# Teleoperation and Remote Monitoring of a ScorBot ER-4U Robotic Arm in an Academic Environment

**Franklin Salazar L., Jorge Buele, Homero J. Velasteguí, Angel Soria, Edith Elena Tubón Núñez, Clara Sánchez Benítez, Gabriela Orejuela T.**

***Abstract*: The present research work develops a system that uses a programming method implemented for tele-operation of a ScorBot ER-4U. Its objective is remote control by a human operator, adding the possibility of creating routines from the remote site. This indicates a transformation in processes of production or control within industry and academic environments. To achieve the telecommunication purpose, two interfaces named client/server were developed in the LabVIEW software, which will process information through a TCP/IP communication protocol. Through the client interface, the operator gives specific orders for the robot to execute them. While the server interface allows the visualization of the instructions that arrive from the client and determines if there are changes in the requested positions to be executed in the robot. It transmits a frame with the information to the ScorBot through Serial communication. In ScorBase, a routine that reads the serial port data detecting errors and executing the movement requested by the TCP client is executed permanently. The drawback of the project is the delays in the execution of the movements, produced by characteristics of the environment of handling and programming of the robot. Although they are not lapses greater than a second, generate a vision of discontinuity in movements determined***

***Index Terms*: Process control, robotic arm, TCPIP, teleoperation.**

## I. INTRODUCTION

During the evolution of industrial machines, mankind fascination for devices capable of performing activities and movements similar to that of human beings has been a topic of constant research and development [1], [2]. As a result, modern robotics puts more emphasis in providing the user with greater innovation to facilitate their daily activities. To achieve the task of replicating human characteristics and activities, when operating in industrial environments [3], [4], as is the case with robot arms [5]. Those devices used to be categorized as elements of science fiction, currently the reality is that this technology is widely used in industrial processes, with complex infrastructures, a topic that needs to be well known and dominated by those who work in the industrial field [6]-[8].

An important aspect in the use of robots is the "Laws of robotics". Those are expressed in the following way:

- A robot cannot harm a human being or with its inaction allow a human being to suffer damage [9].
- A robot must obey and receive orders from a human being, unless such orders enter into conflict with the first law.
- A robot must protect its own existence, as long as such protection does not enter into conflict with the first and second law.

On the other hand, the International Robotic Federation (IFR[1]) defines an Industrial Robot as follows: "An Industrial handling Robot is understood as an automatic handling machine, reprogrammable and multifunctional with three or more axes that can position and orient material, pieces, tools or special devices for the execution of different jobs in the different stages of industrial production, either in a fixed or moving position" [10], [11].

In [12] a proposal is presented where the control of a robotic arm ScorBot ER-III is carried out implementing an FPGA board, which was connected to stereo cameras that capture live images of the robotic arm's current position. Also, with mathematical calculations the target position of the robotic arm can be computed, and SCPI commands can be transmitted via RS-232 interface to the robotic arm to actually move it in the desired position.

In [13] the simulation is performed in MATLAB/Simulink, and implementation of different types of neuronal structures to evaluate their performance as a controller, using a robotic arm for that purpose. In [14] the kinematic analysis of the ScorBot ER-4U robot arm using a Multi-Layered Feed-Forward (MLFF) Neural Network. The algorithm is tested on hardware (ScorBot ER-4U) and reliable results were obtained. The modeling and simulations were done using MATLAB 8.0 software.

**Revised Manuscript Received on December 22, 2018.**

**Franklin Salazar L.**, Facultad de Ingeniería en Sistemas, Electrónica e Industrial, Universidad Técnica de Ambato, Ambato, Ecuador.

**Jorge Buele**, Facultad de Ingeniería en Sistemas, Electrónica e Industrial, Universidad Técnica de Ambato, Ambato, Ecuador.

**Homero J. Velasteguí**, Facultad de Ingeniería en Sistemas, Electrónica e Industrial, Universidad Técnica de Ambato, Ambato, Ecuador.

