# Industrial Control Robot Based on Augmented Reality and IoT Protocol

William Montalvo[1], Pablo Bonilla-Vasconez[2], Santiago Altamirano[2], Carlos A. Garcia[2], and Marcelo V. Garcia[2,3](✉)

[1] Universidad Politecnica Salesiana, UPS, 170146 Quito, Ecuador
wmontalvo@ups.edu.ec
[2] Universidad Tecnica de Ambato, UTA, 180103 Ambato, Ecuador
{pbonilla8622,santiagomaltamirano,ca.garcia,mv.garcia}@uta.edu.ec
[3] University of Basque Country, UPV/EHU, 48013 Bilbao, Spain
mgarcia294@ehu.eus

**Abstract.** The use of robotic systems into production processes has resulted in the reduction of manufacturing costs and an improvement in product quality, However, as technologies advance, complex and high-cost control systems are required, hence the need to create low-cost and efficient control systems, with low complexity when operating them. For these reasons, the research presents a control system for the Scorbot ER 4U manipulator arm using a low-cost embedded board and augmented reality platform that brings to the users a real experience with virtual add-ons, thus facilitating the training and management of robotic equipment. A virtual control proposed in Unity 3D allowing the application to be much more interactive and very easy for the user. For the validation of the system, operating tests were carried out, resulting in movements with great precision and in a very simple way without having greater complexity when indicating positions to which the manipulator must reach.

**Keywords:** Industry 4.0 · Cyber-Physical Production Systems (CPS) · Robotics · Embedded systems · Augmented Reality · Unity 3d

## 1 Introduction

The applications that are developed in Virtual Reality (VR) or Augmented Reality (AR) can be oriented to the processes of teaching-learning [4,5], in the academic part and to the training - qualification in the industrial scope, these processes previously mentioned can be applied individually or in collaborative works between users, for which it is considered: (i) environments with a user in which tasks that can be performed individually are considered such as assembly of mechanical parts, doors [3], spot welding, precision welding [17,18], electronic control units [2], electric motors, among others.

In recent years, robotics has aimed to find solutions to the needs of all industrial fields. Technological evolution influences developing more research topics in


L. T. De Paolis and P. Bourdot (Eds.): AVR 2020, LNCS 12243, pp. 345–363, 2020.
https://doi.org/10.1007/978-3-030-58468-9_25

the robotics community. This evolution has been dominated by human needs, mainly in works that are high risk to man, hence the idea of replacing them with industrial robots [7].

In the current context of the technological world, robotic systems have had a significant advance in cost and implementation, with the emergence of low-cost Single Board Cards (SBC) that can be developed efficient control systems for robotic automation to really low prices compared to the systems that are handled in the robotics and automation market [1]. However, despite being a major breakthrough, these new technologies have only taken place in the major world powers, there is limited knowledge of them in less industrialized countries.

The main goal of this research is the development of augmented reality (AR) platform for the control and handling of a robotic manipulator such as the Scorbot ER 4U, using low-cost microcontrollers or so-called low-cost embedded cards. The first step is performed the cinematic analysis of the robotic arm, and thus define the method of control of the manipulator, which will be designed in an AR environment using the Unity 3D engine, together with the Meta 2 glasses increases for interaction with the user, finally, the users can send messages to the embedded card using the Message Queuing Telemetry Transport (MQTT) protocol, and in turn, the low-cost controller card control the robotic arm, finally, these messages will be validated checking that the robot's movement is the desired one.

The design of the document is as follows, Sect. 2 shows some related works that have been used as a starting point for this research. Section 3 describes the state of art that introduce the elements used in the methodology of this research. Section 4 illustrates a case study of AR for industrial training. The proposed solution for the case study is presented in Sect. 5. Finally, some conclusions and future work are established in Sect. 8.

## 2 Related Works

This section highlights the importance of using systems based on industry 4.0, for the control of robotic joints from virtual environments using low-cost cards, this is what details McLin et al. research [9], where an approach is proposed to establish bidirectional interactions between robots in the physical and virtual worlds, allowing changes that occur in a physical robot to be automatically reflected by its virtual counterpart, and vice versa, based on this research can be established what are the requirements to create a virtual environment that reflects the physical robot.

Similarly in the research developed by Yoshiaki et al. [11], where he explains about a new software architecture for a Human-Robot Interaction simulator (HRI) in virtual environments, a cloud-based virtual reality platform, called "SIGVerse", which reduces costs to develop real robots and real-world interaction experiments with the help of the virtual. This information underpins the understanding of how virtual platforms are generated to generate movements in physical devices achieving synchronization between the two.

Similar research shows that efficient control systems can be designed for manipulators such as SCORBOT, so, it is exposed by Mota Muoz in [12], which explains the implementation of a Proportional Derivative Gravitational (PD+ G) controller for the second and third articulation of the SCORBOT ER-4u robot. For which the dynamic parameters of the robot were calculated through laboratory measurements and calculations in a CAD model. The dynamic model was got by Euler Lagrange method, based on this the authors of this paper can perform the study of the behavior of the robot and how to set the parameters that should be considered to create a controller of the Scorbot manipulator.

