Enviado: 19/03/2023 *Recibido: 25/03/2023* *Aceptado: 08/05/2023*

UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS

VISIÓN ELECTRÓNICA

https://doi.org/10.14483/issn.2248-4728

VISIÓN ELECTRÓNICA

3 DOF robot programmed with PIC 18f45k22 for handling materials in a robotic cell

Robot de 3 DOF programado con PIC 18f45k22 para manipulación de materiales en una celda robotizada

***David Fernando Monroy Moya*[1] , *Diego Alfonso Rojas Sarmiento*[2] , *Fabián Barrera Prieto*[3]**

Abstract

This paper presents the interaction between different devices for the operation of the prototype of a 3-degree robotic arm applied to a robotic cell, which was programmed with a low-cost microcontroller (PIC18F45K22), using serial communication peripherals (UART and I2C), LCD and three presence sensors (infrared). In addition, the robot work area is presented, making use of Peter Corke's robotics toolbox in Matlab, as well as the printed circuit board (PCB) designed in EasyEDA for the electrical connections of the microcontroller and the other electronic components that work simultaneously with the robot. Finally, the results obtained by means of points and trajectories with different operating modes are presented through physical and simulated tests in Matlab.

Keywords: 3 DOF, I2C, Microcontroller, Peter Corke, Robotic arm, UART.

[1] Current position: Student / Universidad ECCI / Mechatronics Engineering Program / Colombia. E-mail: davidf.monroym@ecci.edu.co

[2] MSc. Mechatronics Engineering, Universidad Militar Nueva Granada, Colombia. Current position: Full-time Professor / Universidad ECCI / Mechatronics Engineering Program, Colombia. E-mail: drojass@ecci.edu.co

[3] MSc. Mechatronic Systems, Universidade de Brasília, Brazil. Current position: Full-time Professor / Universidad ECCI / Mechatronics Engineering Program, Colombia. E-mail: fbarrerap@ecci.edu.co

Resumen

Este documento presenta la interacción entre diferentes dispositivos para el funcionamiento del prototipo de un brazo robótico de 3 grados aplicado a una celda robotizada, el cual fue programado con un microcontrolador de bajo costo (PIC18F45K22), utilizando periféricos de comunicación serial (UART e I2C), LCD y tres sensores de presencia (infrarrojos). Además, se presenta el área de trabajo del robot, haciendo uso del toolbox de robótica de Peter Corke en Matlab, así como la placa de circuito impreso (PCB) diseñada en EasyEDA para las conexiones eléctricas del microcontrolador y los demás componentes electrónicos que trabajan simultáneamente junto al robot. Finalmente, se presentan los resultados obtenidos mediante puntos y trayectorias con diferentes modos de funcionamiento mediante pruebas físicas y simuladas en Matlab.

Palabras clave: 3 DOF, I2C, Microcontrolador, Peter Corke, Brazo robótico, UART.

1. Introduction

Robotic arms present characteristics that nowadays facilitate the elaboration and manipulation of different objects destined to the automation of different tasks granting a multifunctionality to the companies that make use of this type of manipulators that look for mainly, greater speed and displacement or on the contrary, precision and dexterity. Nowadays in all industrial fields, robots are much more precise than humans in the execution of operations, but also faster, more dynamic, stable and resistant [1]. One of the main applications of industrial robots is the loading and unloading of production, these activities are carried out in a robotic cell, which consists of one or more production machines, the robot and a material handling mechanism to deliver the parts in and out of the cell [2], furthermore robotic cells are highly automated production systems capable of producing a wide variety of jobs [3] and depending on the degree of interaction required, the human and the robot can even work at the same time on the same work piece in what is called collaborative work [4]; moreover, they count on the integration of several devices in the task to interface with other systems and human operators [5] due to the rapid development of robotics and automation techniques, to adjust cell orientations [6].

The design used for this project is a 3 DOF (3R) anthropomorphic robot, which moves in a three-dimensional work area, and can be integrated with a high variety of applications when compared to other robots that work in two dimensions. Likewise, it is important to highlight that its small number of rotational joints does not limit it to basic tasks, since it can move in work areas of considerable size without presenting difficulties. This type of robotic cells handle sensors to detect obstacles or people within the robot's working area, i.e., for effective supervision systems, a large distance between the human and the robot is needed and, for close interaction, a large advanced sensing capability is required [7], even in some cases the use of sensors inside the robot is used for safety [8]. The management and control of the servomotors that the robot has for the movement and rotation of the joints, is done through a driver (PCA9685) that allows to control up to 16 PWM channels , which works through I2C [9].

For this project the use of communication peripherals such as I2C and UART is indispensable, due to the integration between electronic devices (sensors and actuators), it is also important to mention that the sensors used are digital so they do not require the use of ADC modules "available in the vast majority of modern 8-bit microcontrollers" [10], also the inclusion of multimedia technologies (Matlab) and microcontroller units (MCU) play an important role in the development of the project [11].

Firstly, the I2C communication is based on master-slave networks of different devices through an 8-bit addressing where the least significant bit represents read or write and the next 7 bits is the data to send or receive [12]. The device named as master will be a microcontroller of the microchip brand of reference PIC18F45K22, since due to its hardware characteristics, mainly its peripheral modules allow applications in numerous automation and robotics tasks counting with 28 ADC channels, up to 2 CCP modules, SPI, I2C, USART Module and others [13]. On the other hand, the slaves of the I2C communication will be the PCA9685 and the LCD which is connected to a pin expander module (PCF8574) that allows the LCD communication over the I2C bus. On the other hand, the sending and receiving of data to the microcontroller is done by asynchronous serial communication (UART) being a type of full-duplex serial communication, so it is widely used in data communications and control systems [14].

The programming for the inverse kinematics processing of the 3R robot is done through Peter Cork's robotics *toolbox*, generally used to represent the kinematics and dynamics of robotic manipulators in Matlab by means of homogeneous transformation matrices [15]. Likewise, the programming was worked with an object-oriented language to achieve communication between Matlab and the microcontroller because its environment can be worked graphically, and it also contains simpler and easier development tools, similar to other languages such as JAVA [16].