**Angel Soria**, Electrical and Computer Engineering, Purdue University, West Lafayette, United States of America.

**Edith Elena Tubón Núñez**, Facultad de Ingeniería en Sistemas, Electrónica e Industrial, Universidad Técnica de Ambato, Ambato, Ecuador.

**Clara Sánchez Benítez**, Facultad de Ingeniería en Sistemas, Electrónica e Industrial, Universidad Técnica de Ambato, Ambato, Ecuador.

**Gabriela Orejuela T.**, Facultad de Mecánica, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador

This document seeks to contribute with an academic material for the training of the ScorBot ER-4U robot. To do so, it is required to generate a system capable of manipulating, programming, and remote monitoring of a five degrees of freedom robotic arm, that communicates via Ethernet or WiFi interface. After performing the respective tests, the effectiveness in the speed of execution for teleoperation of the robotic arm is determined.
This project is divided into 4 sections, including the introduction as section 1, section 2 presents the methodology, section 3 discusses about the obtained results. Finally, the conclusions and future work are presented in section 4.

## II. METHODOLOGY

### A. *Teleoperation*

Is a group of technologies that includes the operation or remote management of a device by a human operator. Teleoperation is the action that a user or operator makes to manipulate a device from different distances [15], [16]. A fundamental element when talking about teleoperation is human supervision and control. Meaning that one or more human beings are programming, and are continuously receiving information from a computer that interconnects the sensors and actuators of a control process or in an industrial environment.

### B. *Characteristics of the Robot Arm ScorBot ER-4U*

The system was designed for the ScorBot ER-4U robot. It has a vertical articulated design, similar to an arm, which uses 6 articulations for its movement, as shown in Fig 1 [17], [18]. The robotic support has a limitation of 310° in its base rotation, 158° in its shoulder rotation, 260° in the rotation of its hand and 260° in the wrist. It also has a radio operation of 610 mm that allows short distance operations or processes [14]. The system has owner control software that directly limits teleoperation. The precision data of each point on the average displaceable axis is 2.07m, but it should be considered that the linear axis moves at a distance of 3.0m.

This robot presents a versatile and reliable system for educational use and can be mounted on a table, pedestal, linear sliding base, mobile platform, etc. Thanks to its speed, it is considered suitable both for independent operations and for integrated use of automatic work cells and FMS applications, such as: robotic welding, artificial optical systems and automatic parts handling. That is, it brings together properties that imitate and greatly extend the capabilities of natural organisms, such as human beings.

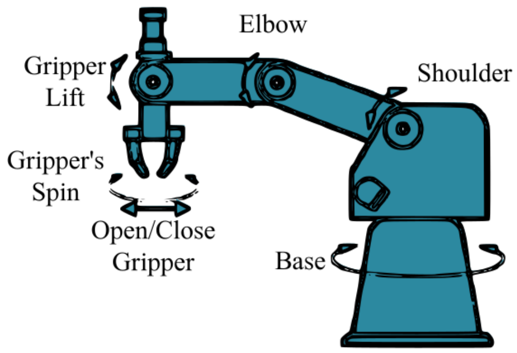


**Fig. 1. Axes work of robot ScorBot ER-4U.**

For the operation of the arm and its accessories, a USB-Controller is used to communicate with a computer where the ScorBase software (proprietary software for the ScorBot control) is installed, with the respective drivers required for the model to be used. The USB-Controller has a power supply of 24 VDC for the motors of the robot arm, two additional connections that form axes 7 and 8, used for the motors of the conveyor belt. In addition, the controller has digital and analog input and output ports for the connection of sensors, switches, actuators, transmitters, and other devices. The programming of the robot in a generic way is developed through the RoboCell software, which consists of a software package that integrates four components:

- ScorBase, a robotic control software package that provides a simple tool for the programming and operation of robots and their accessories.
- A graphic display module that provides 3D simulation of the robot and other devices in a virtual work environment.
- CellSetup to create and modify virtual work environments.
- A demonstration program for 3D simulation.

