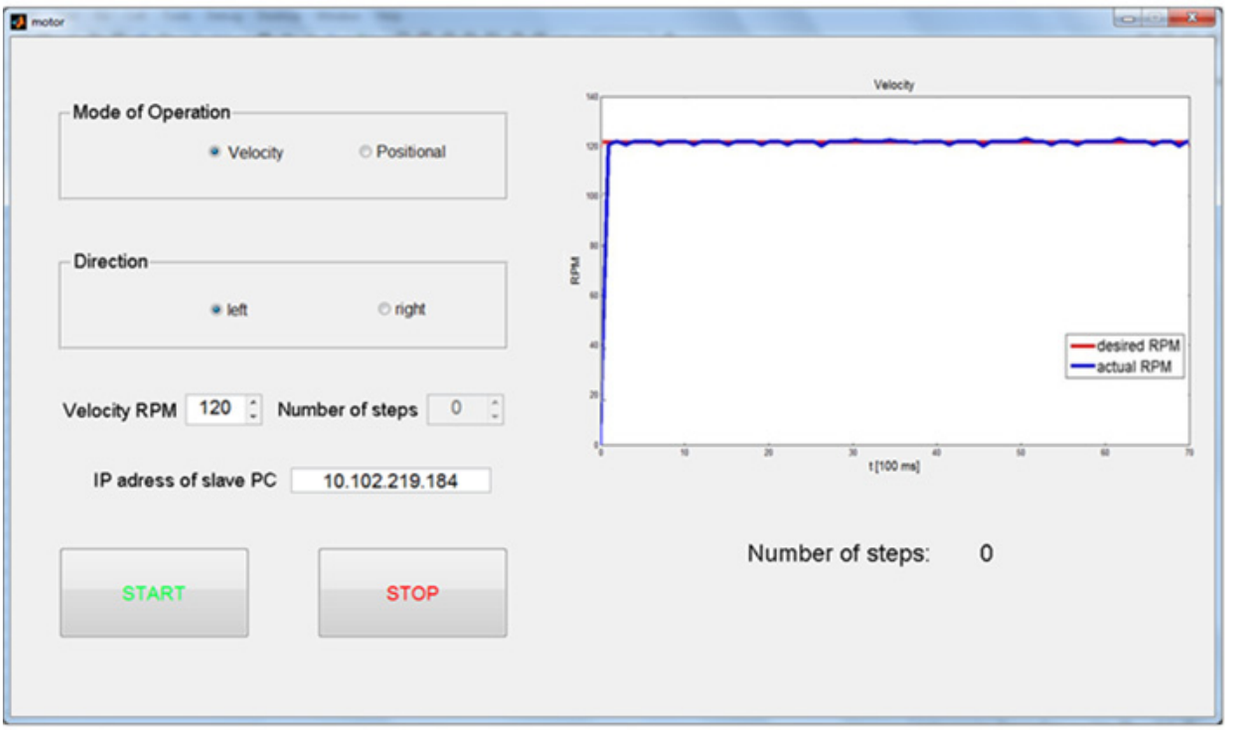


Figure 11. Graphical User Interface for stepper motor system

How hardware is programmed for stepper motor is shown on Figure 12.

b. Robotic arm

After making the hardware, the next step was to program the circuit itself. The code is written in MPLAB IDE v7.5 [21]. Figure 13. shows how hardware is programmed.

To establish communication via ETHERNET, Real VNC program [22] and MATLAB Server are used. Real VNC program is used to obtain visual feedback by camera, and MATLAB Server is used for transmission of control messages from client PC to server PC. VNC Server was installed on server PC (PC connected to realized hardware structure), while VNC Viewer was used by client PC. By running the VNC Viewer and entering IP address of server PC, connection between two computers is established.