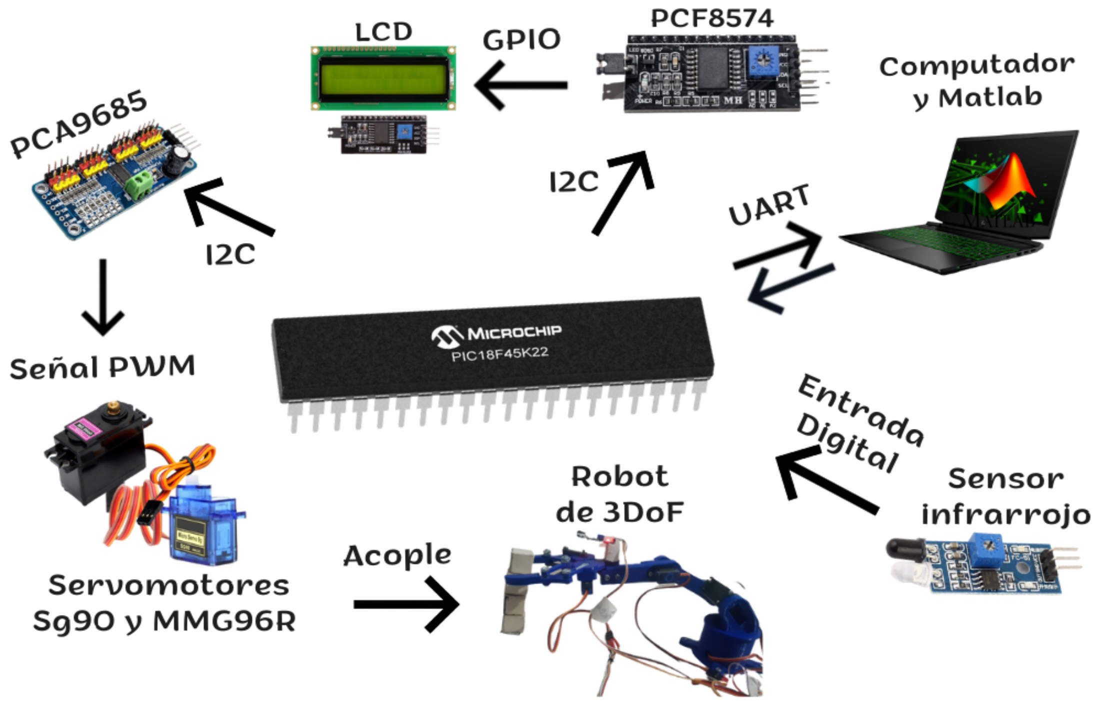
The equations for the robot kinematics are implemented using Matlab's own language for the points and trajectories to be performed by the robot; in addition, the kinematic model is useful when the complexity of the solutions is too high and requires time for its compression [17]. Likewise, the programming of the PIC microcontroller was done in C language, taking into account the main functions of interacting (receiving and/or transmitting information) with the other components of the project.

2. Methodology

2.1. Development

This project focused on industrial robotics and PIC microcontroller programming, with the main objective of performing the interaction between devices such as the computer, PIC (18F45K22), PCA9685, LCD (PCF8574) and infrared presence sensors, through UART and I2C communication protocols; as shown in Figure 1.

Figure 1. Interaction between devices and peripherals.



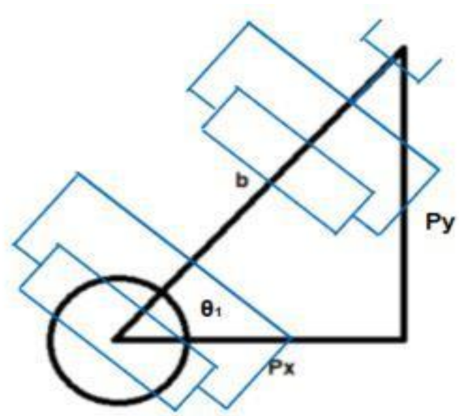
Source: own.

The approach of the inverse kinematics equations of the robot for the development of the programming algorithm with points and trajectories to be followed in each of the different operation modes, as well as the solution of the equations where the meaning of each variable is detailed and its development supported by graphics for a better understanding of them, is presented in subsection 2.2.1. Likewise, the approach of the algorithms and the flow charts of the programming of the codes for the microcontroller (C language) and for the inverse kinematics, through the MPLABX IDE and Matlab, respectively, are presented in subsection 2.2.2 of this document.

2.1.1. Inverse Kinematics 3R

Considering Figure 2, the calculation of the inverse kinematics of the 3R robot was performed from the variable 'b' which is the distance from the reference point (SC{0}) of the robot to the end effector (SC{4}), as well as represented in equation (1). With the same figure we obtain the equation (2) for the angle θ_1 of the robot describing the rotational motion of the first joint.

Figure 2. 3R robot top view.



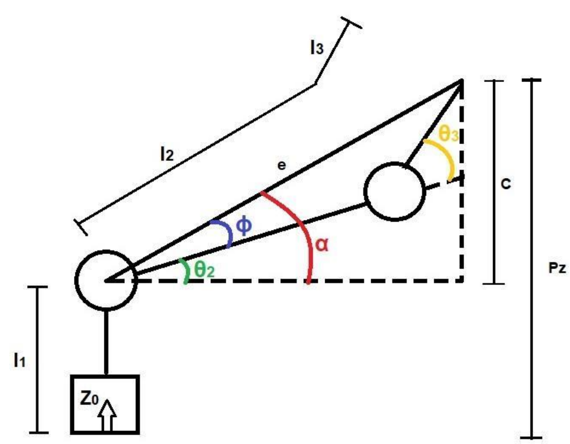
Source: own.

$$b = \sqrt{Px^2 + Py^2} \qquad (1)$$

$$\theta_1 = \left(\frac{Py}{Px}\right) \qquad (2)$$

Based on Figure 3, equation (3) is obtained, which is the distance between the first joint and the end effector with respect to the Z0 axis, being the necessary distance for the calculation of α y ϕ from equations (4) and (5), in order to obtain θ_2 note in equation (6).

