International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I1P109
Volume 72 Issue 1, 81-92, January 2024

Original Article

Design and Implementation of a Robotic Arm Controlled by a Joystick

Jose Antonio Chinchay-Delgado[1], Bryan Alexander Santos-Buhezo[1], Cristian Castro-Vargas[1*]

[1]*Faculty of Engineering, Universidad Privada del Norte, Lima, Peru.*

**Corresponding Author : cristian.castro@upn.pe*

Received: 21 August 2023 Revised: 22 November 2023 Accepted: 12 December 2023 Published: 07 January 2024

Abstract - *This project consists of implementing a robotic arm in the work environments of companies that handle heavy goods and product assemblies and perform welding or not so common tasks such as handling substances that endanger workers' health. The design of a miniature arm will be implemented using servo motors and a cardboard structure with a Bluetooth mechanism that will allow this to be controlled in different ways. This project aims to optimize processes and protect the integrity of workers.*

Keywords - *Robotic arm, PWM, Control via Bluetooth, HC-05, Servo motor.*

1. Introduction

Currently, various accidents occur in different contexts, such as loading and unloading heavy products, materials handling, and welding, among others. This situation allows us to glimpse an almost prevailing need to implement a significant change [1].

Some companies that have heavy machinery neglect their maintenance. Sometimes, the poor condition of said machinery can cause failures that could end in accidents; unfortunately, many of them become fatal [2]. The reality of accidents related to heavy machinery is broad and covers different modalities. Crucially, many of the most dangerous activities are associated with directly using heavy machinery [3].

The loading processes of some companies are slow, either due to the inexperience of the workers or the low efficiency of the vehicles and loading tools. According to this project, the implementation and use of a robotic arm would increase efficiency in the loading and unloading processes. Likewise, they would minimize the time used for the tasks described above, which would have a positive economic impact on companies that decide to opt for this option [4].

Workers may be at risk by exposing themselves in loading processes. This is usually due to poor security implementation, faulty tools, or calculation errors. In this sense, this project also seeks to safeguard the safety of workers who handle heavy loads daily. Instead, they will control the robotic arms remotely, and there is even room for the implementation of Artificial Intelligence (AI) for process automation [5].

It is important to highlight that implementing robotic arms is not limited to carrying out loading processes. Likewise, its use can be extended to welding or handling dangerous substances in the pharmaceutical field [6].

Given its extensive use for a few hundred years, there are a variety of fields in which it can be applied, from bioengineering to medicine [7]. The most modern models and prototypes usually include wireless technologies since these allow more “comfortable and precise” control [8]. There are even wireless robotic arms that replicate the position of the arm of the person controlling them [9].

From Earth to space, an incredible example is the “European Robotic Arm” project, designed for the International Space Station [10]. Even on other planets, projects related to the design, implementation, and use of robotic arms are carried out. For example, there are NASA's "Rovers", which use vehicular mechanics as well as robotic arms. These types of projects are usually very popular among different researchers who are dedicated to the development and improvement of these technologies [11].

It is pertinent to note that even the structure of a robotic arm can be used for work related to the soil and its physical properties, such as excavation and mining [12].

2. Literature Review

In the investigation of [13] in Japan, harvesting in agricultural fields was studied. These have heavy crops such as watermelon, pumpkin, and melon, among others. This study develops and implements a robotic arm with four degrees of freedom, which automates the collection process.

It also facilitated the implementation of the robotic arm in different industries through the implementation of small modifications.

A low-cost crane (not very complex to build) was designed in a study [14] with a very fine clamp. The purpose of this study is to demonstrate the manipulation of dangerous loads in circumstances of destruction, to achieve precise work taking into account the safety regulations.

A study [15] was carried out, considering technological knowledge in education to awaken students' abilities through modern and traditional teaching methods. Currently, industries, in general, are immersed in the research and development of technologies; that is why a robotic arm with four angles of freedom is developed to satisfy the client's needs.

A robotic arm was designed and implemented in the study [16] to safeguard workers' health. The prototype is in charge of handling toxic or dangerous substances, which can harm the personnel of the companies that work with or handle said substances.

The following study [17] has a problem: the cost of a laboratory with multiple points for the students. For this, a remote-controlled robotic arm is designed. This will allow remote access to a laboratory so that students can experiment properly.

It was concluded that robotic arms and their applications are extremely important in different scenarios. A prototype of a wireless robotic arm is designed and implemented, which helps to optimize the loading, welding, or other processes in which time is reduced and profits are increased, guaranteeing the safety of workers and carrying out processes with precision.

Determine the type of wireless control device, as well as the tools needed for the wireless connection. Design the prototype of the robotic arm and program the wireless communication device together with the electronic circuits of the arm. Simulate the operation of electronic circuits.

3. Materials and Methods

3.1. Selection of Work Area

The project will be implemented in the Sodimac company, as shown in Figure 1, which works in the field of intermediation of construction and home improvement items. This includes heavy products such as wood, iron, cement, bricks, etc. This company owns cargo vehicles, which break down due to constant use (specifically at the bottom of the wheels). To solve this problem, it is desired to implement a prototype robotic arm, which will be designed to carry out many tasks of loading and transporting products [18].

Fig. 1 Sodimac warehouse

Fig. 2 Arduino 1

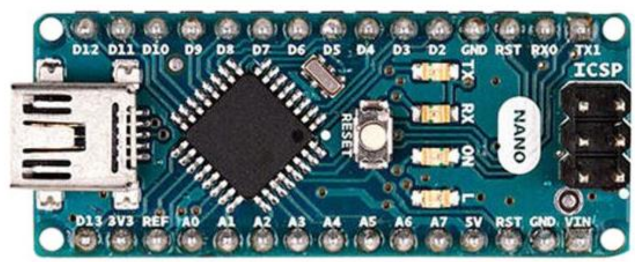

Fig. 3 Arduino nano

3.2. Material Identification and Selection

The components that will be used for the design of the circuit will be determined, which are the following:

Figure 2 shows the main component, which will allow the connections to be made and provide the logic through programming, including libraries such as the servomotor.