According to recent advances in hardware technology he describes that new interfaces can be created with Augmented Reality (AR) for the control and manipulation of robotic systems, as Courtney Hutton states in [6] which refers to the new technologies that are present in activities for defining task-oriented robots, the research explains how to start developing an AR system in Unity 3D$^{\text{TM}}$ software to control robots, under the simple "point where to move" system.

On the other hand, AR can also be used for data visualization as explained by Florian Leutert et al. in [8] who exposes on AR as a form of Mixed Reality that is a technique that improves the natural environment of the user with additional useful virtual information to help the user with their tasks; details the correct way in how information should be integrated and oriented within the environment, which must be dynamic, interactive and real-time in the user's workspace. Under this consideration, efficient reality environments can be designed, where there is a virtual environment with augmented reality support for user interaction.

With the support of the researches above addressed, it becomes feasible the development of the proposed topic, since, supported by the topics developed by the different authors can be developed a control method for the manipulator SCORBOT-ER 4U, from an environment virtual with augmented reality support which is interactive and intuitive and using low-cost devices as hardware means for physical communication.

## 3 State of Technology

This section will explain all the concepts that will be applied for the development of the research proposal.

### 3.1 Augmented Reality (AR)

It is a technology that allows to add layers of visual information about the real world around us, using devices such as mobile phones, technological glasses, tablets, computers, camera, among others. This helps us generate experiences that provide relevant knowledge about our environment, and we also receive that information in real time. Among its main features are the ease of interacting in the real world with elements of the virtual world, thus mixing the best of both and the inclusion of information in real time.

AR has brought with it a great contribution to various fields such as: medicine, education, entertainment and in recent years various applications have been developed for the industrial field, brings great advantages to the various fields between the which highlight the optimization of times on daily tasks of many workers, implement new channels to show information to people and ease for training staff [3].

### 3.2 Unity 3D™

Unity 3D™, created by Unity Technologies [16], is a powerful cross-platform 3D engine and easy-to-use development environment. For 3D interfaces, Unity enables specification of texture compression, mipmaps, and resolution settings for each platform supported by the game engine, and provides support for embossed mapping, reflection mapping, parallaeude mapping, occlusion screen space (SSAO), dynamic shadows using shadow maps, texture rendering, and full-screen post-processing effects.

It's cross-platform, supporting C-programming and Visual Studio integration. Unity also offers JavaScript as an assembly language and MonoDevelop as an IDE for those who want an alternative to Visual Studio. It comes with powerful animation tools that simplify creating 3D scenes or creating 2D animations from scratch.

### 3.3 SCORBOT ER 4U

The SCORBOT-ER 4u [15] robot is an industrial robot, a general purpose programmable machine that has certain anthropomorphic characteristics. This arm along with the robot's ability to be programmed makes it ideal for a variety of production tasks. The SCORBOT-ER 4u robot is a versatile and reliable system for educational use. The robot arm can be mounted on a table, a pedestal or a linear sliding base.

The specifications are as follows: It is a 5-axis vertical articulated robot, the maximum operating radius is 610 mm, at the end it has a clamping clamp whose maximum aperture is 75 mm. The actuators are servo motors of 12 v in direct current whose maximum weight capacity is 15 oz and a speed of 600 mmsec also has optical feedback sensors on each axis. The transmission is by gear and timing belt. The weight of the robot is 10.8 gk

### 3.4 META 2 Development Kit

Technical features of Meta 2 Development [10] include a transparent AR headset that displays digital content or holograms with a 90-degree field of view and a high-resolution 2550x1440 display. Designed for comfort, the headphones allow the user to unobstructed everything under the eyebrows so they can easily make eye contact with other people or wear lenses comfortably. The kit also includes a 720p front camera, a four-speaker audio system near the ear, brightness and volume control, and a sensor array for hand interactions and positional tracking. There is also a 9-foot video cable (HDMI 1.4b), data and power.

### 3.5 Raspberry PI (RPi) 3B+ Card

Among its main features we can mention the speed of its processor that is 1.4 GHz which makes its data processing capacity very high, indispensable to control a process or system in real time. By possessing physical and wireless USB, Ethernet, Bluetooth and wifi communication ports allow the possibility to develop a number of applications under most communication protocols (MQTT, TCPIP, AMQP. Works with operating systems such as Windows and Linux derived systems (Raspbian) [13].

Raspbian is a free Debian-based operating system optimized for Raspberry Pi hardware. The RPi card is used for the creation of the Scorbot manipulator control system, possesses digital input/output pins (GPIOS), necessary to operate the servo motors and read the feedback devices (encoders) of the Scorbot arm, and thus give movement to the robot joints, RPI will be the controller of the robotic arm in the proposed project.

### 3.6 Message Queuing Telemetry Transport (MQTT) Protocol

MQTT is an application layer protocol that works above the TCP/IP transport layer protocol for transferring messages. It is a suitable protocol for devices with limited resources and was invented by Dr. Andy Standford-Clark and Arlen Nipper in 1999. It's a lightweight, open, simple, and easy-to-use protocol, making it ideal for communication in resource-constrained environments like IoT.

The protocol works under the publisher-subscriber mode through the creation of a broker, to filter messages, these are arranged in topics organized hierarchically. Figure 1 illustrates the working method of mqtt. The communication parameters are as follows: (i) Broker Address, (ii) Define topic name and (iii) Define the communication port.