### C. *Operation and Remote Robot Programming*

The remote operation of the robot works on the base of the block diagram shown in Fig. 2. The operator manipulates or schedules routines from independent languages. The communication between the programing environment and the robot goes through two different communication protocols: RS232 and TCP/IP. The position control of the robot arm five axes was done with a communication interface that uses RS-232 protocol. Where ScorBase software and LabVIEW programming environment exchange the execution information of movements. A serial port emulator is executed between the two programming interfaces, in which two virtual serial ports are created, allowing the exchange of RS-232 packets. Using a control script programmed in proprietary software for the control of the first port and the management of LabVIEW on the second.

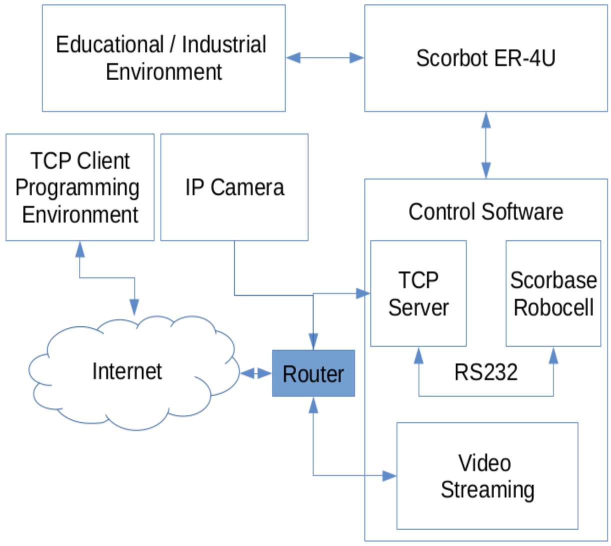


**Fig. 2. System block diagram.**


*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*

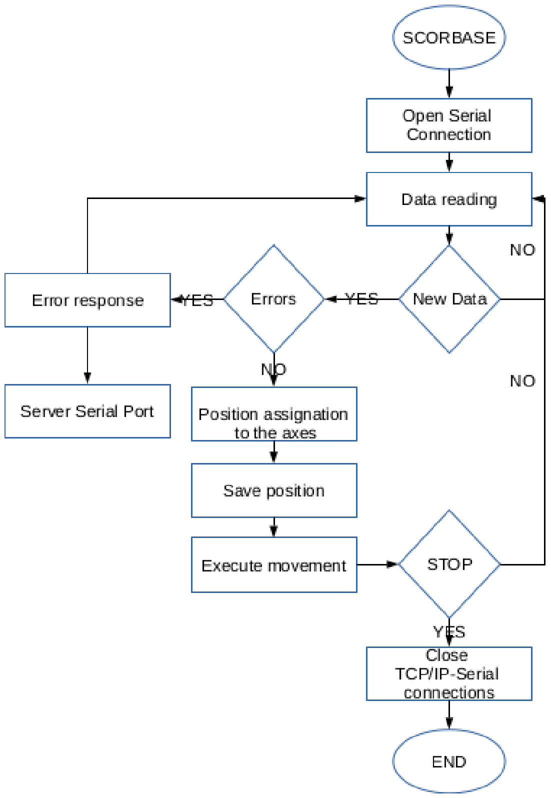


**Fig. 3. Programming diagram of the routine executed in ScoreBase.**

### *D. ScorBase Routine*

The software programmed is shown in Fig. 3, takes the data from LabVIEW through the serial port, which specifies the position to which each axe must be addressed. At the moment of receiving data, the errors are checked during the transmission. If the received information does not detect errors, the position is memorized in a variable and subsequently the displacement of the axes is executed. In case of detecting errors, a request is sent for retransmission of the data to the server through the serial port. If the control script does not receive any information, the robot does not perform any action. If the script detects communication errors, it does not execute the movements until it receives the correct data frame. Stopping the execution of the ScorBase program closes the serial port used.