Figure 3 and Table 1 show the secondary plugin for the circuit, which will provide the option to use the master HC-05 Bluetooth module.

Table 1. Arduino Nano

Component	Characteristics	
	Current Max.	Voltage
arduino nano	500mA	5-20V

Table 2. Breadboard power supply

Name	HW-131
Volt. Tendon	6.5 V a 12.0 V
Volt. Out 1	3.3 V o 5.0 V (variable)
Volt. but Out 2	3.3 V o 5.0 V (variable)
voltage output	USB 5.0 V
max current out	700 mA
Indicator	LED
Button	Off on

Fig. 4 HW-131 Breadboard power supply

Figure 4 and Table 2 show the components that allow the voltage to be delivered and regulate it, which will be used in the circuit (mainly servomotors). Likewise, to achieve a "smooth and continuous" movement, the type of motors to be used must be considered and implementing new programming codes [21].

Figure 5 and Table 3 show the type of servomotors implemented in the robotic arm to provide mobility.

Table 3. SG90 Servomotor

Name	Servomotor SG90
Sheet. (s)	0.1
Torque (kg/cm)	2.5 kg/cm
Weight (g)	14.7 g
Volt	4.8 V a 6.0 V

Fig. 5 Servo motor SG90

Table 4. MG90 Servomotor

Name	Servomotor MG90
Well	0.1s /60 degrees (4.8V)
Weight (g)	13.4 g
Volt.	4.8 V a 6.0 V

Fig. 6 MG90 Servomotor

This component, which can be seen in Figure 6 and Table 4 is a servomotor that will allow the movement of the robotic arm. Unlike the previous one, the gears are metallic.

Figure 7 and Table 5 show the components that will allow communication via Bluetooth, either in slave (receiver) or master (transmitter) mode.

Table 5. Modulo bluetooth HC-05

Name	Module HC-05
USB	v1.1 y v2.0
Freq.	2.4 GHz
Can. of trans.	Approx. 4dBm
Transmission	2.1 Mbps/160.0 kbps 1.0 Mbps/1.0 Mbps
Sensitivity	Approx. -84dBm
Feeding	3.3 V and 50 mA
Temp.	-5°C a 45°C