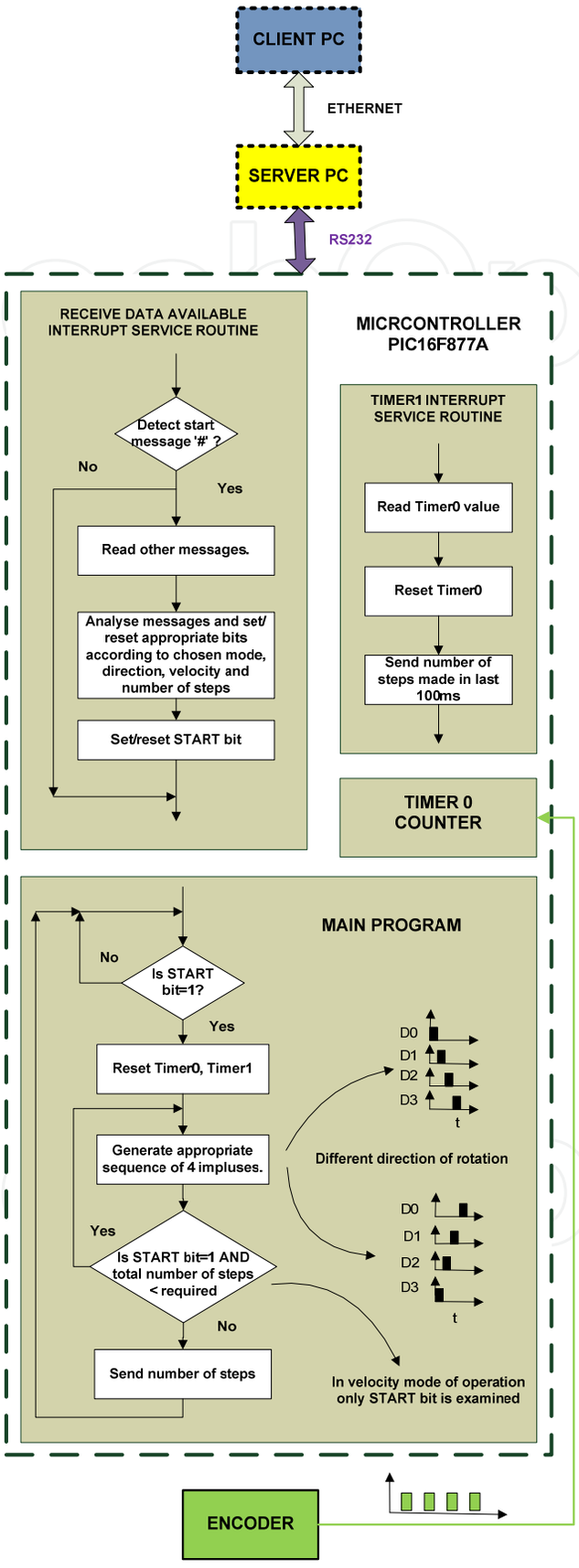


Figure 12. Schematic display of the software for hardware structure with stepper motor