### *E. ScorBase Routine Client-Server Management*

The application that allows remote control of the robots was developed in LabVIEW Software, and it has two main parts. A TCP server that runs in the same computer that controls the robot routine using ScorBase, in which a communication socket is opened on port 8006 to receive the requests of the clients through the TCP protocol. The second part is the TCP client that generates the information to request the server to manage the communication, to locate the robot in a certain position.

In the server software shown in Fig. 4, TCP/IP communication socket is configured, as well as a Serial communication port. The first receives continuously the information coming from the TCP/IP clients in a data frame, containing the information of the specific location to where the client requests the robot to execute its movements. The program continuously reads the data from the TCP/IP port. When a request for a change in the position of the robot axes is detected, the server passes the received frame to ScorBase through the serial port. Once the information is sent the system condition is to expect an answer, with this method the system determines the existence of errors in the communication. For the case of errors, the serial port retransmits the last data frame received by the TCP/IP socket until a communication without errors is detected. When the Server program is stopped, the Serial port and TCP/IP socket are closed. The data frame used to transmit the information of the client to the server is the same used to send the information from the server to software proprietary, as it is shown in Fig. 5.

The data frame is composed of three different sections: start indicator, axes position data and end indicator. The length of the frame is 52 bytes representing the data of the locations of each axis of the ScorBot, and 4 bytes that correspond to the identification of the start and end of the frame. The data transmission starts with the characters 0xFF 0xFF, corresponding to the start indicator. Continuously the ID of each axis is sent with the value of its location, using 2 bytes to identify the axes and 6 bytes to send the value of the location. The transmission of the frame ends with bytes 0x00 0x00 indicating at the end of the frame. The start and end indicators are used to detect the correct data reception management and error handling by the serial port and ScorBase. In addition to serial communication, it also uses detection errors by parity par.

The data section of the positions of the robot axes is composed of two fundamental parts. First, the axis identifier is sent and subsequently the position of the value is transmitted. The desired format consists of 6 characters, taking this form a frame of constant magnitude. The values of the axis's positions are transmitted consecutively in each data frame, between the start and end indicator.

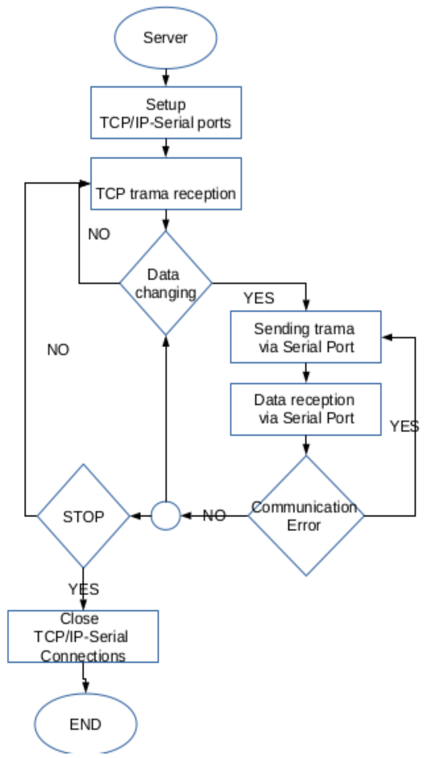


**Fig. 4. Programming diagram of the application of the Server.**


Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

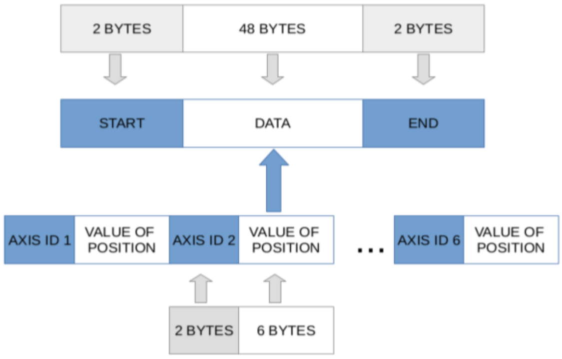


**Fig. 5. TCP/IP- Serial data frame for the transmission of the position of the robot axes.**

The software programmed in the TCP/IP client performs different functions such as supervision of the process executed in the robot. To achieve this, video streaming from an IP camera is used to save positions, and read the saved positions, thus requesting the movements of the robot axes based on that. The diagram of the executed program is presented in Fig. 6. The application contains a graphical interface that allows the user to specify the position of each axes of the robot. For this purpose, when initiating the client execution, the communication with the server is established. Variables are created for the robot axes, which can be manipulated from an internal memory of the LabVIEW software. This allows the creation of routines from the storage of data in the hard disk of the client computer or from the graphic interface. The frame is created from the variables of the axes and transmitted to the server continuously.