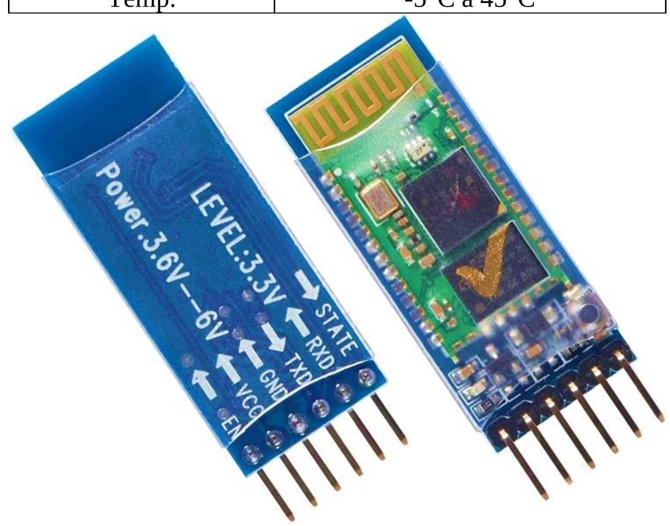

Fig. 7 Bluetooth module HC-05

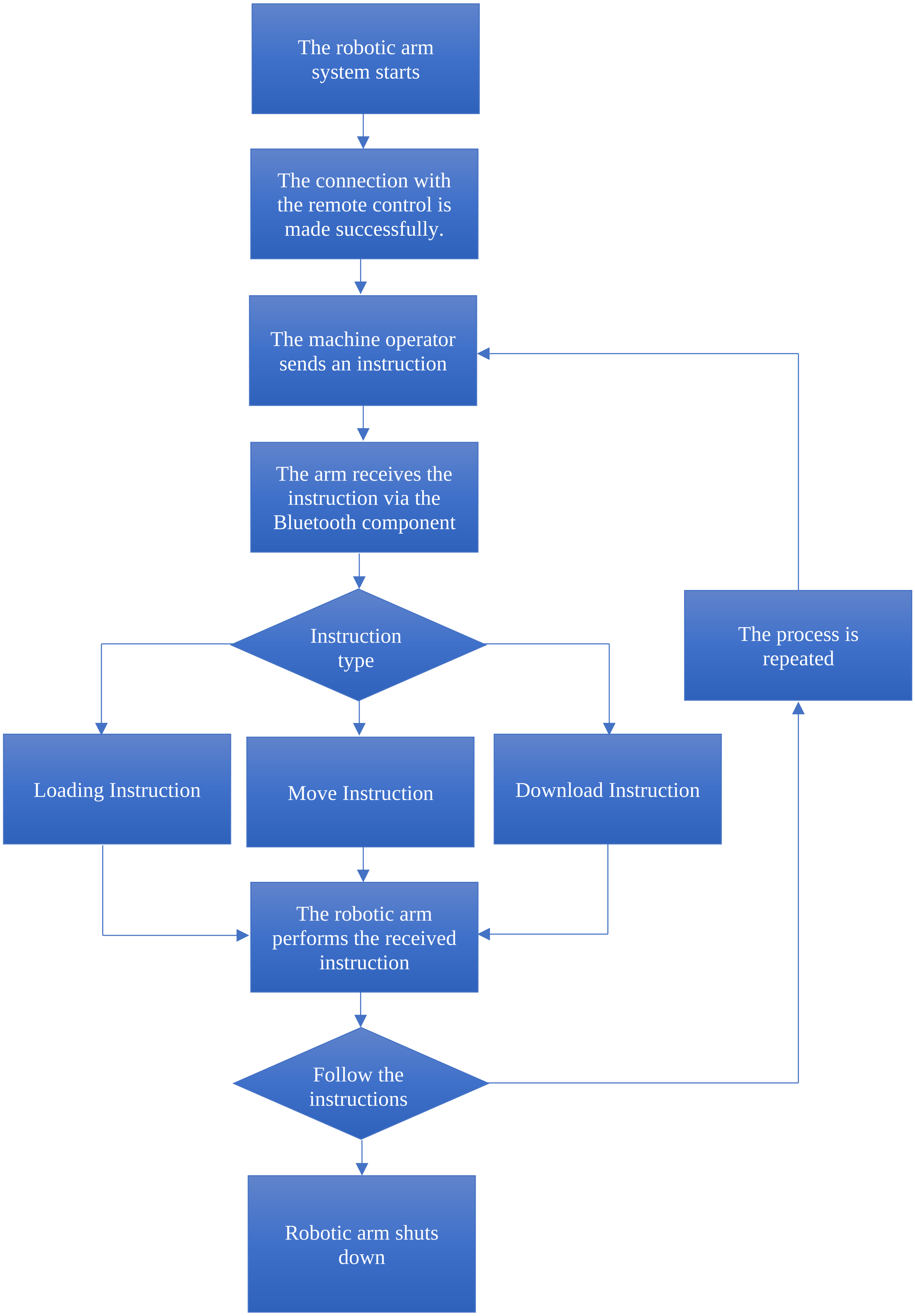

Fig. 8 Flowchart

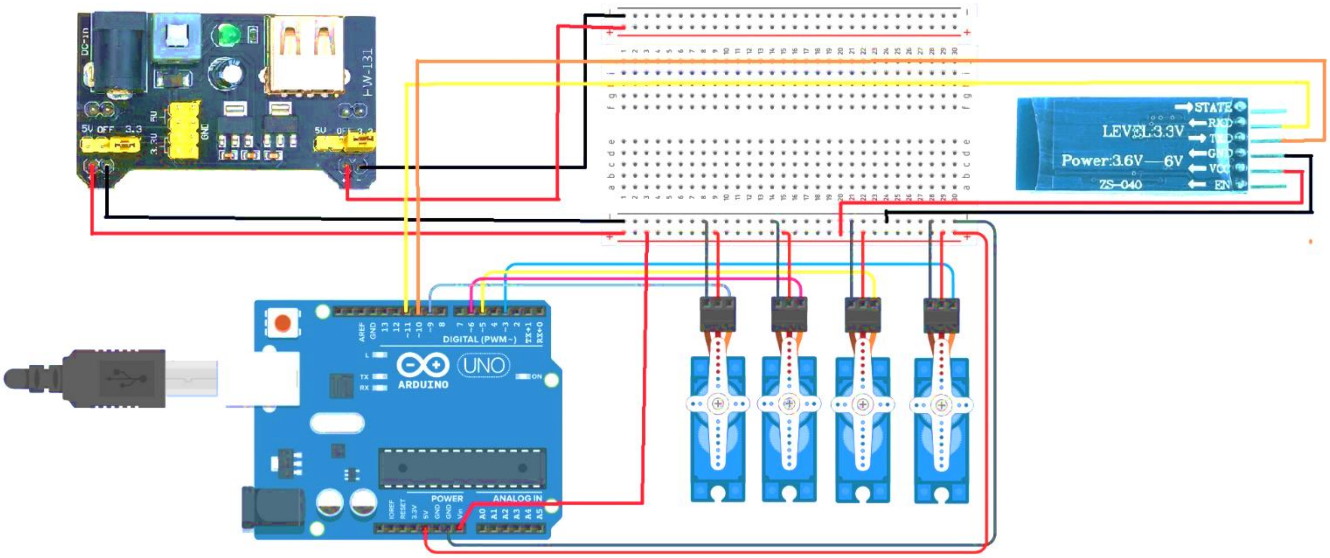

Fig. 9 Circuit design

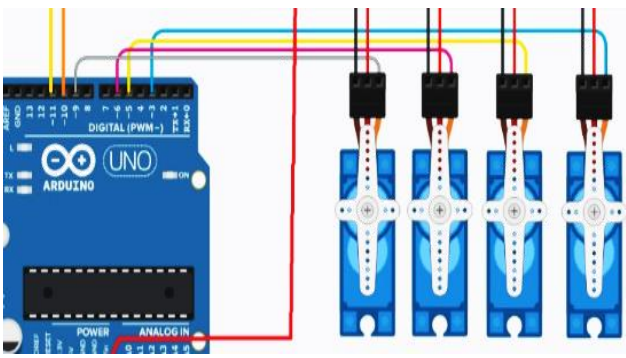

Fig. 10 Servomotor and arduino

Fig. 11 Arduino bluetooth component

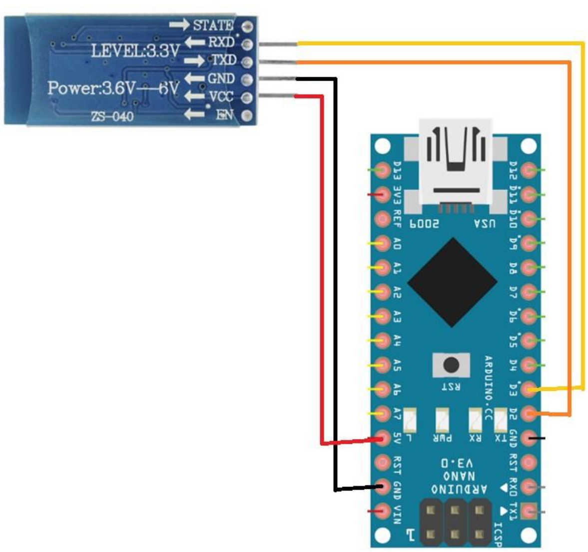

Fig. 12 Component bluetooth joystick

Fig. 13 HW - 131 power supply

3.3. Circuit Design

The diagram found in Figure 8 shows the operation of the robotic arm. From the beginning, it is explained how he receives the instruction and the process he goes through to complete it.

In Figure 9 and Figure 10, the circuit design and the functionality of the robotic arm can be observed through the use of servo motors. A joystick will control the servomotors. In turn, it will be connected to a computer to send signals through the serial monitor and later through Bluetooth. For this reason, two HC-05 modules are required. The first will be connected to the "emitter" Arduino, as shown in Figure 12, while the second will be in charge of receiving the signal, which can be seen in Figure 11.

The HW-131 component will be used to feed the servomotors correctly. This is a “breadboard power supply”, as shown in Figure 13. It can supply voltages of 3.3 V and 5 V, depending on the configuration.

Finally, the breadboard will allow you to connect and organize the circuit correctly. The design of this circuit is based on controlling the servomotors through a Bluetooth signal.

3.4. Circuit Programming

The attach method of the servo library is implemented by providing the pins to which the servomotors are connected, and then the positions are passed.

First, the variables to be used must be defined, including the necessary libraries, as shown in Figure 14. The Software Serial and Servo libraries are defined. There are variables for the pins, the initial positions, the servomotors themselves, and the integer variable dt.Figure 15 is the setup method in which the values are initialized.

```
#include <SoftwareSerial.h>
#include <Servo.h>

SoftwareSerial esclavo(10, 11);  // RX, TX


int servoPin1 = 5;
int servoPin2 = 6;
int servoPin3 = 9;