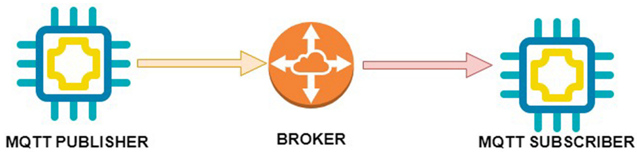


**Fig. 1.** MQTT Architecture

## 4 Case Study

The hardware architecture is the conceptual design and the fundamental operational structure of the system. It is the physical support where the software resides. The hardware architecture consisting of a Meta 2 glasses [14] as the AR device. Meta 2 is a stand-alone device that has an own graphic processor that runs the processes natively on its hardware. These glasses provide real-time tracking, stable and accurate.

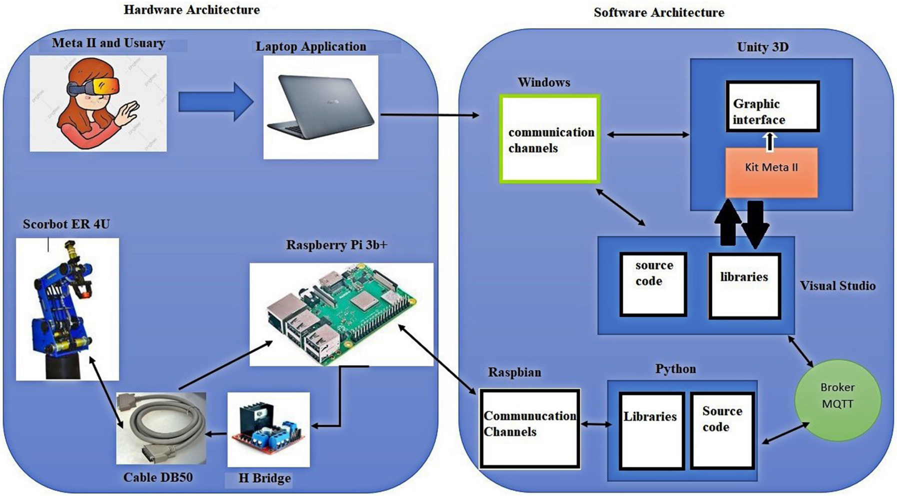


**Fig. 2.** Software and Hardware Architecture

Meta 2 glasses are connected to the computer through communication cables that will allow interaction with the developed AR system. As shown in Fig. 2, once a precise location of the space is achieved and maintained, the user just by putting on the viewer will already have the reach to the virtual objects and the virtual training sequences that will be overlaid on the real environment through the computer that contains the application and because the device maps the environment and tracking the user's actual location, the user can use defined hand movements to interact with the 3D images.

The main goal of this research is the development of a control system for the Scorbot ER 4U robotic arm, from an augmented reality environment platform created in Unity 3D and with the use of Meta II augmented reality goggles, a low-cost arm controller is designed used the RPI card which interacts with the robotic manipulator through the GPIO inputoutput pins. The MQTT communication protocol is used for communication between the virtual environment and the controller.

The software architecture provides a general reference necessary to guide the construction of the augmented reality system for user training, allowing all application objectives and restrictions to be covered.

This way, as shown in Fig. 2, the system design has an interface level at which the previously selected development engine will design the graphical interface and the set of instructions with which the user can interact. It also has a development level that will allow the encoding of the top-level instructions, and where the user interface and general development libraries of the application will be added, and then compiled them, all this will be done using Unity 3D software and Visual Studio. Finally, the base level will allow managing the communication channels of the Hardware architecture with the system.

# 5 Implementation of the Solution Proposal

This stage presents the design of the system architecture both at the hardware and software level, describes all the components that will be part of the control system, for the description Unified Modelling Language (UML) diagrams, are used.

## 5.1 Hardware Architecture

The hardware architecture consists of the design of a physical communication network that allows the interaction of all hardware components of the system supported in the software. Figure 3 has a diagram of components of the hardware that are part of the proposed control system.

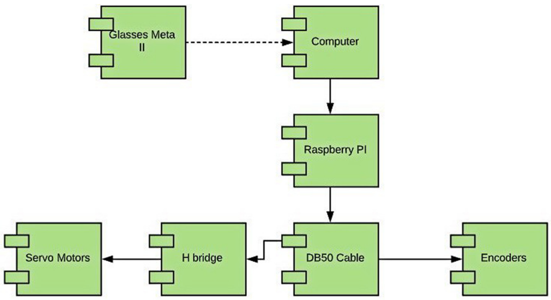


**Fig. 3.** Diagram of Physical Components Control System Scorbot ER 4U

For user interaction with the system the Meta 2 augmented reality goggles is used, these smart glasses offer the ability to track in real-time, stable and accurate, provide a mapping of the environment and follow the actual location of the user. The glasses interact with the virtual world through the computer by using communication channels such as the HDMI port and the USB3.0 port. In this way the user when placing the virtual reality glasses will be able to visualize the Scorbot robot in 3D, having the possibility to change the position of the effector of the robot indicating where it should go simply by moving its hand.