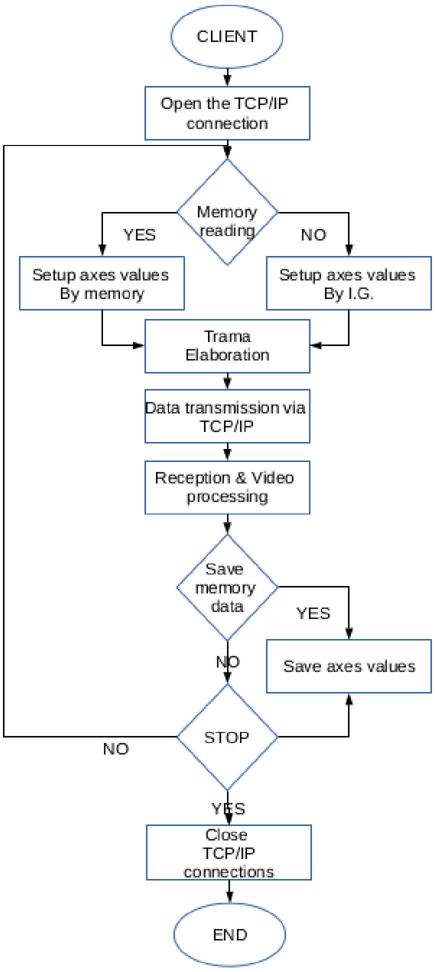


**Fig. 6. Diagram of the client's application TCP/IP.**

In parallel, a streaming of video is executed to visualize the situation and movements of the robot in real time. The video is taken and transmitted through an IP camera, and is processed in LabVIEW as a video from a local camera using the IPC software. It is an application executed on the host TCP/IP client that allows the use of IP cameras as web local cameras. For the case of this research topic any IP camera can be added, or by default a camera with a controller that can be accessed from a communication network with the TCP / IP protocol. When the application stops the TCP/IP client closes the communication socket. The TCP communication is managed by the code shown in Fig. 7. For this case the controller establishes the IP address of the server. The program allows a duplex communication between the server and the client, executing a subVI 1 that controls the reception if data, and then a subVI 2 to send the data frame.

The transmission subVI sends a first parameter with the information of the length of the frame, and afterwards it sends the data. In the reception part first, the number of bytes of the data frame are obtained to then recover the received data.

The frame in the client is established in a function developed in the block diagram, as shown in Fig. 8. Each axis of the robot is related to a variable handled from the front panel. Which can be written directly or by reading a cluster stored in the memory of the computer. Within the function we have a subVI 2 that limits the values of the variables to the extreme magnitudes that each axis reaches.