int servoPin4 = 3;

int servoPos1 = 90;
int servoPos2 = 90;
int servoPos3 = 90;
int servoPos4 = 90;

Servo myServo1;
Servo myServo2;
Servo myServo3;
Servo myServo4;

int dt = 100;
```

Fig. 14 Variable programming

```
void setup() {
  esclavo.begin(38400);
  Serial.begin(9600);

  myServo1.attach(servoPin1);
  myServo2.attach(servoPin2);
  myServo3.attach(servoPin3);
  myServo4.attach(servoPin4);


  myServo1.write(servoPos1);
  myServo2.write(servoPos2);
  myServo3.write(servoPos3);
  myServo4.write(servoPos4);
}
```

Fig. 15 Setup method

In the programming of the “loop” method, Figure 16 shows a structure of conditions that evaluate the slave variable to assign its value to a char-type variable later. Subsequently, with the variable that receives the value, the different scenarios are evaluated depending on the value it contains, as can be seen in the programming flowchart in Figure 17.

Once the condition is met, it will enter the structure and increase (or decrease) the servomotor's position related to the character that contains the variable. For example, the "w" increases because it represents going forward, or the "s" decreases because it represents going backwards. Once the position is updated, the servomotor position must be overwritten.

```
void loop() {
  if (esclavo.available()) {
    char data = esclavo.read();

    if (data == 'w' && servoPos1 < 175) {
      servoPos1 += 5;
      myServo1.write(servoPos1);
    } else if (data == 's' && servoPos1 > 5) {
      servoPos1 -= 5;
      myServo1.write(servoPos1);
    } else if (data == 'a' && servoPos2 < 175) {
      servoPos2 += 5;
      myServo2.write(servoPos2);
    } else if (data == 'd' && servoPos2 > 5) {
      servoPos2 -= 5;
      myServo2.write(servoPos2);
    } else if (data == 'q' && servoPos3 < 175) {
      servoPos3 += 5;
      myServo3.write(servoPos3);
    } else if (data == 'e' && servoPos3 > 5) {
      servoPos3 -= 5;
      myServo3.write(servoPos3);
    } else if (data == 'r' && servoPos4 < 175) {
      servoPos4 += 5;
      myServo4.write(servoPos4);
    } else if (data == 'f' && servoPos4 > 5) {
      servoPos4 -= 5;
      myServo4.write(servoPos4);
    }
    //delay(dt);
  }
}
```