Figure 3. 3R robot side view.



Source: own

$$c = Pz + l_1 \quad (3)$$

$$\alpha = \left(\frac{c}{b}\right) \quad (4)$$

$$\phi = \left(\frac{l_3 \cdot sin\, sin\, \theta_3}{l_2 + l_3 \cdot cos\, cos\, \theta_3}\right) \quad (5)$$

$$\theta_2 = \alpha - \phi \quad (6)$$

Finally, for the calculation of the angle θ_3 from figure 3 we obtain the $sin\theta_3$ and the $cos\theta_3$ to consequently obtain the angle value, see equations (7), (8) and (9).

$$cos\, cos\, \theta_3 = \frac{b^2 + c^2 - l_2^2 - l_3^2}{2 \cdot l_2 \cdot l_3} \quad (7)$$

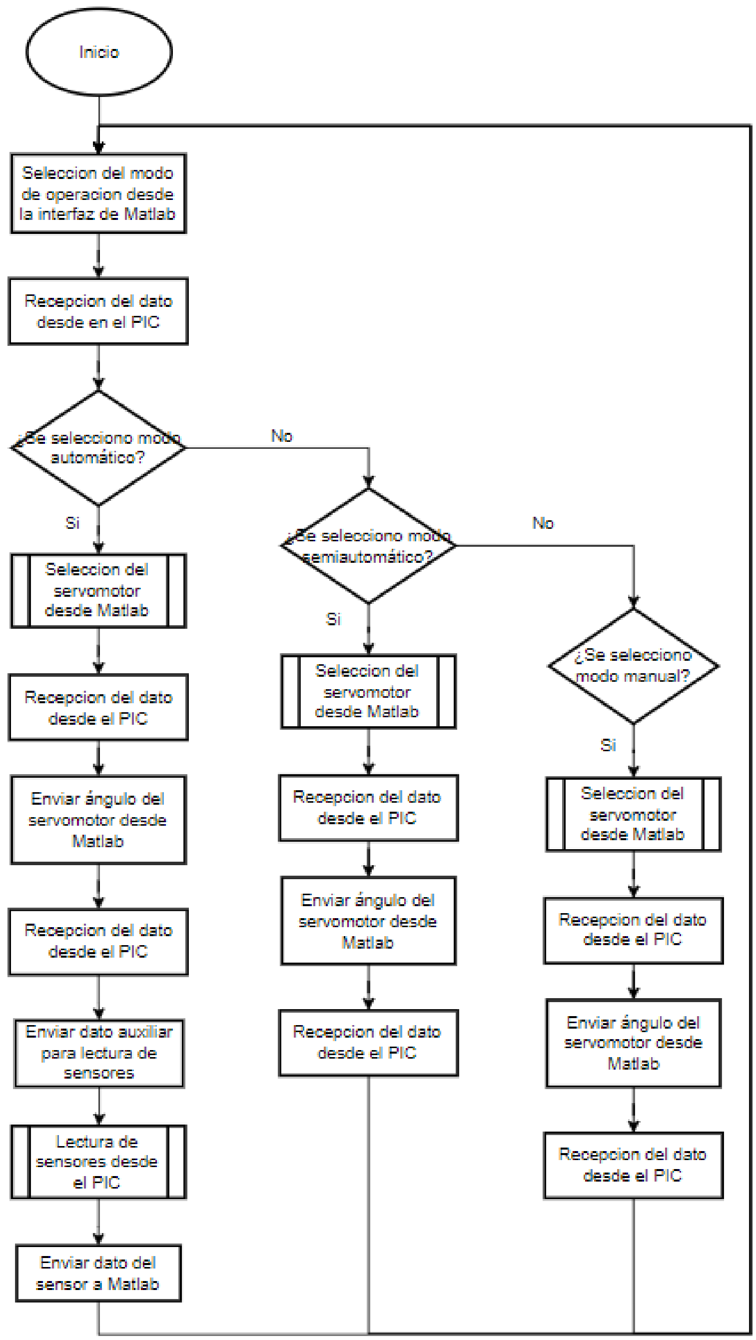
$$sin\, sin\, \theta_3 = \sqrt{1 - \theta_3} \quad (8)$$

$$\theta_3 = \left(\frac{sin\, sin\, \theta_3}{cos\, cos\, \theta_3}\right) \quad (9)$$

2.1.2. Algorithms and flowcharts.

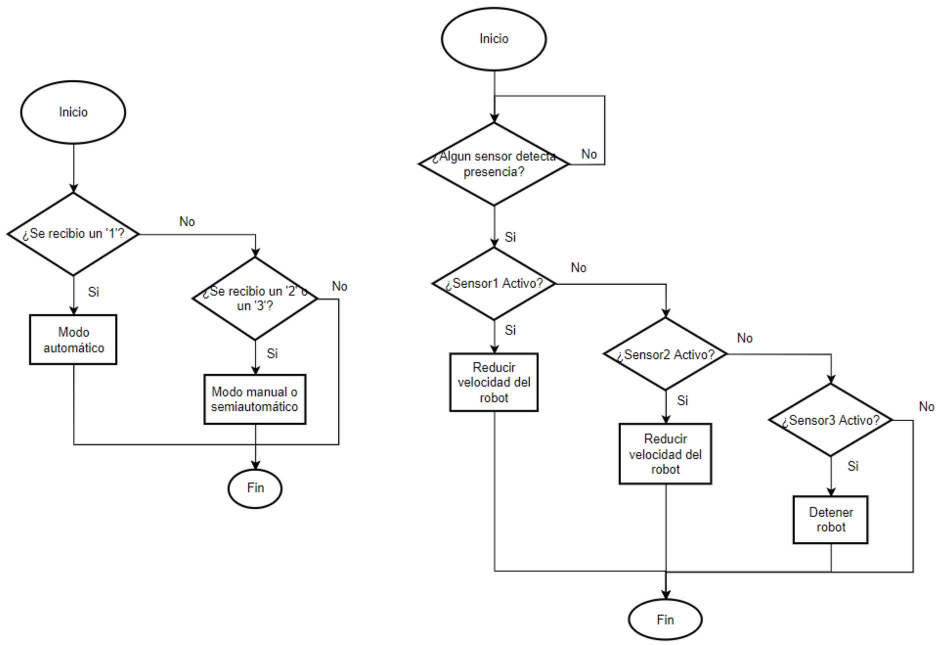
The development and implementation of the code is characterized by constant sending and receiving of data from the microcontroller through interruptions, since it is storing and operating information from the sensors and positioning angles for the servomotors that allow the movement of the robot, then, Figure 4 presents the general structure of the code implemented in the MPLABX IDE only with the use of interruptions without the existence of lines of code in the infinite loop. Figures 5 and 6 present the structure and classification of the sensors and servomotors.