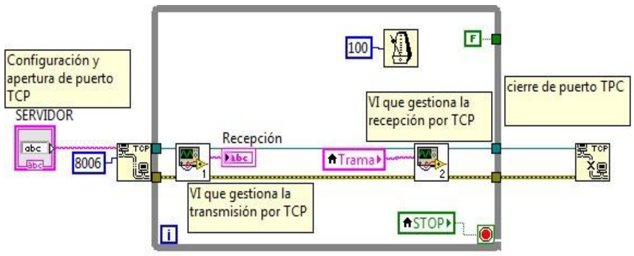


**Fig. 7. Management of TCP client communication.**

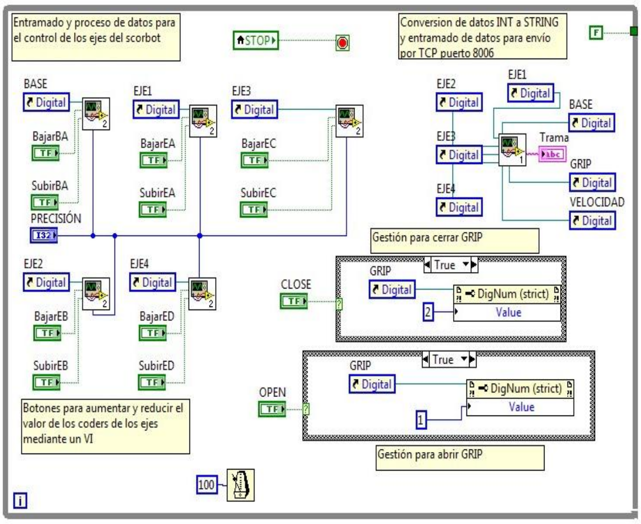


**Fig. 8. Arming the data frame of the positions of the axes.**



*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*

In each cycle the data is taken from the screen and through the subVI 1, the information is transformed from an integer type to a string to create a fixed size frame. Using a start, end sequence and 6 characters in a specific location for the value of each axis. The memory positions identified by a whole number save the values of the robot axes in a cluster, LabVIEW data logger is used for this purpose, thus storing the data in an extension file *log. Each stored location occupies a total of 48 bytes plus an identifier of the saved position (32-bit unsigned integer variable), with the possibility of storing up to 232 different positions. The data is stored in a file in the LabVIEW server program directory, storing the information on the hard drive disk of the computer. Which is accessed in any moment, to load the saved positions. In this way it is possible to program direct control routines in the graphic language interface provided by LabVIEW from the client side.

## III. RESULTS

Programming a simple routine created the designed system. LabVIEW recorded the sequence of positions to follow the ScorBot ER-4U robot on the client computer. Then loaded at different times and transmitted to the server to execute the robot's movement in a cyclic way. The routine used was the displacement of a plastic of bottle, as is shown in Fig. 9. Where in memory position (0), a reference position for LabVIEW, known as home, was established, from which the movements of each of the robot axes will be executed. In the following memories, actions will be established for the axes corresponding to the capture of the bottle in the respective position for the memory (1). Execution of the movements of the clamp (2), elevation in the vertical axis (3), displacement in the axis axial (4), decrease in the vertical axial (5) and opening of the clam (6).

The memories were saved with the position of the axes in each numeral value, moved the axes manually from LabVIEW. It being possible to create a routine from the client, requesting the server to execute consecutively the location of the axes in positions (1), (2), (3), (4), (5), (6). Returning finally to position (1) to repeat the process in a cyclical way. This sequence of the programming in LabVIEW and executed in ScorBase, is shown in Fig. 10. The execution of commands sent from LabVIEW to the robot has a delay of about one second, in the manual movements to establish and save the positions of the axes in the memories. Delay is perceptible. However, in the execution of routines the delay is despicable.