RPI interacts with the Scorbot arm to drive the servo motors and receive the signal coming from the encoders, for such interaction the boards use the GPIO input/output pins. The embedded card is also connected to the virtual world via the computer, through the ethernet or wifi communication ports and supported by the MQTT data transmission and reception protocol receives all the signals coming from the PC where the META 2 goggles are connected.

Between RPI, the Scorbot arm actuators and sensors have two additional peripherals for communication. The H-bridges chip needed to raise the current and voltage of the signal sent by the embedded card, and the DB50 cable that connects directly with the elements of the robot, it should also be mentioned that to have sufficient supply of current and voltage it is essential to use an auxiliary source.

Figure 4 provides the hardware architecture between RPI and the Scorbot manipulator arm.

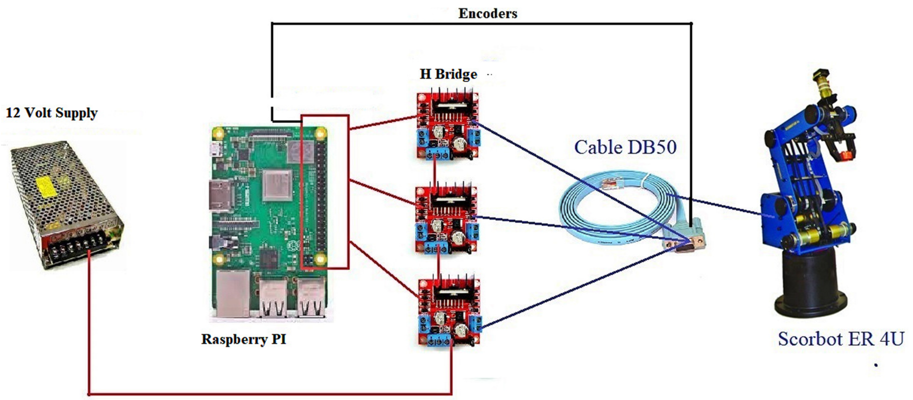


**Fig. 4.** Raspberry pi and Scorbot hardware architecture proposed

## 6 Software Architecture

It consists of two levels of assembly, the first, is developed in Visual Studio and allows the coding of the instructions in the language C# for the control and movement of the 3D model of Scorbot which was previously designed, instructions to communicate under the MQTT protocol and code to interact with the META 2 glasses, in addition, at this level also add all the libraries necessary for the user interface, control of objects in the workspace and communications.

The second level of assembly is developed in the Thonny Raspbian IDE which is a code manager for Python language, in this assembler are encoded the instructions for sending and receiving signals through the GPIO pins, thus managing to interact with the robot Scorbot. Also, there are a C# code to establish communication through the MQTT broker, and finally, all the libraries necessary for the use of raspberry GPIO pins and MQTT libraries are added.

### 6.1 Virtual Environment

The virtual environment component diagram shows the different objects that the application has (see Fig. 5).

(i) **Scorbot ER 4U model**: Inside the graphical interface the user can display the 3D model of Scorbot, which through the augmented reality device can be interacted with. (ii) **Home button:** A button labeled "Home Position" can be displayed, which when pressed sends the robot to its home position.

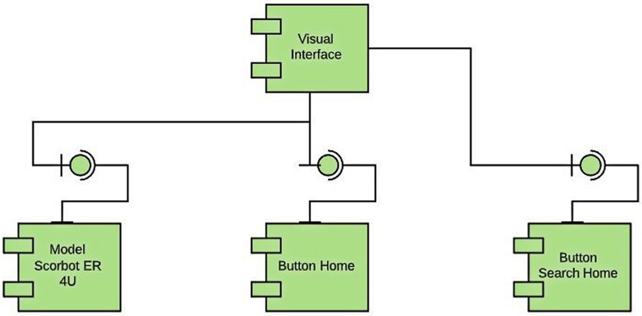


**Fig. 5.** Virtual interface component diagram

(iii) **Search Home button:** Inside the interface is a button labeled "Search Home" that when pressed sends a message to the microcontroller for the physical robot to start its search for the home position.

### 6.2 Human-Robot Interface (HRI) Design

It bases its operation on the manipulation of 3D objects, which are contained within a scene, in order to manipulate the objects that require script coding C# Sharp language. These scripts implement a class with attributes and methods. For the development of the proposed control system, scripts have been developed for both the control and handling of the robot and for communication with the embedded card. This interface is programming used 5 different classes which are detailed below:

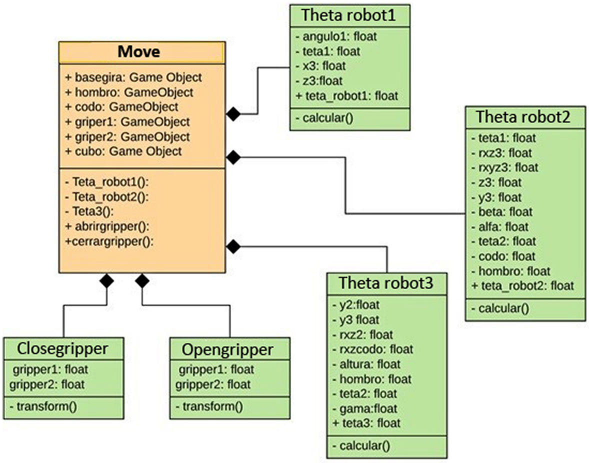


**Fig. 6.** Scorbot Movement Class Diagram

**Scorbot ER 4U Movement Design.** Figure 6 shows the classes used for Scorbot arm control, each class has their attributes and methods.Furthermore, there is a composition relationship between the classes. As the "movement" class can be observed predominates over the others i.e. without the "movement" class the others cannot continue to function independently. The description of each class is presented in the Table 1:

**Table 1.** Description of the Scorbot Er 4U movement classes

| Class name | Description |
|---|---|
| Move | Is the main class for scorbot arm control, this class is able to access each element of type Game Object that is in the scene of the application. Each Game Object element provides attributes to subclasses, these attributes are required to perform the calculation of the values that the arm joints should take based on the kinematics |
| Theta_robot1 | It contains the encoding of the calculations needed to find the value that joint 1 should take |
| Theta_robot2 | Calculates the value that joint 2 should take, all its attributes are of type float, and like the previous class its attributes are taken from the main class |
| Theta_robot3 | A class that calculates the value that joint 3 should take, based on the attributes of type float provided by the main class |
| Opengripper | Class that contains the encoding to open the gripper. Its attributes are of type float provided by the main class |
| Closegripper | A class that contains the encoding to close the gripper. Its attributes are of type float provided by the main class |

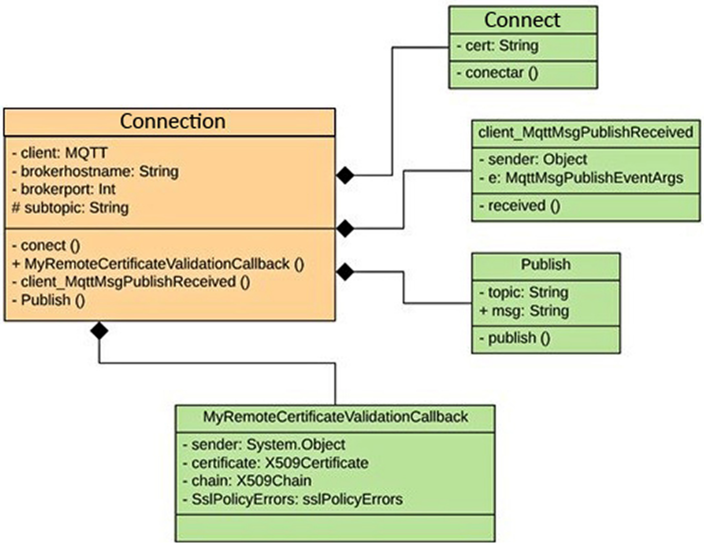


**Fig. 7.** Scorbot Movement Class Diagram

**Communication Implementation in Unity 3D Software.** Figure 7 shows the classes that are used for MQTT communication protocol implementation in Unity 3D software. The MQTT communication stack used is Mosquitto Broker. Each class has their respective attributes and methods, it is important to note that there is a composition relationship between the "communication" class and the other classes, i.e. the classes they cannot function independently without the "communication" class. Each class is described below in the Table 2:

**Main Scene Class Diagram.** Figure 8 shows the classes that are used for the main scene, each class has their respective attributes and methods, it is important to note that there is a composition relationship between the "main" class and the other classes, the classes do not can operate independently without the "main" class. Each class is described below in the Table 3:

### 6.3 Raspberry Software Interface Design

Figure 9 shows the classes that are used for the control of General Purpose Input/Outputs (GPIO) pins in RPI, each class has their respective attributes and methods, it is important to note that there is a composition relationship between the "control" class and the other classes. In Table 4 is presented the description of the all classes used:

**Table 2.** Description of MQTT communication classes for Unity 3D

| Class name | Description |
|---|---|
| Connection | It is the main class for establishing MQTT communication, makes the connection to a Broker and subscribes to a topic |
| Connect | The class which sets the necessary parameters to communicate using MQTT protocol. Requires attributes of type string, MQTT, and int, which the main class provides, performs a single action that is to initialize the MQTT protocol communication |
| Client_MqttMsg Publish Received | A Class to receive messages that comes from the topic to which have been connected requires attributes of type object and MQTT |
| Publish | A class that allows messages to be sent to a topic, requires string-type attributes |
| MyRemote Certificate Validation Callback | A class that provides security and encryption parameters for messages that are exchanged within a topic in MQTT communications, requires private attributes of the MQTT class |

**Table 3.** Description of the main classes

| Class name | Description |
|---|---|
| Main Class | The main class of the module, because it is the one that allows me to access each "GameObject" element that represents a scene within the application |
| Interaccion_Trex | Class that allows the user to manipulate the Scorbot 3D arm |
| Finish | Class that allows the user to close the application |
| Home | Class that allows the message to be sent to the robot's home position |
| Position Home | Class that allows the message to be sent so that the robot searches for its position at home |

**MQTT Communication Implementation in Raspberry Pi.** Figure 10 depicts the class diagram of the MQTT communication in RPI. This class has attributes of type MQTT like client, and the userdata, also attributes of type String that are what define the Broker's IP address, the name of the topic, and the message that is received. The methods used are to create connection (on_conecct), to receive the message (on_message), and to initialize the client (mqtt.client). The MQTT communication stack used is Mosquitto Client. These are the parameters required to communicate under the MQTT protocol with the Unity environment.