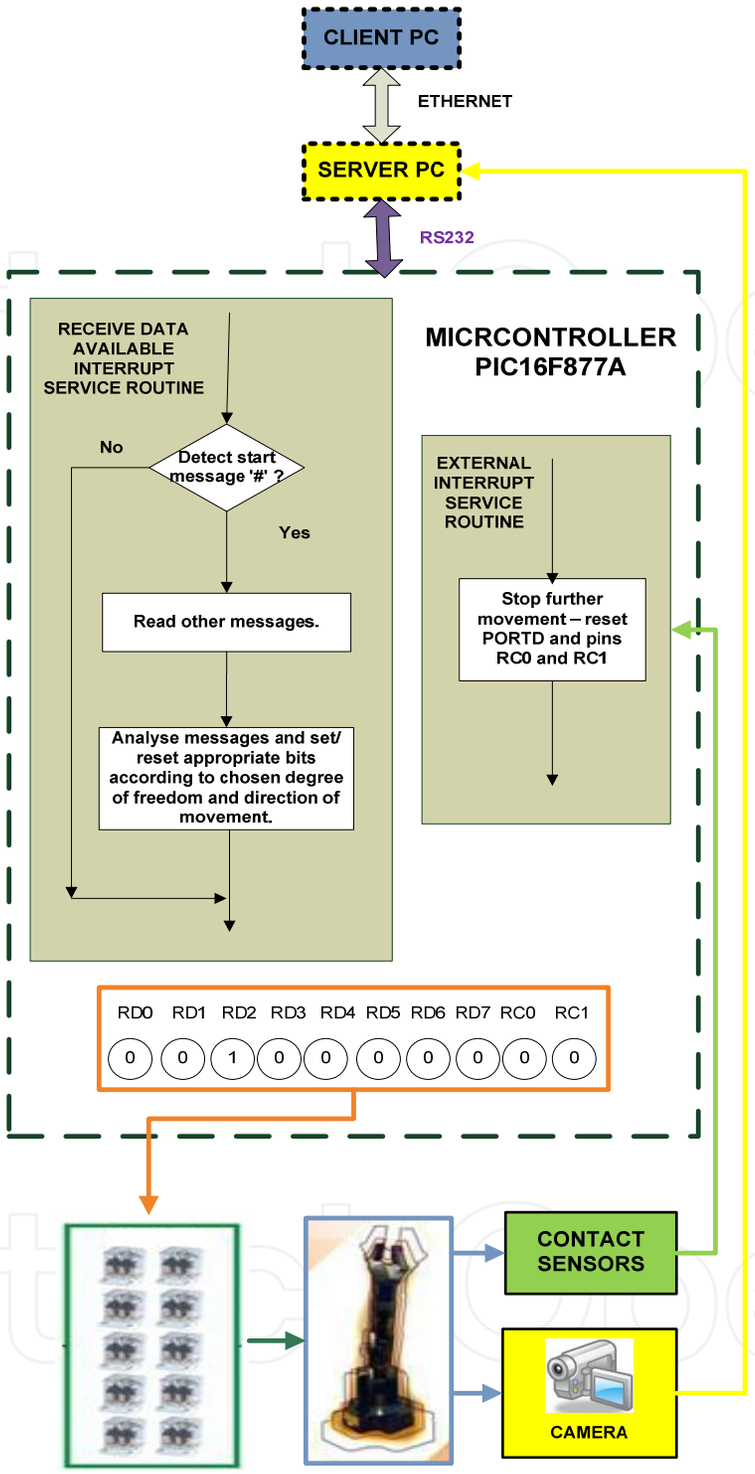


Figure 13. Schematic display of the software for hardware structure with robotic arm

When user chose degree of freedom and direction of movement, function from MATLAB is called. This function codes user's requirement into a short message which is sent to server and then to microcontroller via RS-232. Depending on the content of received message, microcontroller generates appropriate value to the PORT C or PORTD which is described in subchapter 2. section A.

Using the GUI (Figure 14.) the user chooses degree of freedom to manipulate with and the direction of movement.