Fig. 16 Loop method

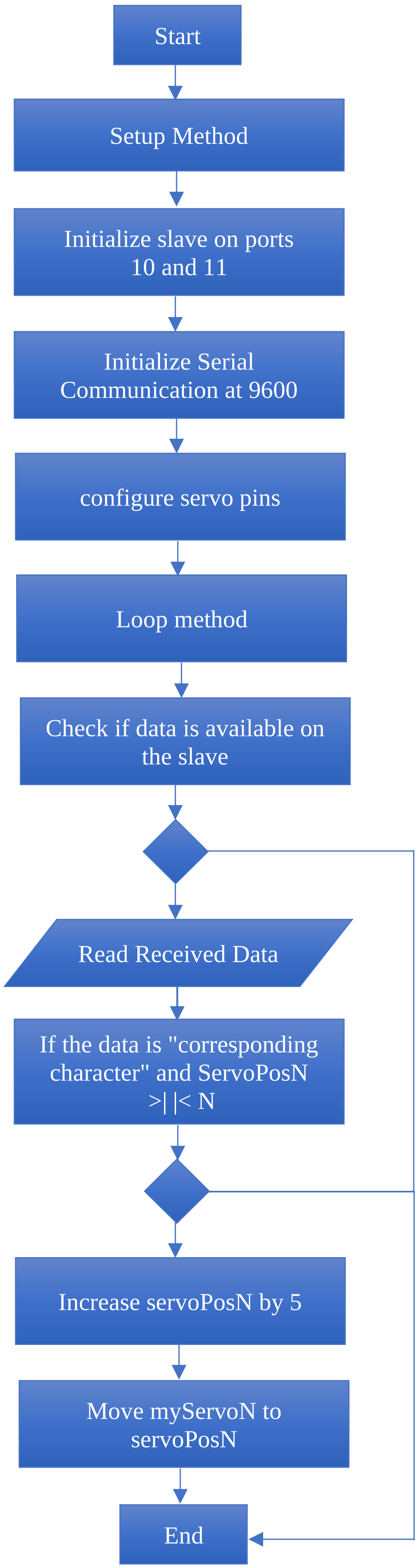

Fig. 17 Programming flow diagram

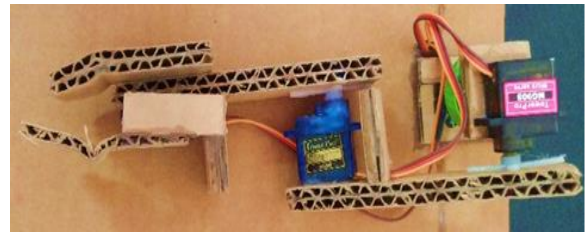

Fig. 18 Arm assembly

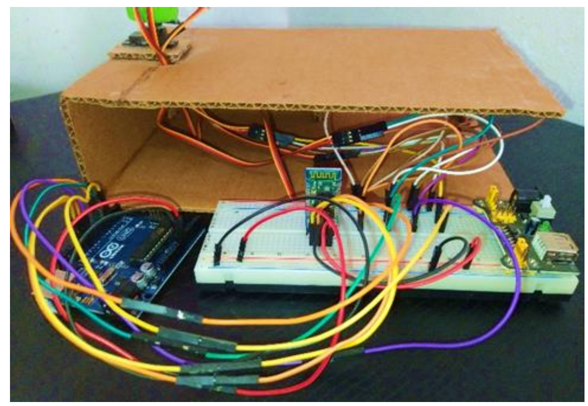

Fig. 19 Arm base

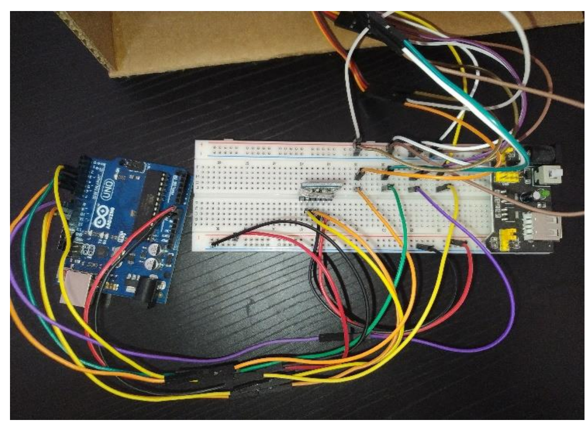

Fig. 20 Arm circuit

3.5. Circuit Implementation

For the implementation of the circuit, cardboard will be used for the body of the robotic arm. For the part of the arm, the parts were cut and glued with the servomotors as shown in Figure 18. It should be noted that it can even be simplified to three angles of freedom [19]. Based on previous research, it was decided to implement a robotic arm with four angles of freedom. All of this is to optimize design and costs [20].

A cardboard base is placed on it for better support, and the circuit will go below it. The robotic arm is fixed on the base, passing its cables underneath; it can be seen in Figure 19.

The circuit with the Arduino, breadboard, power supply, Bluetooth, etc., is placed in the lower part of the base. It can be seen in Figure 20.

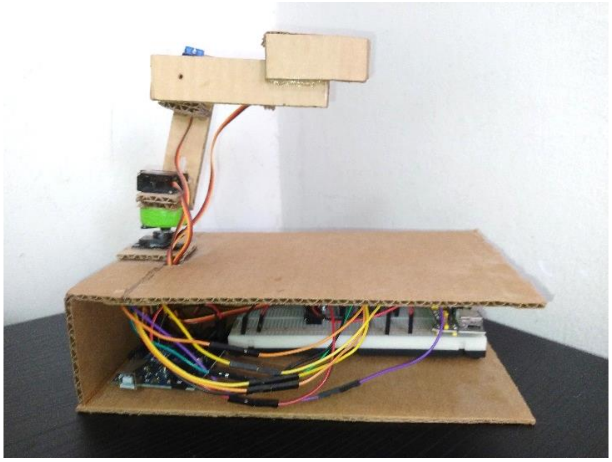

Fig. 21 Implemented circuit

Fig. 22 Joystick

Everything will be made up of the cardboard base, the Arduino, breadboard, power supply, Bluetooth component, etc. In the upper part, the arm with the connections between the servomotors is connected directly to the circuit. It can be seen in Figure 21.

A joystick and a program called Processing will be used to implement the control, which can be seen in Figure 23. This will help with the programming and instructions of the control, which can be seen in Figure 22.