Figure 4. General algorithm for reading, selection of operation mode and servomotors, reception of servomotor angle and distinction of presence sensors.



Source: own

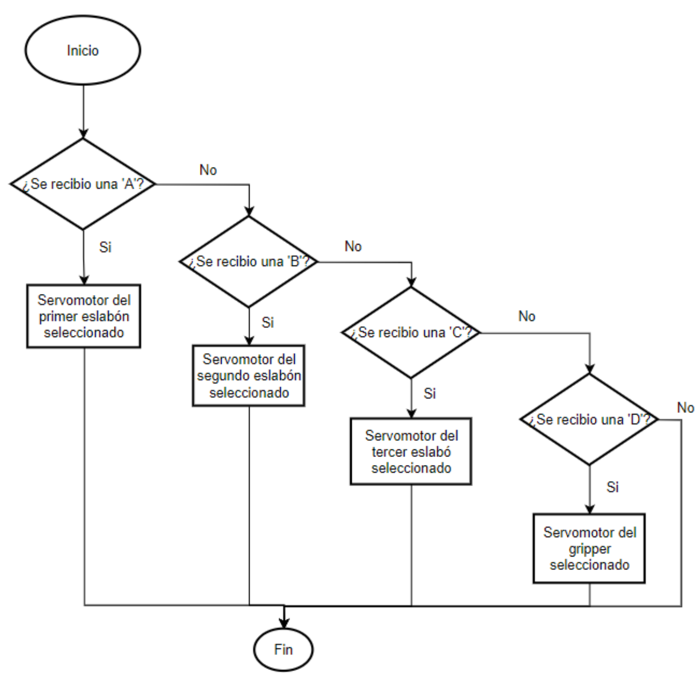
Figure 5. Flowchart for code implementation in Mplabx - (a) Selection of the operation mode and (b) distinction of the presence sensors.



(a). (b).

Source: own

Figure 6. Flowchart for code implementation in Mplabx - Servomotor selection.



Source: own

3. Design

During the creation of the code, the measurement of the links and degrees of freedom of each joint were taken into account, since the physical assembly was based on them; likewise, different servomotors were used according to the torque required for the movement of each joint, taking into account that three of its servomotors have a movement from 0° to 180° and a fourth servomotor that allows the opening or closing of the end effector from 0° to 50°. Tables 1 and 2 show some specifications of link measurements and degrees of freedom of each joint allowed by its respective servomotor.

Table 1. Measurement of the links of the 3R robot.

	Measure
Link 1	6.5 cm
Link 2	8.4 cm
Link 3 (gripper included)	19 cm

Source: own

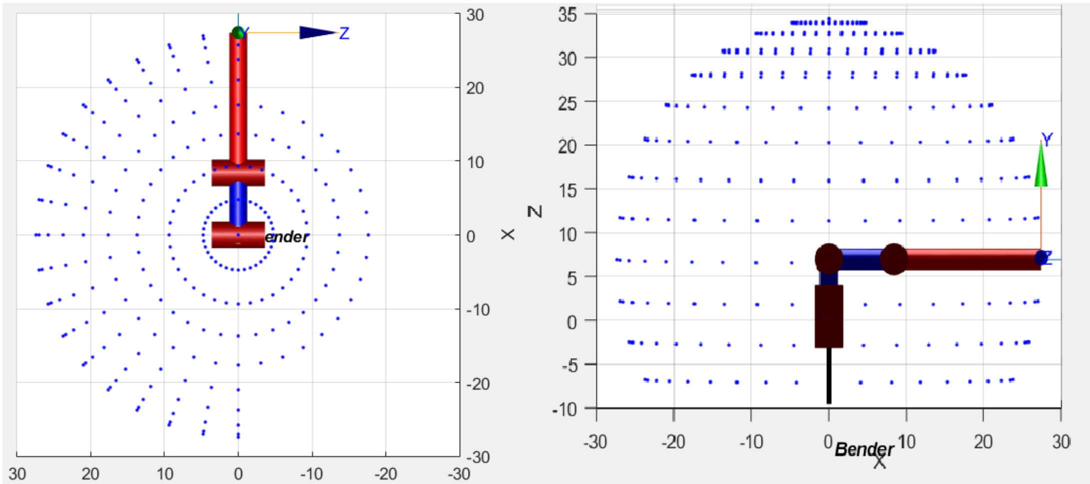
Table 2. Rotational motion of each joint and end effector of the 3R robot.

	Rotational movement
Articulation 1	0° a 180°
Articulation 2	-45° a 135°
Articulation 3	-90° a 90°
End effector	85° a 135°

Source: own

On the other hand, and as a complement to the tables shown above, Figure 7 shows the working area of the robot from two different perspectives in 2D, likewise, Figure 8 shows the working area from an isometric view (3D). In each of the following figures the group of points that can be reached by the end effector are indicated for a better relation of tables 1 and 2, and of the geometric limitations of the 3R robot.