### 6.4 Interface Development in Unity

Within the virtual environment should be positioned the 3D model of Scorbot which is built in the Blender software. With the model already in the workspace, the authors proceed to give physical and kinematic characteristics for the

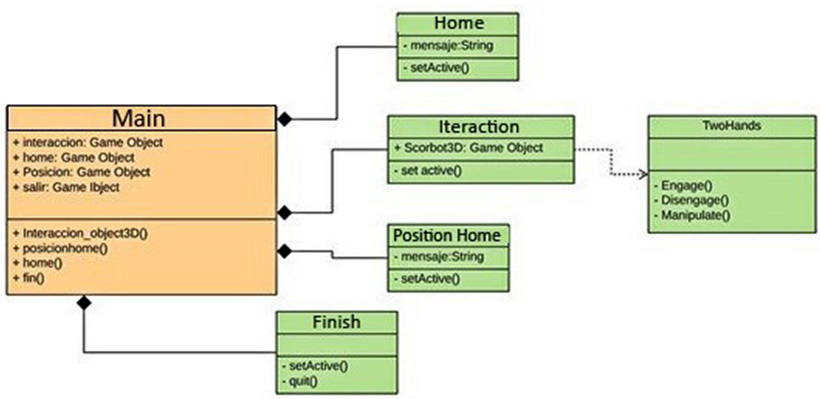


**Fig. 8.** Scorbot Movement Class Diagram

**Table 4.** Description of the control GPIOs classes

| Class name | Description |
|---|---|
| Control | Main class that provides subclasses with all the attributes needed for GPIO pin control |
| Motor Turn Base positive | A class that drives the motor at the base of the Scorbot manipulator in clockwise direction |
| Motor Turn Base negative | A class that drives the Scorbot robot base motor counter to the previous class |
| Motor Turn Elbow positive | A class that drives the Scorbot arm elbow motor in clockwise direction |
| Motor Turn Elbow negative | A class that drives the Scorbot manipulator elbow motor in the opposite direction to the previous class |
| Motor Turn Shoulder positive | A class that drives the Scorbot arm shoulder motor in clockwise direction |
| Motor Turn Shoulder negative | A class that drives the Scorbot manipulator's shoulder motor in the opposite direction to the previous class |
| Motor Turn Gripper positive | A class that drives the gripper engine of the Scorbot manipulator in clockwise direction |
| Motor Turn Gripper negative | A class that drives the scorbot manipulator's gripper motor in the opposite rotation to the previous one |

movement of each part of the robot. In addition, is inserted into the scene a small white cube that is a pointer that indicates where the end of the robot should be placed. Finally, buttons and labels are created that will be inside the scene, a background image has also been added to give the scene a pleasant look according to the situation. Figure 11 shows the design of the virtual environment.

The AR system for robot control is evaluated through several attempts in real-time by some different kind of users such as industrial technicians and students (See Fig. 12). The AR platform gets satisfactory evaluation results. When users tested the application, the developers had the opportunity to ask them a series of open questions as an interview about their perception of the developed system. The users said that they found the application interesting and that they found it innovative and interactive to be able to see the 3D models on the screen and to be able to move them.

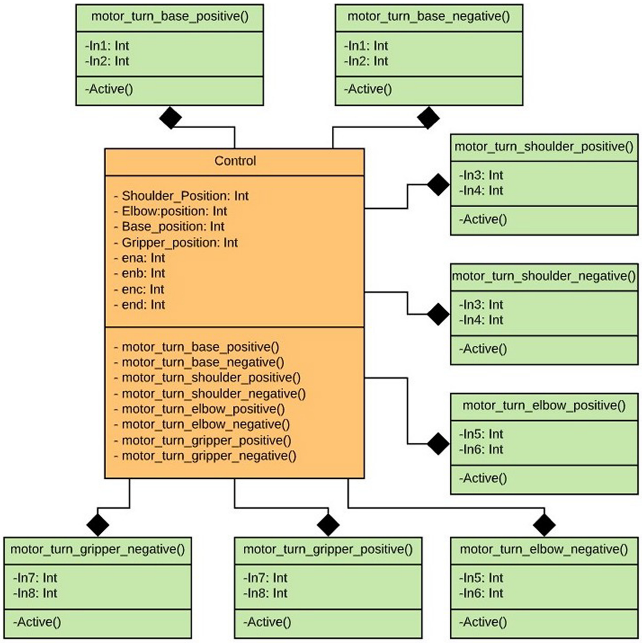


**Fig. 9.** Scorbot Movement Class Diagram

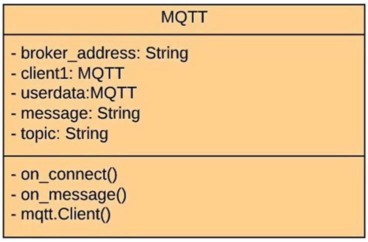


**Fig. 10.** MQTT into Raspberry Pi Class Diagram

## 7 Analysis and Discussion of Results

This section explains different tests that are conducted with the designed AR applications on the local machine connected to the Meta II glasses and RPI for a time of 10 min the following. The results of the different test are detailed below:

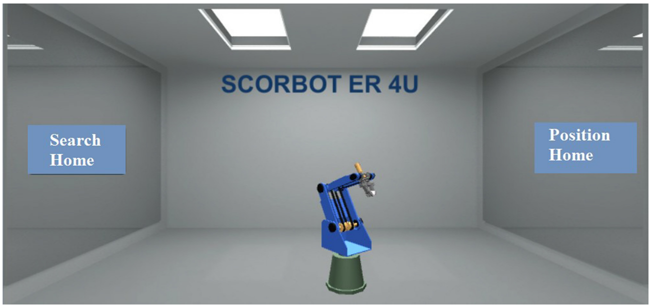


**Fig. 11.** HRI Virtual Environment

**Fig. 12.** HRI Virtual Environment Evaluation by Users

### 7.1 MQTT Network Latency Test

In this test can be observed that network latency remains constant with an average of 45.42 ms on most of the connection for the time evaluated, there are constantly high and low peaks having an upper limit value of 60 ms and a lower limit value of 1 ms (See Fig. 13).

The results show low network latency, which denotes that the established communication protocol is fast enough to perform processes in real-time, being the constant graph in most of the process ensures that the response of the low-cost hardware responds according to the virtual medium by having a synchronization between the virtual robot and the physical robot.

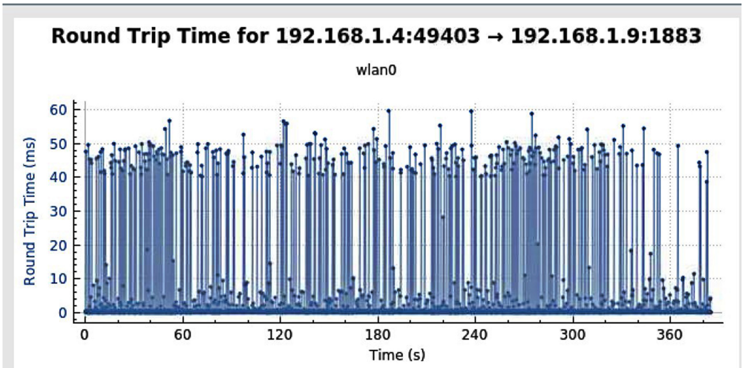


**Fig. 13.** MQTT Network latency

### 7.2 Throughput

The results for MQTT Throughput reflect that there is sufficient bandwidth for data transmission between unity and raspberry pi, having a bandwidth in megabits ensures that there has not been lost data since the data to be transmitted does not exceed kilobits, this ensures that all data sent from the virtual environment arrives complete to the embedded card thus having the desired movement in the Scorbot robot.

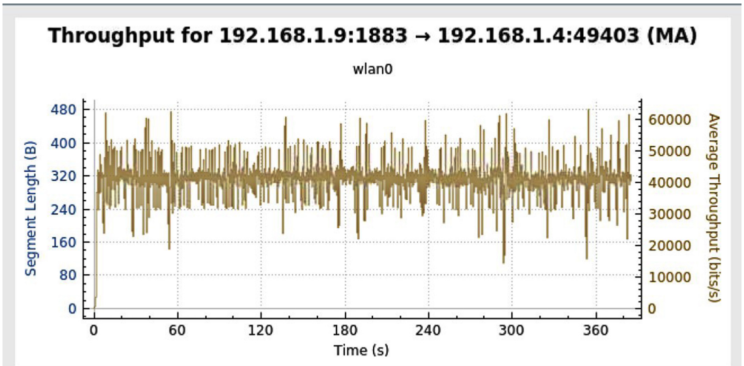


**Fig. 14.** MQTT Network Throughput

### 7.3 Raspberry PI Temperature

For RPI temperature data was taken from both the GPU and the processor, it should be noted that the data recorded was before and after communicating with unity, this to better appreciate how the percentage increases in the work card when the control system is launched.

The results obtained reflect the use of card resources when the control system is put into action because for the initial stage of the graph the RPI is on but without communicating with the unity application, for the second stage communication with unity was launched, and it can be seen that the recessed card minimally increases its temperature indicating that the resources it occupies are very low, and as time goes on it keeps the temperature ensuring that in no Time the card comes to use a lot of resources (See Fig. 15).

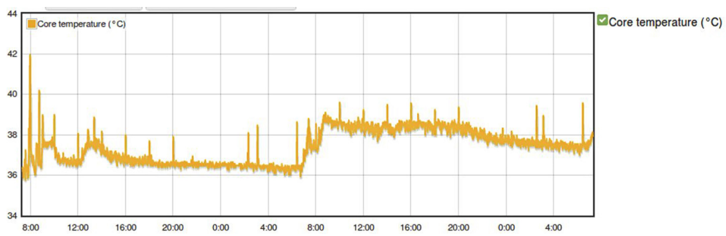


**Fig. 15.** RPI Temperature

# 8 Conclusions and Future Work

The use of the Raspberry Pi card as the controller of the Scorbot ER 4U robotic arm is an effective means of giving movement and control to the manipulator, as it adapts perfectly to the vast majority of communication protocols and thus interacts with different platforms and environments. The processing speed it offers is industrial-type giving open up the development of fairly robust cyber-physical systems and this, in turn, supports the needs of Industry 4.0, finally working with an open-source system and free software which it is important because it gives the possibility of to operate with cross-platform systems and does not require any software developed by private entities making it a great reliable and low-cost alternative for process control anywhere in the industry.