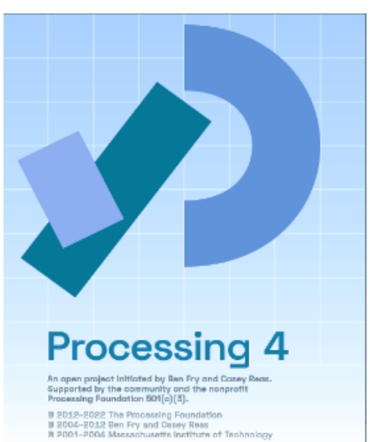

Fig. 23 Processing program

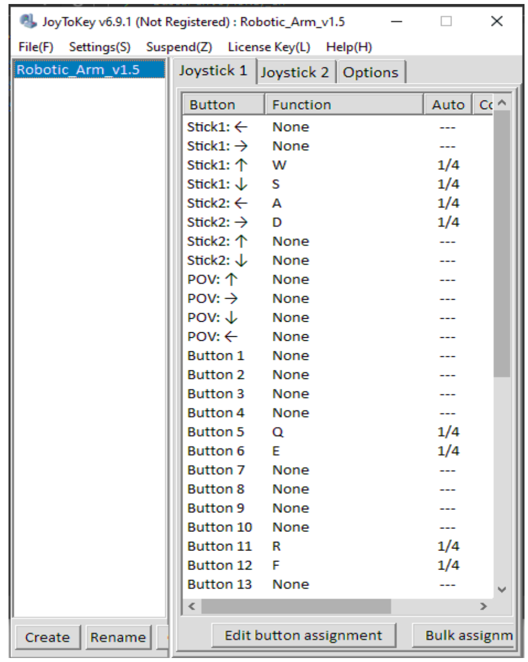

Fig. 24 Joystick configuration

The "JoyToKey" program opens, which will be used to change the joystick pulsations to simulate the computer/laptop keyboard. Figure 24 shows the configurations already made for the project.

Then, in Processing, click on the "Run" button; this will open a popup window allowing you to use the predefined keys to send data through the serial monitor without needing to press "Enter" after each command. It can be seen, in Figure 25 that some variables are established, the "setup()" method and the "draw()" method.

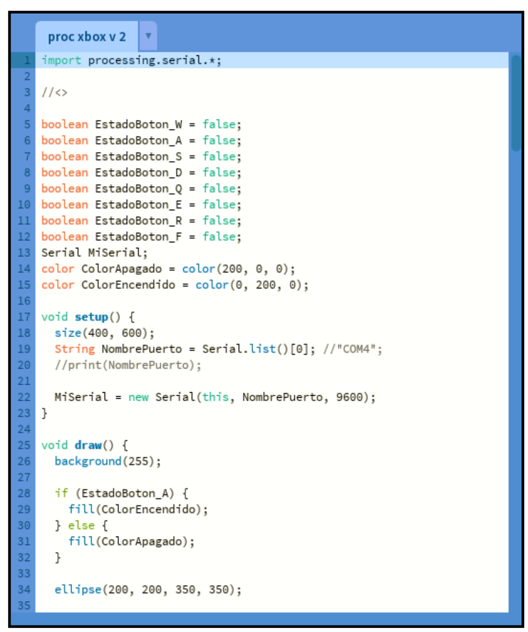

Fig. 25 Joystick programming

```
void keyPressed() {
  if (key == 'a') {
    if (EstadoBoton_A != true) {
      EstadoBoton_A = true;
      MiSerial.write('a');
    }
  }

  if (key == 'w') {
    if (EstadoBoton_W != true) {
      EstadoBoton_W = true;
      MiSerial.write('w');
    }
  }

  if (key == 's') {
    if (EstadoBoton_S != true) {
      EstadoBoton_S = true;
      MiSerial.write('s');
    }
  }

  if (key == 'd') {
    if (EstadoBoton_D != true) {
      EstadoBoton_D = true;
      MiSerial.write('d');
    }
  }

  if (key == 'q') {
    if (EstadoBoton_Q != true) {
      EstadoBoton_Q = true;
      MiSerial.write('q');
    }
  }

  if (key == 'e') {
    if (EstadoBoton_E != true) {
      EstadoBoton_E = true;
      MiSerial.write('e');
    }
  }

```

Fig. 26 Joystick conditionals

```
  if (key == 'r') {
    if (EstadoBoton_R != true) {
      EstadoBoton_R = true;
      MiSerial.write('r');
    }
  }

  if (key == 'f') {
    if (EstadoBoton_F != true) {
      EstadoBoton_F = true;
      MiSerial.write('f');
    }
  }
}

void keyReleased() {
  if (key == 'a') {
    if (EstadoBoton_A != false) {
      EstadoBoton_A = false;
    }
  }

  if (key == 'w') {
    if (EstadoBoton_W != false) {
      EstadoBoton_W = false;
    }
  }

  if (key == 's') {
    if (EstadoBoton_S != false) {
      EstadoBoton_S = false;
    }
  }

  if (key == 'd') {
    if (EstadoBoton_D != false) {
      EstadoBoton_D = false;
    }
  }
```

Fig. 27 Joystick conditionals

The “keyPressed()” method evaluates keypresses. Figure 26 shows the conditionals to capture the keystrokes (in lower case) “A”, “W”, “S”, “D”, “Q”, and “E”. Boolean values are set to “true”.