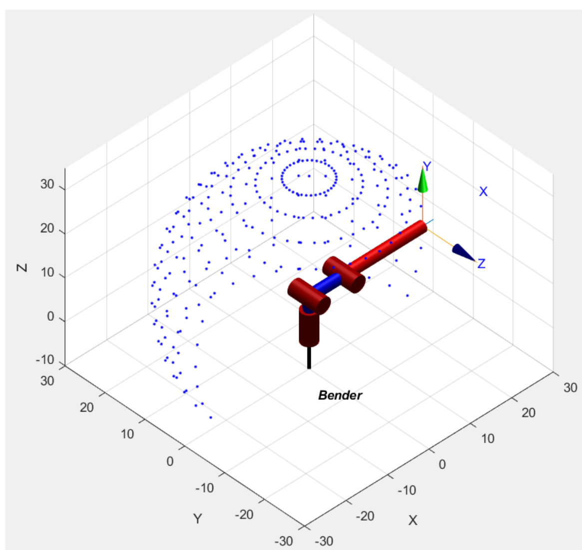
Figure 7. 3R robot working area from different 2D perspectives ((a) top view and (b) side view).



(a). (b).

Source: own

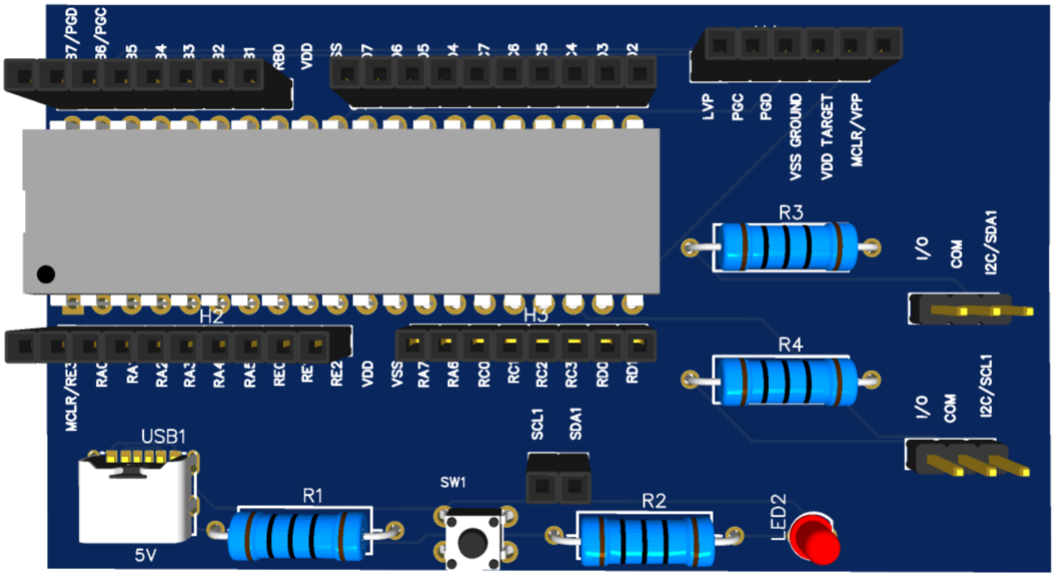
Figure 8. Work area of the 3R robot isometric view.



Source: own

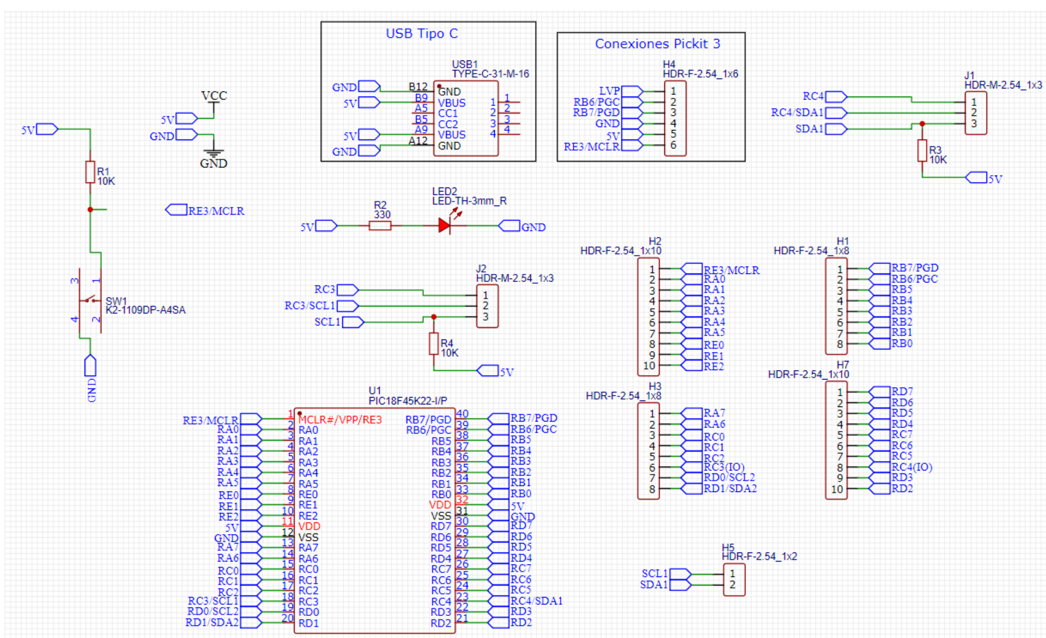
The printed circuit board (PCB) is presented in Figure 9, where its design is made in the EasyEDA double layer software with some special features such as USB type C connection for external power supply, special digital inputs for the connection of the Microchip programmer (PICKit3) and a pin selector that allows choosing the mode to be used in the PIC18F45K22, corresponding to UART communication, I2C communication, or digital inputs and outputs mode. Figure 10 shows the printed circuit board (PCB) design and the connection details for each of its functions.

Figure 9. Printed circuit board (PCB) layout.