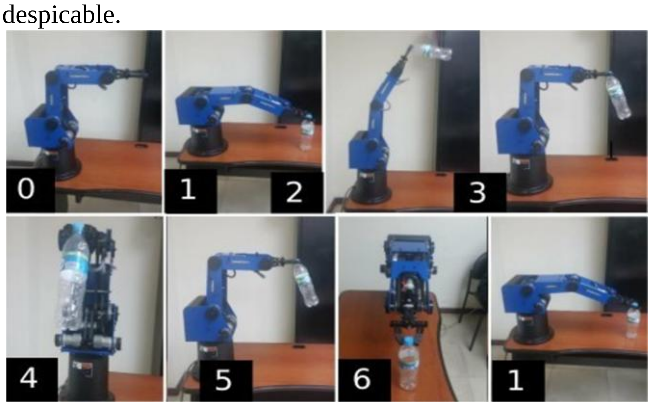


**Fig. 9. Programming sequence executed from the client.**

| | |
|---|---|
| 1 | Load script file: SERIALPORT.VBS |
| 2 | Llama Subrutina SCRIPT.INIT |
| 3 | Llama Subrutina SCRIPT.OPENPORT |
| 4 | INICIO: |
| 5 | Espere 1 (10cent. de segundo) |
| 6 | Poner Variable IS_READ = SCRIPT.READ |
| 7 | Poner Variable SHOULDER = SCRIPT.EA |
| 8 | Poner Variable ELBOW = SCRIPT.EB |
| 9 | Poner Variable PITCH = SCRIPT.EC |
| 10 | Poner Variable ROLL = SCRIPT.ED |
| 11 | Poner Variable BASE = SCRIPT.BASE |
| 12 | Poner Variable GRIPER = SCRIPT.GR |
| 13 | Poner Variable VEL = 1 |
| 14 | Si BASE == 15000 salta a INICIO |
| 15 | Teach Position 1 by Joints. Coordinates: BASE SHOULDER ELBOW PITCH ROLL |
| 16 | Ir a la Posicion 1 velocid. 1 |
| 17 | Si GRIPER == 1 salta a ABRIR |
| 18 | Si GRIPER == 0 salta a INICIO |
| 19 | Si GRIPER == 2 salta a CERRAR |
| 20 | ABRIR: |
| 21 | Abrir Pinza |
| 22 | Salta a INICIO |
| 23 | CERRAR: |
| 24 | Cerrar Pinza |
| 25 | Salta a INICIO |
| 26 | Llama Subrutina SCRIPT.CLOSEPORT |
| 27 | Mordaza 10 ( mm ) |

**Fig. 10. Program running on ScorBase.**

Being able to perform complete sequences of movements with the robotic arm, allows its use in various activities that involve danger or difficulty for operators. It is used in high-impact industrial environments, as it can support a weight of up to 2.5 kg, which is greater than other types of robotic arms. As "Black Hawk", which lacks safety elements and the ability to place tools in the end effector, due to the limitations in its design. The service robot Jaume-2 is also presented. that presents an architecture to perform tasks only in academic and experimental environments. In order to evaluate the area of application of the prototype designed, some previously conducted investigations are cited below, which may motivate future work. As can be seen in [18] where the design and construction of a dynamic and versatile robot is presented. This could be used in the deactivation of bombs, examination of radioactive environments and combat oriented military/police operations, among others. Another application is presented in [19], where a suitable robotic arm with a moving platform and infrared sensors accomplishes trajectories by dodging objects. Its field of application is the treatment of explosive materials and for military purposes.

## IV. CONCLUSIONS AND FUTURE WORK

The implementation of this type of technology allows the human being the possibility of improving their quality of life in the industrial field, using this type of prototypes in activities that involve danger. Through the Client and Server made in the LabVIEW software, there is a TCP/IP communication with a permissible delay for the programming and modification of robot routines remotely, ensuring the physical well-being of the robot's manipulative entities; especially in high risk environments. The programming done from the LabVIEW software for the ScorBot ER-4U is of high resolution, in reference to the positioning of the axis of the same one. Presents location control of up to one step on each axis.

This software is limited to the initial and event control executions that can only be executed and reset from the ScorBase proprietary software.

The perception of depth in remote programming is limited by the characteristics and location of the camera. Therefore, several of the IP cameras were distributed in different plans, to obtain a better perception of the depth in the different axes of coordinates. The execution of movements of manual form that is executed from LabVIEW can be done at different speeds. However, at high speeds the resolution of the position is lower. Therefore, for long movements high speeds must be used, and then low speeds to reach the precise location. Due to the limitation characterized by the use of proprietary software as communication interface. It is necessary to use a mobile platform for the displacement of the arm in case of collisions in high risk environments.

As future work, authors propose the use of new technologies to reduce the latency produced in the control process. In addition, connect a mobile platform that gives greater autonomy to the robot and thus be able to expand the current application area. In the same way we will look for ways to strengthen the communication protocol, to provide greater security to the end user. In this way, it can be implemented in industrial environments and perform the corresponding experimental tests.