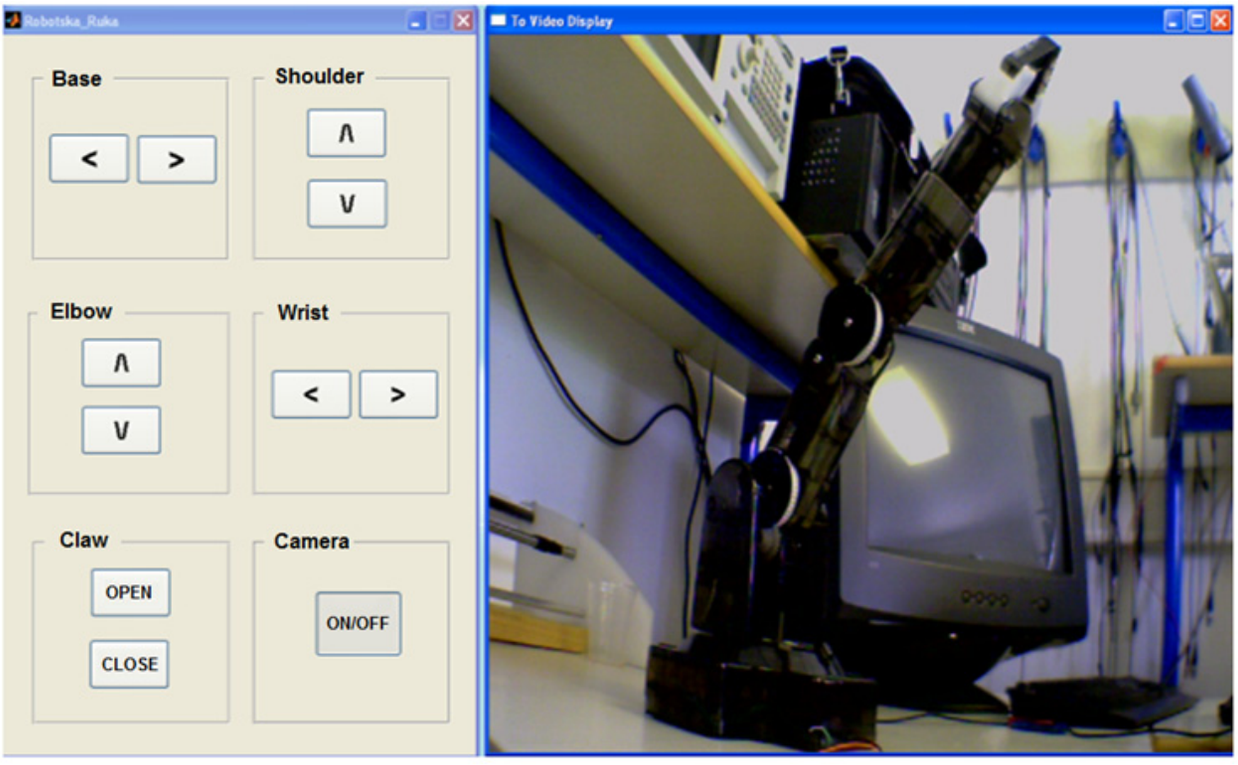


Figure 14. Graphical User Interface for robotic arm system

The user monitor movement of robot arm by camera. Beside this visual feedback, for every DOF contact micro-sensor is implemented to avoid possible damage of the arm. When DOF reaches its final position, the micro-sensor becomes active and further movement is stopped.

5. Testing results

The effectiveness of a stepper motor is rated by the response of a single phase to the provided trigger impulse. The experimental results obtained by applying one impulse are shown on Figure 15., Figure 17. and Figure 19. The responses to certain defined velocities are shown on Figure 16., Figure 18. and Figure 20.

The responses are almost instant as shown on Figure 15., Figure 17. and Figure 19. The small deviation is the result of mechanical characteristics of stepper motor. The noise seen in those figures is caused by imperfection of the encoder and its disc. The response signal collected from encoder is raised by 0.5 (V) because of the encoder's saturation, which represents a nonlinear acting.

Figure 16., Figure 18. and Figure 20. show various measurements of desired and actual velocity. It can be seen that the actual velocity follows the desired one. At certain moments there are deviations of actual velocity. These deviations are caused by restrictions of the

used encoder. The shape of those deviations is the result of linear approximation of the characteristic.