Source: own

Figure 10. Electrical schematic for PCB design.

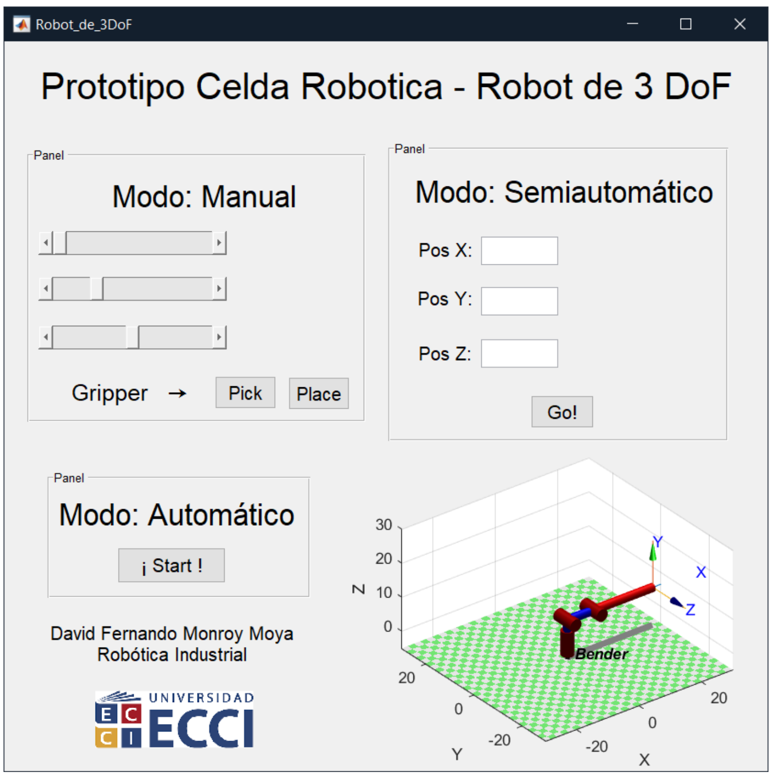


Source: own

4. Results

The operation of the robot is divided into 3 operating modes: (1) automatic, (2) manual and (3) semi-automatic. Their selection depends on the one chosen by the user through an HMI interface made with the Matlab GUI option, which is shown in Figure 11, where, apart from the mode selection, the simulated position of the robot and its final effector in the workspace can be visualized.

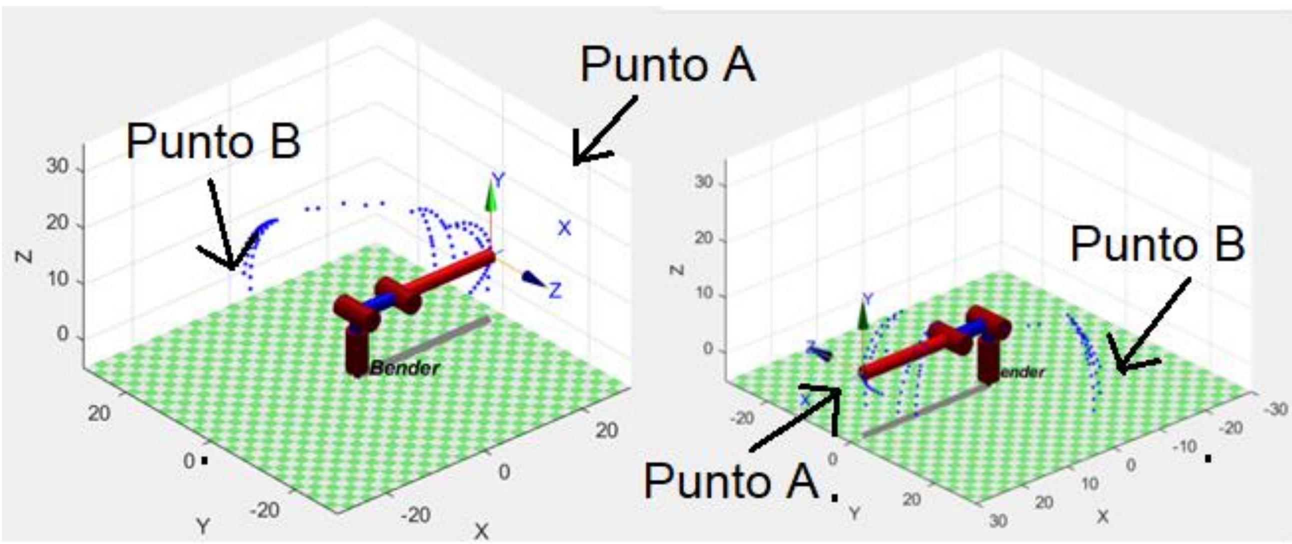
Figure 11. Human-Machine Interface (HMI) for the control and supervision of the 3R robot movements.



Source: own

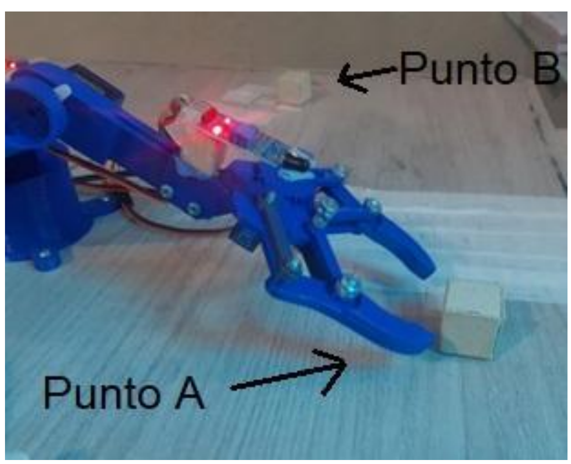
The automatic mode is characterized by following a *pick and place* process, where the main objective is to pick up cubes with a volume of $2cm^3$ located in a working space of $60cm^2$; the operability of this mode depends on the inverse kinematics of the 3R robot as its operation consists of moving to the position of the objects (point A) in order to move them at least 10 cm away (point B) to stack them one by one in a column of 4 positions for a total of 4 cubes per sequence. The programming of this mode is given by points and trajectories carried out by interpolation of angles in order to create an effect of simultaneous movement for each servomotor and consequently the joints move synchronously, ie the movement of the servomotors is executed at the same time and not one after another, then in Figure 12 and Figure 13 is presented the trajectory performed by the robot.