Figure 27 also shows the keystrokes (in lower case) "R" and "F". In addition, there is the beginning of the “keyReleased()” method, which captures the information when a key is no longer pressed. In this case, and in contrast to the “keyPressed()” method, the Boolean values are returned to “false”.

```
  if (key == 'q') {
    if (EstadoBoton_Q != false) {
      EstadoBoton_Q = false;
    }
  }

  if (key == 'e') {
    if (EstadoBoton_E != false) {
      EstadoBoton_E = false;
    }
  }

  if (key == 'r') {
    if (EstadoBoton_R != false) {
      EstadoBoton_R = false;
    }
  }

  if (key == 'f') {
    if (EstadoBoton_F != false) {
      EstadoBoton_F = false;
    }
  }
}

void sendSerialData() {
  if (EstadoBoton_A) {
    MiSerial.write('a');
  }

  if (EstadoBoton_W) {
    MiSerial.write('w');
  }

  if (EstadoBoton_S) {
    MiSerial.write('s');
  }

  if (EstadoBoton_D) {
    MiSerial.write('d');
  }
```

Fig. 28 Joystick conditionals

As in Figure 27, Figure 28 shows conditionals for four more keys. Likewise, the start of the “sendSerialData()” method, which is responsible for sending the information through serial communication, is shown. In this way, the information is sent without the need to press “Enter” after each iteration, unlike the Arduino IDE.

Figure 29 shows the final part of the “sendSerialData()” method. In addition, there is the “mouseClicked()” method, which is used to interact with the example figure generated through the “draw()” method.

Finally, in the section seen above, you can modify the instructions given by the joystick through conditionals.

```
  if (EstadoBoton_Q) {
    MiSerial.write('q');
  }

  if (EstadoBoton_E) {
    MiSerial.write('e');
  }

  if (EstadoBoton_E) {
    MiSerial.write('r');
  }

  if (EstadoBoton_E) {
    MiSerial.write('f');
  }
}

void mouseClicked() {
  float distancia = dist(200, 200, mouseX, mouseY);
  if (distancia < 175) {
    EstadoBoton_A = !EstadoBoton_A;
    MiSerial.write('a');
  }
}
```

Fig. 29 Joystick conditionals


Fig. 30 Arm in the base position

4. Results and Discussion
4.1. Results
Once the implementation is finished, the corresponding tests are carried out.

The arm is turned on, and a left turn movement is made from the reader's perspective. It can be seen in Figure 30 and Figure 31.

As you can see from the base, it rotated at an angle of 90°. Now, there will be a turn from top to bottom.


Fig. 31 Arm rotating to the left

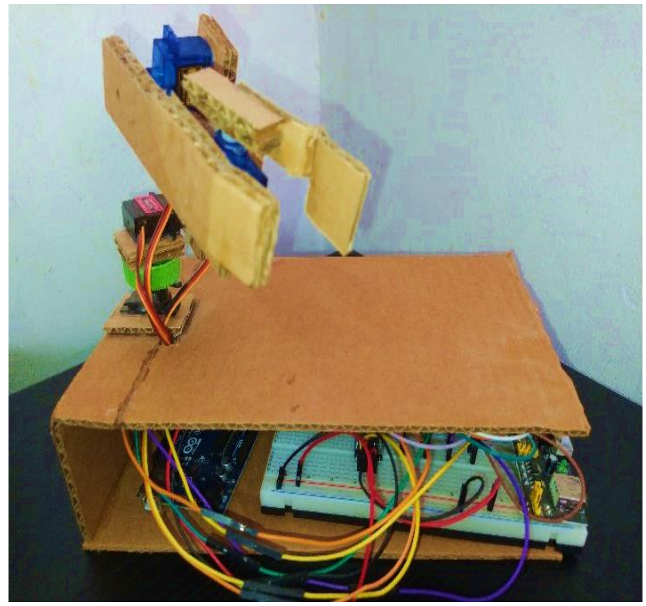
Fig. 32 Turn down comparison

In Figure 32, the arm rotates downward from the base. In the following image, you can see a similar turn, but from the servo motor that is at the top to the base. Now, the servomotor that controls the hand or spikes of the robotic arm that will allow the objects to be manipulated will be shown.

In Figure 33, you can see the functionality of the spikes or hand of the robotic arm.


Fig. 33 Robotic arm grip

Table 6. Explanation of the movement of the robotic arm

Command	Action
The "wrist" of the robotic arm moves according to the signal sent, 45 degrees. (Opening range 15-165 degrees)	Figure 31
The "shoulder" of the robotic arm moves according to the signal sent, approximately 85 degrees. (Opening range 15-165 degrees)	Figure 32
The "wrist" of the robotic arm returns to the starting position of 90 degrees. (Range 15-165 degrees)	Figure 33

Table 6 shows the explanation of the movements used by the robotic arm, mentioning the corresponding figures.

4.2. Discussions

The objective of this project is to design and implement a robotic arm for the Sodimac company that works with many heavy products and, therefore, with heavy machinery. With the design and implementation of this prototype, the creation of the robotic arm that will carry out the tasks proposed by the said company was achieved.

Implementing the project will allow optimizing the processes of the company in question and keep workers safe. This is because they will not be doing many jobs with heavy goods directly but instead using the robotic arm and leaning on it. In the development of the project, it was possible to appreciate that there were some failures, such as poor voltage control, which caused the servomotors not to work correctly. They had tics or simply did not carry out the instructions provided in the programming. To solve this, the HW-131 component, which is a power supply that allowed the error to be corrected, was implemented.

Servomotors were implemented in the project without much power, which is why cardboard was used as the body of the robotic arm so that it does not have much weight, but on a real scale, these motors must have much more power to be able to perform the task that is sought.

The project's proposed objectives were completed, thus achieving the implementation of the robotic arm with multiple servomotors that facilitate the movement and manipulation of the robotic arm. In addition, it has a joystick that allows remote control.