## ACKNOWLEDGMENT

To the dean Pilar Urrutia and authorities of FISEI of Universidad Técnica de Ambato (UTA) and Facultad de Mecánica of Escuela Superior Politécnica de Chimborazo (ESPOCH), for supporting this work and future research.

## REFERENCES


1. M. Shakir, M. Hammood and K. Muttar, "Literature review of security issues in saas for public cloud computing: a meta-analysis," *International Journal of Engineering & Technology*, vol. 7, no. 3, pp. 1161-1171, 2017.
2. S. M. Zanoli, L. Barboni, F. Cocchioni and C. Pepe, "Advanced process control aimed at energy efficiency improvement in process industries," *2018 IEEE International Conference on Industrial Technology (ICIT)*, pp. 57-62, 2018.
3. J. J. Ng, "Statistical process control chart as a project management tool," *IEEE Engineering Management Review*, vol. 46, no. 2, pp. 26-28, 2018.
4. A. Șipoș, "The alcoholic fermentation process temperature automatic control," *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, pp. 1-6, 2018.
5. L. Jhang, C. Santiago and C. Chiu, "Multi-sensor based glove control of an industrial mobile robot arm," *2017 International Automatic Control Conference (CACS)*, pp. 1-6, 2017.
6. Y. Xie, J. Pan, J. Yan and J. Li, "Design of the fast speed two-arm robot in limited space," *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 652-656, 2017.
7. H. H. Kim, S.O. Park, J. H. Kyung, H. M. Do and M. C. Lee, "A study for estimating reaction force of robot arm by using PDSPO," *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 258-262, 2017.
8. D. He and Y. Guo, "Finite element analysis of humanoid robot arm," *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 772-776, 2016.
9. M. E. Shoshiashvili and I. S. Shoshiashvili, "Principles of construction of control devices for mechatronic pipe-lay complexes," *2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, pp. 1-4, 2016.
10. Y. Tang, X. Xing, H. R. Karimi, L. Kocarev and J. Kurths, "Tracking Control of Networked Multi-Agent Systems Under New Characterizations of Impulses and Its Applications in Robotic Systems," *IEEE Transactions on Industrial* Electronics, vol. 63, no.*2*, pp. 1299-1307, 2016.
11. Z. Yan, S. Guo, L. Shi, Y. Wang, G. Li and W. Peng, "Study on slave side of interventional surgery robotic system focused on the feed-back force detection," *2016 IEEE International Conference on Mechatronics and Automation*, pp. 420-425, 2016.
12. R. Szabo and A. Gontean, "SCORBOT-ER III robotic arm control with FPGA using image processing with the possibility to use as them as sun trackers," *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 563-566, 2017.
13. M. S. Kazemi and M. J. Dominguez, "Simulation and evaluation of neuro-controllers applied in a SCORBOT," *2016 IEEE International Conference on Automatica (ICA-ACCA)*, pp. 1-9, 2016.
14. R. R. Kumar and P. Chand, "Inverse kinematics solution for trajectory tracking using artificial neural networks for SCORBOT ER-4u," *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pp. 364-369, 2015.
15. X. Xu, B. Cizmeci, C. Schuwerk and E. Steinbach, "Model-Mediated Teleoperation: Toward Stable and Transparent Teleoperation Systems," *IEEE Access*, vol. 4, pp. 425-449, 2016.
16. P. Malysz and S. Sirouspour, "Cooperative teleoperation control with projective force mappings," *2010 IEEE Haptics Symposium*, pp. 301-308, 2010.
17. A. S. Anthony and A. P. Pallewatta, "Four Legged Walking Robot with Smart Gravitational Stabilization," *Kelaniya International Conference on Advances in Computing and Technology (KICACT - 2017)*, pp. 29, 2017.
18. D. N. S. R. Kumar and D. Kumar D, "VNC server based robot for military applications," *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, pp. 1292-1295, 2017.