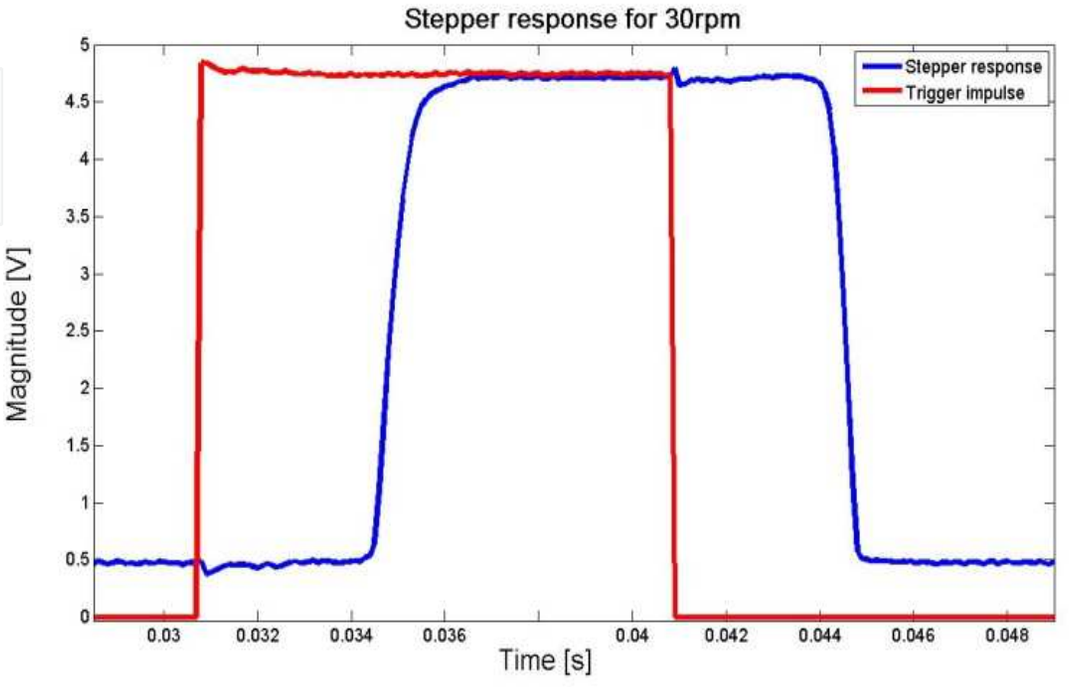


Figure 15. Response to one step for velocity of 30 RPM

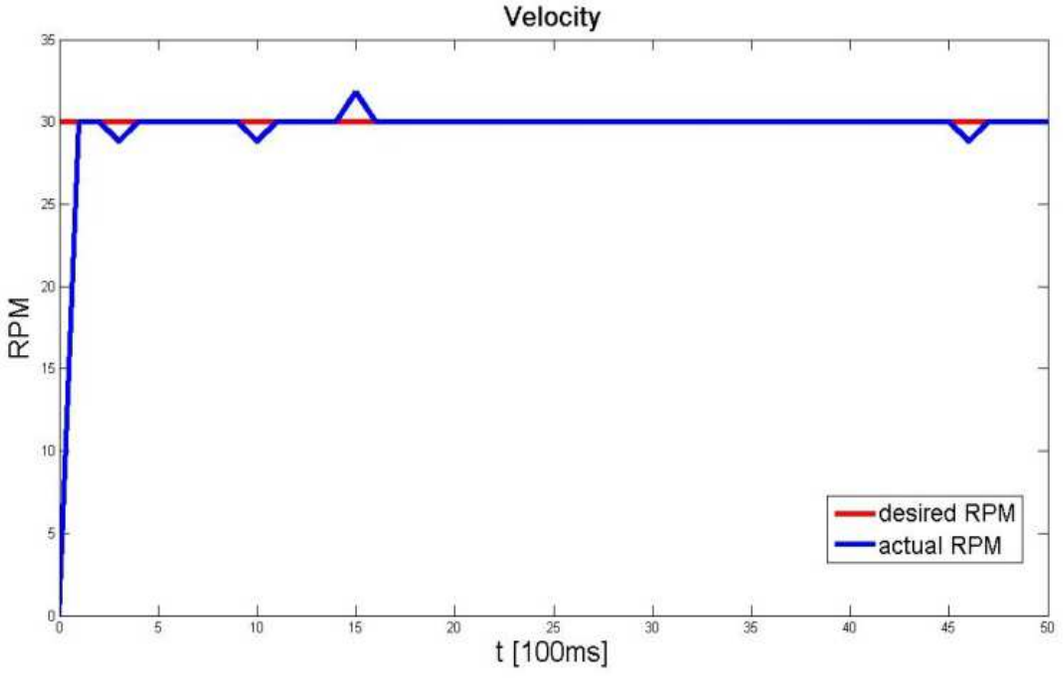


Figure 16. Desired and actual velocity of 30 RPM

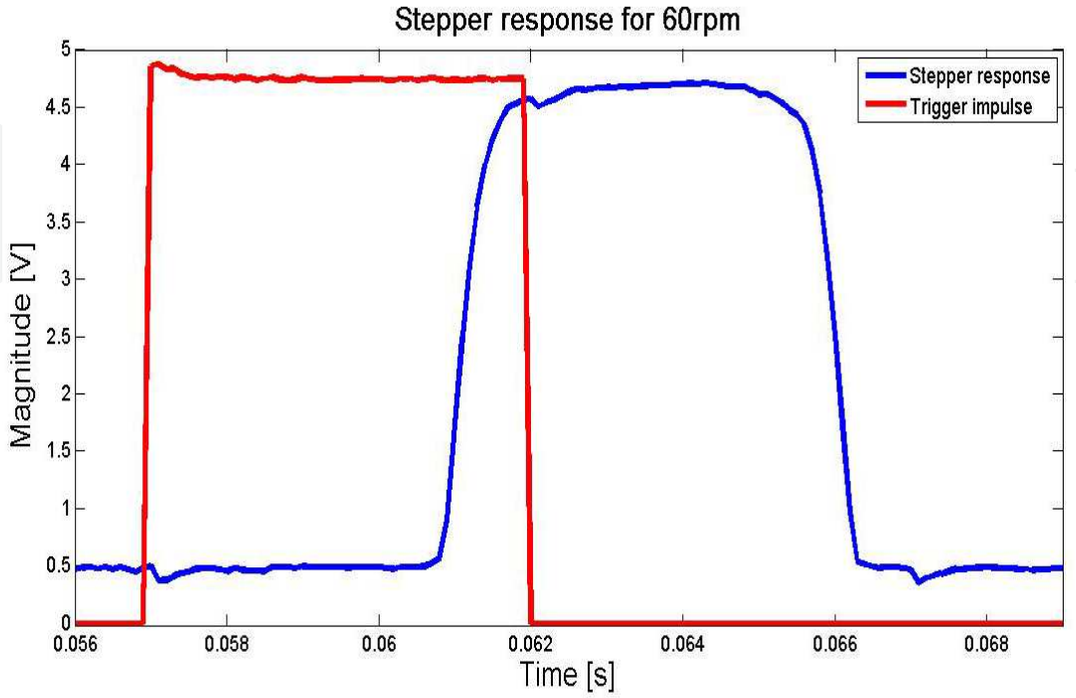


Figure 17. Response to one step for velocity of 60 RPM

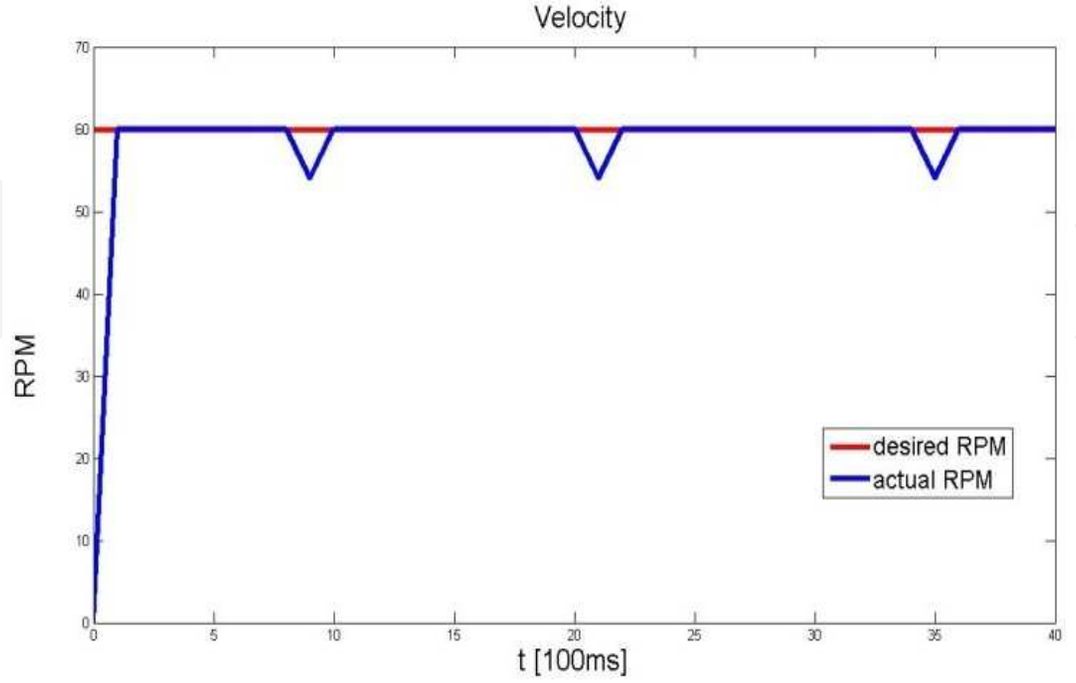


Figure 18. Desired and actual velocity of 60 RPM

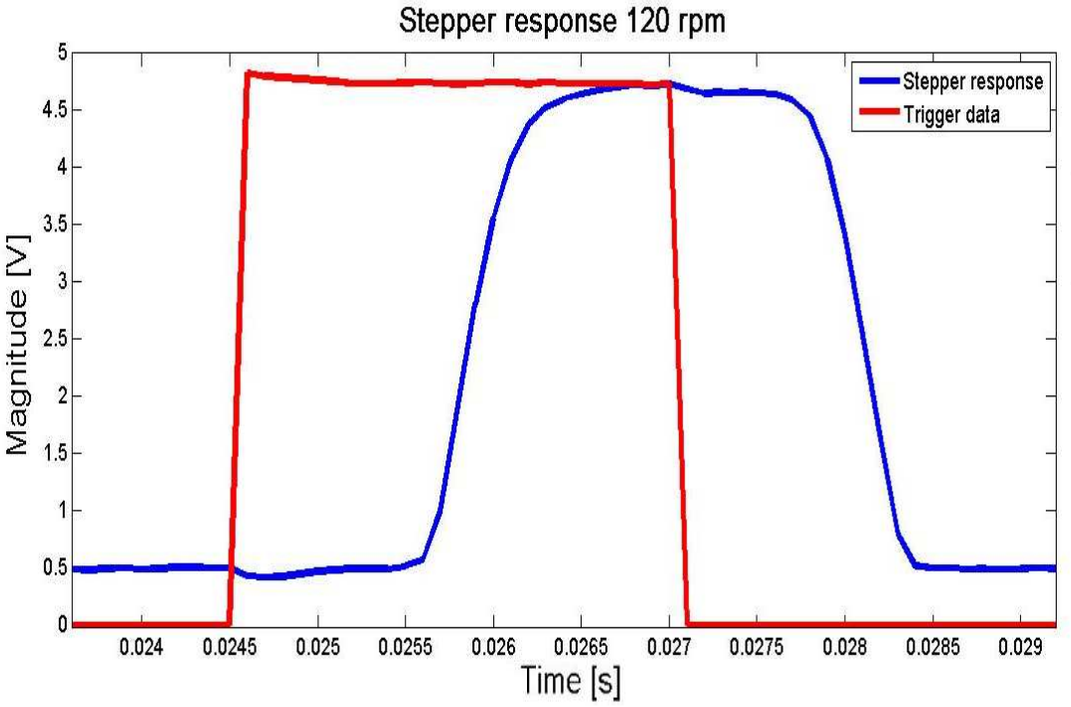


Figure 19. Response to one step for velocity of 120 RPM

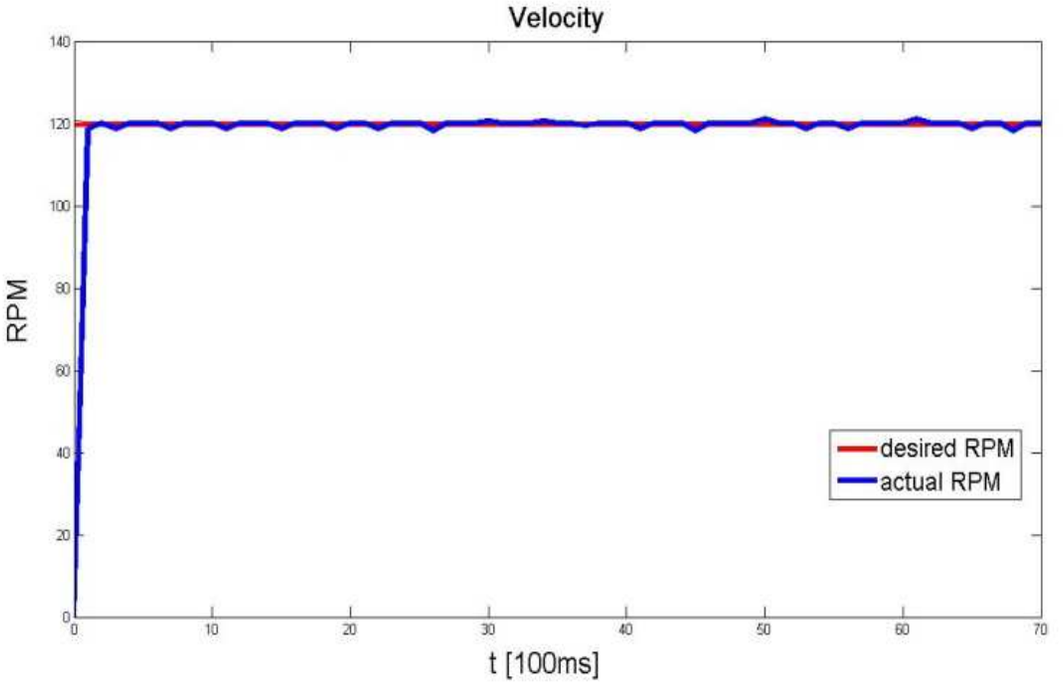


Figure 20. Desired and actual velocity of 120 RPM

6. Conclusion

This paper presented the design of a remotely controlled stepper motor and robotic arm via web server. Operating algorithms and GUI were realized for both systems. Through the GUI for the stepper motor user can operate the motor in two modes: velocity and positional. The feedback received from the encoder is sent through the established connection from server to the client. Experimental results demonstrate the effectiveness of the remotely controlled stepper motor. Using the GUI for the robotic arm, user can operate each joint of robotic arm separately. The feedback received from the camera is sent through the established connection from server to the client. Experimental results are not shown in this paper, because as stated before, this model of robotic arm does not possess encoders. That is the reason why camera was used as visual feedback. The system operates in real time and visual feedback provides us information about current state of robotic arm. System is based on microcontroller and its development is not expensive, unlike the systems which are based on other technologies i.e. PLC. These systems are used in environments which are dangerous for humans.