Figure 12. Trajectory in automatic mode of the robot (simulation).



Source: own

Figure 13. Trajectory in automatic mode of the robot (physical)**.**



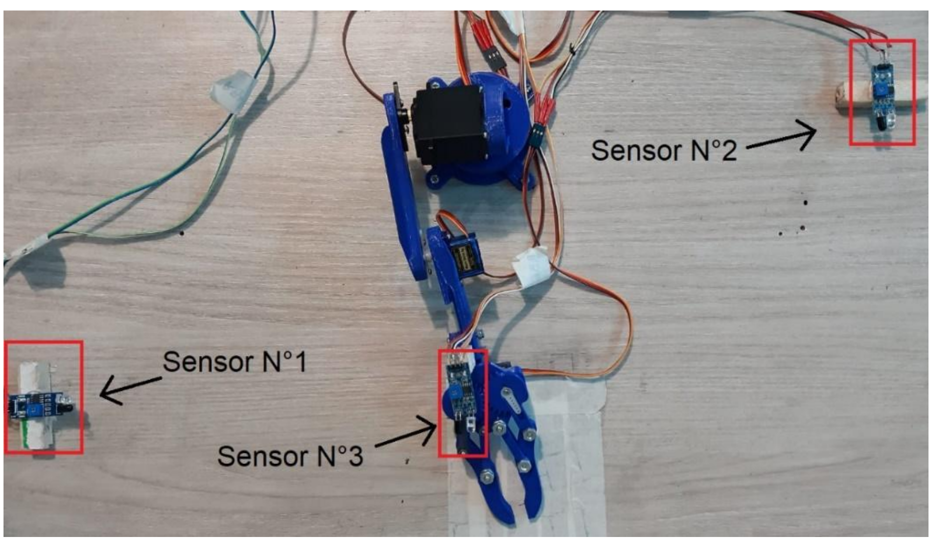
Source: own

The semi-automatic mode works by means of the equations of the inverse kinematics of the 3R robot, with the purpose of entering a vector with X, Y and Z coordinates by keyboard through an HMI interface, so that the robot advances and positions the end effector in that coordinate entered by the user, as long as it is within the range of the robot, that is, in the work area.

The manual mode refers to the operation of the stuffed animal machines with the notable difference that each servomotor is operated through sliders located in the Matlab HMI interface and for the manual operation of the end effector two buttons are incorporated that provide the option to open or close the gripper.

The integration of the sensors with the 3R robot (see Figure 14) is performed in the work area that emulates the prototype of the robotic cell, where three infrared sensors are used, two of them located in different corners of the robotic cell to reduce the speed of movement of the robot when detecting the presence of a person by reducing the speed of the servomotors; in addition, a third sensor located at the top of the end effector is implemented, which when activated or detecting presence completely stops the operation of the robot.

Figure 14. Sensors attached to the 3R robot.



Source: own

The LCD display is coupled with an interface module for I2C communication based on the PCF8574 integrated, mainly used for the reduction of connections and better coupling of the peripheral to the project. The LCD is mainly used as an output peripheral for the visualization of the angles received from Matlab to the microcontroller, as shown in Figure 15, in order to know information such as the current operation mode and the active sensors.

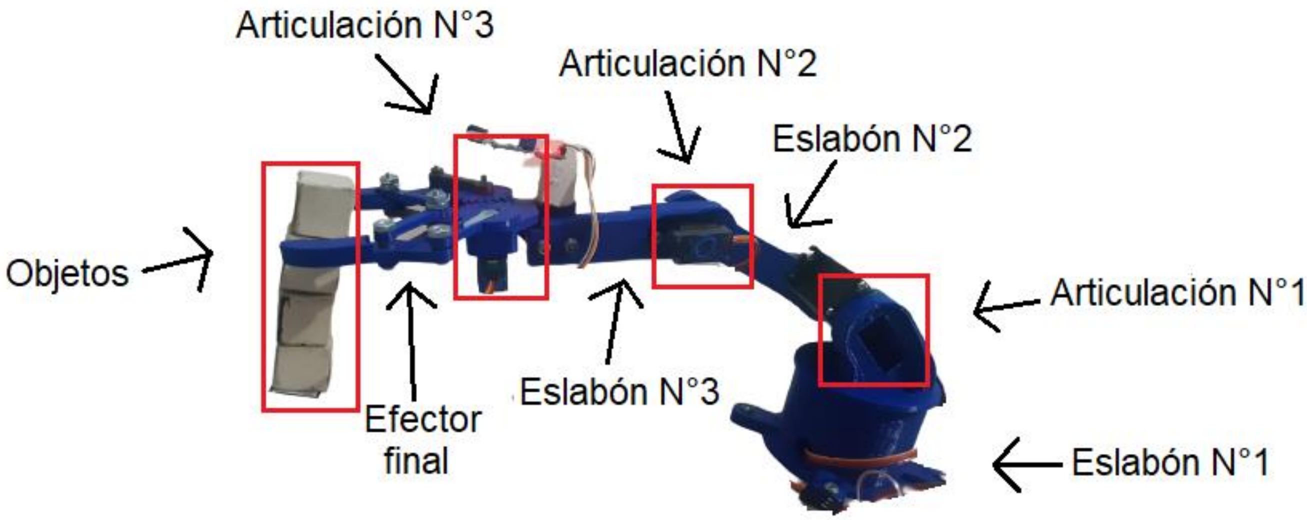
Figure 15. LCD display as a virtual terminal.