5. Conclusion and Recommendations

Regarding the research carried out, it was concluded that this project will be very useful for future projects.It is concluded that the project in question will serve to implement a large-scale robotic arm in the company Sodimac.To carry out this large-scale prototype, the servomotors to be used must be taken into account according to the work to be carried out since the power of these motors and their choice depend on that. The movement of the servomotors was continuous and repetitive at the beginning, preventing its control, but it was managed to be solved with a special power supply for the breadboard.It is advisable to carry out the presentation stage of the prototype, refining the physical details for a better appreciation.It is recommended to implement different types of grips to specialize in different loading processes or tasks to be performed. It is recommended to use a better material for the prototype for a better grip and movement, improving the quality of the servomotors

Funding Statement

The article was funded thanks to the Universidad Privada del Norte, Lima, Perú.

References

[1] Meza Chipoco, Beatriz del Milagro Umiña Malma, and Jeison Cristian, "*Proposal to Evaluate and Control Work Fatigue in Heavy Load Drivers in the ACOINSA Transport Company*," Thesis, Universidad Tecnológica del Perú, pp. 1-99, 2019. [Google Scholar] [Publisher Link]

[2] Buelvas Díaz, and Camilo Ernesto, "*Preparation of a Preventive Maintenance Plan for the Heavy Machinery of the L&L Company*," Thesis, Universidad Autónoma Del Caribe, pp. 1-72, 2014. [Google Scholar] [Publisher Link]

[3] Lina Natalia Paredes-Sanchez et al., "Occupational Safety and Health for Heavy Machinery Operators in the Construction Industry," *Revista Investigación en Salud Universidad de Boyacá*, vol. 9, no. 1, pp. 149-165, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Benavente Inca, Karla Antonella Soria Guadalupe, and Mitchel Dante, "*Design of a Palletizing Robotic Arm for Optimization within the Dispatch Process in the Cement Industry*," Thesis, Universidad Ricardo Palma, pp. 1-132, 2020. [Google Scholar] [Publisher Link]

[5] Luna Gutiérrez, and Jorge Hernán, "*Adaptive Safety System for Robotic Arms*," Pontificia Universidad Católica Del Perú, pp. 1-90, 2021. [Google Scholar] [Publisher Link]

[6] Juan Sebastián Rojas Eslava et al., "Design and Implementation of a Low-Cost Robotic Arm for Automation in the Bacteriological Analysis Process," *INGE CUC*, vol. 8, no. 1, pp. 219-230, 2012. [Google Scholar] [Publisher Link]

[7] Michael E. Moran, "Evolution of Robotic Arms," *Journal of Robotic Surgery*, vol. 1, pp. 103-111, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[8] Mohd Ashiq Kamaril Yusoff, Reza Ezuan Samin, and Babul Salam Kader Ibrahim, "Wireless Mobile Robotic Arm," *Procedia Engineering*, vol. 41, pp. 1072-1078, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[9] Sébastien Mick et al., "Biological Plausibility of Arm Postures Influences the Controllability of Robotic Arm Teleoperation," *Human*

Factors: The Journal of the Human Factors and Ergonomics Society, vol. 64, no. 2, pp. 373-384, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] R. Boumans, and C. Heemskerk, "The European Robotic Arm for the International Space Station," *Robotics and Autonomous Systems*, vol. 23, no. 1-2, pp. 17-27, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[11] Yeison Alejandro Aldana, "Exploration Robotics, a World of Possibilities," *Indagare*, no. 8, pp. 227-234, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] Andres Patrignani et al., "Review of Rain or Shine: An Introduction to Soil Physical Properties and Processes," *Vadose Zone Journal*, vol. 21, no. 3, pp. 1-2, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Ali Roshanianfard, Noboru Noguchi, and Tatsuki Kamata, "Design and Performance of a Robotic Arm for Farm Use," *International Journal of Agricultural and Biological Engineering*, vol. 12, no. 1, pp. 146-158, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] Carlos Adrián Mainetti et al., "Development and Analysis of the Response of a Manipulation Gripper for Robotic Arms," *Ciencia y Tecnología*, no. 18, pp. 53-70, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] Kevin Morales, Carlos Hoyos, and Jesús García-Caicedo, "Development Design and Optimization of the Mechanical Structure of an Anthropomorphic Robotic Arm Developed for Educational Purposes," *Revista UIS Ingenierías*, vol. 18, no. 4, pp. 193-208, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[16] Paola Andrea Rey Molina, and Karent Urrego, "Design and Implementation of a 3D Robotic Arm Prototype for the Handling of Boxes in a Storage Matrix," *Redes de Ingeniería*, vol. 3, no. 2, pp. 32-40, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[17] Jesús Chacon et al., "A Low-Cost, Open-Source Remote Lab for the Dobot Magician Educational Robotic Arm," *Revista Iberoamericana de Automática e Informática industrial*, vol. 20, no. 2, pp. 124-136, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Quilca Tito, and Erick Junior, "*Relationship of Deformations of Concrete Floor Joints with Premature Wear of Drive Wheels of Cargo Stackers in SODIMAC Warehouse Lima Peru 2015*," Thesis, Universidad Nacional Tecnológica de Lima Sur, pp. 1-58, 2015. [Google Scholar] [Publisher Link]

[19] Mohd Aliff, Shujiro Dohta, and Tetsuya Akagi, "Control and Analysis of Robot Arm Using Flexible Pneumatic Cylinder," *Mechanical Engineering Journal*, vol. 1, no. 5, pp. 1-13, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[20] Hanan A.R. Akkar, and Ahlam Najim A. Amir, "Kinematics Analysis and Modeling of 6 Degree of Freedom Robotic Arm from DFROBOT on Labview," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 13, no. 7, pp. 569-575, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[21] Hideki Kajima et al., "A study on a Brachiation Controller for a Multi-Locomotion Robot—Realization of Smooth, Continuous Brachiation," *Advanced Robotics*, vol. 18, no. 10, pp. 1025-1038, 2004. [CrossRef] [Google Scholar] [Publisher Link]