Author details

Vedran Vajnberger, Semir Silajdžić and Nedim Osmić
Faculty of Electrical Engineering,
Department of Automatic Control and Electronics, Sarajevo, Bosnia and Herzegovina

7. References

[1] T. B. Sheridan (1993) Space teleoperation through time delay review and prognosis, *IEEE Transaction on Robotics and Automation*, vol. 9, pp. 592-606.
[2] C. Sayers (1996) Remote Control Robotics, New York: Springer Verlag.
[3] Velagic, J., Coralic, M. and Hebibovic, M. (2004) The Remote Control of Robot Manipulator for Precise Time-Limited Complex Path Tracking, Proceedings of the IEEE International Conference on Mechatronics and Robotics (MechRob2004), Volume 2, September 13-15, Aachen, Germany, pp. 841-846
[4] D. Lee, and M.W. Spong (2006) Passive Bilateral Teleoperation with Constant Time Delay, *IEEE Transactions on Robotics and Automation*, vol. 22, no.2, pp. 269-281.
[5] Vladimir Lucan, Petr Simacek, Jari Seppälä, Hannu Koivisto, "Bluetooth and Wireless LAN Applicability for Real-time Control"
[6] Xin Liu , Yongtian Wang , Yue Liu , Dongdong Weng, Xiaoming Hu (2009) A Remote Control System Based on Real-Time Image Processing, 2009 Fifth International Conference on Image and Graphics
[7] Chui Yew Leong and Abdul Rahman Ramli, Intelligent Systems and Robotics Laboratory (ISRL), Institute Of Advanced Technology, Universiti Putra Malaysia. 43400 Serdang, Selangor. "Development of a real-time embedded remote triggering and monitoring system with SC12"

[8] G.Srinivasarao, S.Sao, "Security system based on stepper motor control using microcontroller"
[9] Ahmet Altintas, Department of Electrical Education Dumlupinar University, Faculty of Technical Education, 43500 Simav, Kütahya, Turkey, "A graphical user interface for programming stepper motors used at different kinds of applications", Mathematical and Computational Applications, Vol. 14, No. 2
[10] Betin, F. Pinchon, D. Capolino, (2000) "Fuzzy logic applied to speed control of a stepping motor drive", G.-A. Dept. of Electr. Eng., Univ. of Picardie Jules Verne, Cuffies , , Industrial Electronics, IEEE, Jun 2000.
[11] Jan B.A. Habraken, Kora de Bruin, Morgan Shehata, Jan Booij, Roel Bennink, Berthe L.F. van Eck Smit and Ellinor Busemann Sokole, "Evaluation of High-Resolution Pinhole SPECT Using a Small Rotating Animal", Departments of Nuclear Medicine, Radiology, and Medical Technological Development, Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands, , Basic Science Investigations.
[12] Fang Liu, Zhenlin Hu, Lei Qiu, Chun Hui, Chao Li, Pei Zhong and Junping Zhang (2010) "Boosting high-intensity focused ultrasound-induced anti-tumor immunity using a sparse-scan strategy that can more effectively promote dendritic cell maturation", *Journal of Translational Medicine*
[13] Salcudean, S. E., Ku, S., and Bell, G., 1997, "Performance Measurement in Scaled Teleoperation for Microsurgery," Proceedings of the First Joint Conference in Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery (CVRMed-MRCA '97), Grenoble, France, pp. 789–798.
[14] Jasmin Velagić, Nedim Osmić, Semir Silajdžić, Tarik Terzimehić and Vedran Vajnberger (2010) Remote Control of Stepper Motor via Web Server, Conference on Control and Fault – Tolerant Systems (SysTol'10), Nice, France, 10/2010.
[15] Vedran Vajnberger, Tarik Terzimehić, Semir Silajdžić and Nedim Osmić (2011) Remote Control of Robot Arm with Five DOF, International symposium for information and communication technologies, electronics and microelectronics – MIPRO 2011, Opatija, Croatia
[16] Liptak, Bela G. (2005). *Instrument Engineers' Handbook: Process Control and Optimization*. CRC Press. pp. 2464.
[17] Yidong Wang, Kaiguo Yan, Guozi Sun, Peihuang Lou, "Serial Communication in DNC Information Systems", The CIMS Center of NUAA, Nanjing 210016, China, Published by Moxa Technologies)
[18] Yidong Wang, Kaiguo Yan, Guozi Sun, Peihuang Lou, "Serial Communication in DNC Information Systems", The CIMS Center of NUAA, Nanjing 210016, China, Published by Moxa Technologies)
[19] Peter Rydesaeter, TCP/UDP/IP toolbox 2.0.6. user guide, March 2001.
[20] htttp://www.ccsinfo.com, User guidance
[21] MPLAB® IDE USER'S GUIDE, 2005 Microchip Technology Inc.,
[22] Personal Edition VNC Server 4.4 User Guide, 2008.,