Source: own

The letter T is constantly sent by Matlab UART communication to the PIC to evaluate if the sensor that stops the robot has detected presence and perform the corresponding action. Then a number followed by a letter is observed, the number indicates the operation mode in which the robot is going to work (1 is automatic mode, 2 is manual mode and 3 is semi-automatic mode) and the letter (A, B, C or D) is the selected servomotor; the next three numbers are the angular value data which will be assigned to the previously selected servomotor and finally the letter 'Y' is understood by the microcontroller to perform the action that manages the rotation speed of each servomotor and vary it according to the active sensors.

Finally, Figure 16 shows the final result of the assembled robot, detailing the elements that make up the 3R robot, such as joints, links, end effector, sensors and the objects used to represent material handling.

Figure 16. Robotic arm 3R.



Source: own

5. Conclusions

The 3R robot integrated with presence sensors and interacting through a low cost microcontroller (PIC 18F45K22), demonstrated that it is a flexible solution for applications that require the manipulation of objects for the automation of a *pick and place* process; likewise, the simulation in Matlab in real time of the movement performed by the 3R robot, allowed to verify the trajectories executed by each mode of operation arbitrarily selected by the operator. In addition, the interaction between different devices and peripherals, through UART and I2C communication, capable of receiving and interpreting information to work together with the prototype of the robotic cell, ensuring a trajectory speed from point A to point B, was performed. The communication of the PIC with Matlab was done through UART communication at a speed of 9600 baud, which works correctly to transmit and receive information, since there is no data loss; however, due to the interpolation of the trajectory calculated in Matlab to generate a simultaneous movement in all joints, the system becomes a little slow, specifically between each movement of the robot with a *delay* of 500 ms. Finally, it was demonstrated through the movement of the 3R robot to the desired points, that the equations proposed for the inverse kinematics of the 3R robot, correctly represent the position and orientation of the end effector.

References

[1] R. V. V. Petrescu et al., "The Inverse Kinematics of the Plane System 2-3 in a Mechatronic MP2R System, by a Trigonometric Method," Journal of Mechatronics and Robotics, vol. 1, no. 2, pp. 75-87, 2017. https://doi.org/10.3844/jmrsp.2017.75.87

[2] S. P. Sethi et al., "Sequencing of parts and robot moves in a robotic cell," International Journal of Flexible Manufacturing Systems, vol. 4, no. 3-4, pp. 331-358, 1992. https://doi.org/10.1007/BF01324886

[3] J. Blazewicz et al., "Scheduling tasks and vehicles in a flexible manufacturing system," International Journal of Flexible Manufacturing Systems, vol. 4, pp. 5-16, 1991. https://doi.org/10.1007/BF01325094

[4] C. Deuerlein et al., "Improved design flexibility of open robot cells through tool-center-point monitoring," Procedia CIRP, vol. 100, pp. 295-300, 2021. https://doi.org/10.1016/j.procir.2021.05.069

[5] G. Veiga et al., "Experiments with service-oriented architectures for industrial robotic cells programming," Robotics and Computer-Integrated Manufacturing, vol. 25, no. 4-5, pp. 746-755, 2009. https://doi.org/10.1016/j.rcim.2008.09.001

[6] Q. Zhao et al., "Robotic Cell Rotation Based on the Minimum Rotation Force," IEEE Transactions on Automation Science and Engineering, vol. 12, no. 4, pp. 1504-1515, 2015. https://doi.org/10.1109/TASE.2014.2360220

[7] G. Michalos et al., "Design Considerations for Safe Human-robot Collaborative Workplaces," in: Understanding the life cycle implications of manufacturing, 2015, pp. 248-253. https://doi.org/10.1016/j.procir.2015.08.014

[8] E. Magrini et al., "Human-robot coexistence and interaction in open industrial cells," Journal of Robotics and Computer-Integrated Manufacturing, 2019, p. 101. https://doi.org/10.1016/j.rcim.2019.101846

[9] Digchip, "datasheet PCA9685PW," DigChip IC database, 2009. [Online]. Available: https://www.digchip.com/datasheets/parts/datasheet/364/PCA9685PW-pdf.php

[10] F. J. Zamora Navarro and A. Valiente Cristancho, "Tasa de muestreo ADC en microcontroladores avanzados de 8 bits," Visión electrónica, vol. 9, no. 1, pp. 128-138, 2015. https://doi.org/10.14483/22484728.11022

[11] E. García-Guerrero et al., "Randomness improvement of chaotic maps for image encryption in a wireless communication scheme using PIC-microcontroller via Zigbee channels," Chaos, Solitons & Fractals, vol. 133, p. 109646, 2020. https://doi.org/10.1016/j.chaos.2020.109646

[12] "I2C - Puerto, Introducción, trama y protocolo - HETPRO/TUTORIALES," HETPRO/TUTORIALES. [Online]. Available: https://hetpro-store.com/TUTORIALES/i2c/

[13] Z. Boric and B. Markovic, "The talking thermometer simulator based on the DS1820 sensor and PIC18F45K22 microcontroller," in: 2012 20th Telecommunications Forum (TELFOR), 2012, pp. 544-547. https://doi.org/10.1109/TELFOR.2012.6419268

[14] P. I. Corke, "A robotics toolbox for MATLAB," IEEE Robotics and Automation Magazine, vol. 3, no. 1, pp. 24-32, 1996. https://doi.org/10.1109/100.486658

[15] Y. Fang and X. Chen, "Design and Simulation of UART Serial Communication Module Based on VHDL," in: 2011 3rd International Workshop on Intelligent Systems and Applications, 2011, pp. 1-4. https://doi.org/10.1109/ISA.2011.5873448

[16] J. Calderón Acero and I. V. Parra Garzón, "Controladores difusos en microcontroladores: software para diseño e implementación," Visión electrónica, vol. 4, no. 2, pp. 64-76, 2010.

[17] A. D'Souza et al., "Learning inverse kinematics," Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Expanding the Societal Role of Robotics in the Next Millennium, 2001. https://doi.org/10.1109/IROS.2001.973374