Establishing communication with the controller under MQTT parameters ensure success in the Control System of the Scorbot manipulator, currently, a number of communication protocols are handled that use different parameters and provide certain advantages to when it comes to establishing communications, the Internet of Things has come in with great force in recent years and new communications have been developed that are mainly based on the quality of data and the security with which they are transmitted from one place to another, MQTT is one of these protocols, the use of this communication system makes that the goal achieve successfully because it provides security and quality in the transmission of the message from the virtual environment to the controller card, there is no loss of information and its method of operating under hierarchical topics ensures that only clients connected to the topic receive the information thus avoiding diverting or hiding messages.

From this research can be developed applications for monitoring parameters in real-time control systems since by integrating augmented reality can be displayed on the work screen important information, in addition, you can be able to develop a system for handling loads through manipulative arms in hazardous environments without the need for the person to undergo these conditions as with the use of a low-cost controller and using the communication system implemented you can manipulate the robotic arm remotely and if you add augmented reality in a very easy and intuitive way.

## References


1. Caiza, G. et al.: Industrial shop-floor integration based on AMQP protocol in an IoT environment. In: 2019 IEEE Fourth Ecuador Technical Chapters Meeting (ETCM), pp. 1–6 (2019). https://doi.org/10.1109/ETCM48019.2019.9014858
2. Cardoso, A. et al.: VRCEMIG: A virtual reality system for real time control of electric substations. In: 2013 IEEE Virtual Reality (VR), pp. 165–166 (2013). https://doi.org/10.1109/VR.2013.6549414
3. Dodoo, E.R., Hill, B., Garcia, A., Kohl, A., MacAllister, A., Schlueter, J., Winer, E.: Evaluating commodity hardware and software for virtual reality assembly training. Electronic Imaging **2018**(3), 468-1–468-6 (2018). https://doi.org/10.2352/ISSN.2470-1173.2018.03.ERVR-468
4. Garcia, C.A., Caiza, G., Naranjo, J.E., Ortiz, A., Garcia, M.V.: An approach of training virtual environment for teaching electro-pneumatic systems. IFAC-PapersOnLine **52**(9), 278–284 (2019).https://doi.org/10.1016/j.ifacol.2019.08.221. 12th IFAC Symposium on Advances in Control Education ACE 2019
5. Garcia, C.A., Naranjo, J.E., Ortiz, A., Garcia, M.V.: An approach of virtual reality environment for technicians training in upstream sector. IFAC-PapersOnLine **52**(9), 285–291 (2019).https://doi.org/10.1016/j.ifacol.2019.08.222. 12th IFAC Symposium on Advances in Control Education ACE 2019
6. Hutton, C., Sohre, N., Davis, B., Guy, S., Rosenberg, E.S.: An augmented reality motion planning interface for robotics. In: 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 1313–1314 (2019). https://doi.org/10.1109/VR.2019.8798010
7. Ivorra, E., Ortega, M., Alcañiz, M., Garcia-Aracil, N.: Multimodal computer vision framework for human assistive robotics. In: 2018 Workshop on Metrology for Industry 4.0 and IoT, pp. 1–5 (2018). https://doi.org/10.1109/METROI4.2018.8428330
8. Leutert, F., Herrmann, C., Schilling, K.: A spatial augmented reality system for intuitive display of robotic data. In: 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 179–180 (2013). https://doi.org/10.1109/HRI.2013.6483560
9. McLin, J., Hoang, N., Deneke, W., McDowell, P.: A mirror world-based robot control system. In: 2017 Second International Conference on Information Systems Engineering (ICISE), pp. 74–77 (2017). https://doi.org/10.1109/ICISE.2017.10
10. Meta 2 development kit (2020). https://www.schenker-tech.de/en/meta-2/
11. Mizuchi, Y., Inamura, T.: Cloud-based multimodal human-robot interaction simulator utilizing ROS and unity frameworks. In: 2017 IEEE/SICE International Symposium on System Integration (SII), pp. 948–955 (2017). https://doi.org/10.1109/SII.2017.8279345

12. Munoz, F.M., Martinez de la Piedra, S., Moctezuma, M.V.: PD+G set-point controller for the second and third joint of a ER-4u scorbot robot. In: 2014 International Conference on Mechatronics, Electronics and Automotive Engineering, pp. 173–178 (2014). https://doi.org/10.1109/ICMEAE.2014.36
13. Raspberry pi card (2020). https://www.raspberrypi.org/
14. Schenker-tech: META 2 - Exclusive Augmented Reality Development Kit (2020). https://www.schenker-tech.de//en/meta-2/
15. Scorbot er-4u educational robot (2020). https://www.intelitek.com/robots/scorbot-er-4u/
16. Unity 3d (2020). https://www.unity.com/
17. Wallace, M.W. et al.: Virtual reality GTAW and pipe welding simulator and setup. US Patent 8,851,896, 7 Oct 2014
18. Zboray, D.A. et al.: Virtual reality pipe welding simulator. US Patent 8,915,740, 23 Dec 2014