



Diseño de un brazo robótico de 6 grados de libertad controlado mediante un microcontrolador PIC y comunicaciones seriales UART e I2C

**Daniel Santiago Caycedo Rivera
Erick Santiago Soto Barrantes**

Universidad ECCI
Escuela Colombiana de Carreras Industriales
Ingeniería Mecatrónica
Bogotá D.C.
2021

Diseño de un brazo robótico de 6 grados de libertad controlado mediante un microcontrolador PIC y comunicaciones seriales UART e I2C

Daniel Santiago Caycedo Rivera
Erick Santiago Soto Barrantes

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:

Ingeniero Mecatrónico

Director (a):

Fabian Barrera Prieto

Línea de Investigación:

Grupo de Investigación/Semillero:

Semillero GIDRA

Universidad ECCI

Escuela Colombiana de Carreras Industriales

Ingeniería Mecatrónica

Bogotá D.C.

2021

No basta tener buen ingenio lo principal es aplicarlo bien -Descartes

A nuestros padres, amigos y en especial a nuestro tutor quienes brindaron su apoyo y conocimiento, durante todo el desarrollo de este proyecto.

Agradecimientos

Agradecemos a nuestros padres quienes nos dieron la oportunidad de estudiar esta carrera y nos apoyaron en todo momento para poder completarla, a nuestros amigos que nos apoyaron incondicionalmente a lo largo de la carrera y del desarrollo del proyecto, en especial a aquellos que nos acompañaron en esas largas noches jugando y sacando unas cuantas risas cuando nos sentíamos bloqueados, a Henry quien nos ayudó en varias ocasiones con sus conocimientos para mejorar e implementar algunos detalles del proyecto. Le agradecemos especialmente a nuestro tutor Fabian Barrera, quien nos brindó los conocimientos para poder proceder con el proyecto, que nos dio los recursos necesarios y el tiempo para corregir los errores que salieron en el camino.

Resumen

Este documento muestra el desarrollo paso a paso de un robot controlado mediante un microcontrolador PIC 18F4550 y un controlador de servomotores PCA9685, los cuales reciben comandos mediante comunicación serial e I2C desde una interfaz de MATLAB establecida en un ordenador. Este sistema es complementado por periféricos para el control y visualización de variables relevantes para el usuario (pantalla GLCD, un teclado matricial y un Reloj en tiempo real). Inicialmente se desarrollan las comunicaciones entre el ordenador, el microcontrolador y la PCA 9685, seguido se realiza el diseño y estudios de la estructura del robot con 6 grados de libertad, con estos datos se aplican los conceptos de cinemática en MATLAB para adquirir los ángulos de posicionamiento de cada servomotor para llegar a un punto específico en el espacio. Finalmente se diseña una PCB en la herramienta de EasyEDA y facilitar el montaje del circuito y evitar errores por cableado.

Palabras clave: Robot 6DOF, Microcontrolador, I2C, UART.

Abstract

This document shows the step-by-step development of a robot controlled by a PIC 18F4550 microcontroller and a PCA9685 servo motor controller, which receive commands through serial communication and I2C from a MATLAB interface established on a computer. This system is complemented by peripherals for the control and visualization of relevant variables for the user (GLCD screen, a matrix keyboard and a Real-time Clock). Initially the communications between the computer, the microcontroller and the PCA 9685 are developed, followed by the design and studies of the robot structure with 6 degrees of freedom, with these data the concepts of kinematics are applied in MATLAB to acquire the angles of positioning of each servomotor to reach a specific point in space. Finally, a PCB is designed in the EasyEDA tool to facilitate the assembly of the circuit and avoid wiring errors.

Keywords: Robot 6DOF, Microcontroler, I2C, UART

Contenido

	Pág
Tabla de contenido	
1. Introducción.....	13
2. Marco teórico.....	17
2.1 Microcontrolador.....	17
2.2 Comunicación Serial UART.....	17
2.3 I2C.....	18
2.4 Pantalla LCD.....	18
2.5 Servo Motor.....	19
2.6 Robot.....	20
2.7 Cinemática.....	21
2.8 RTC (Real Time Clock).....	21
2.9 PCA9685.....	21
2.10 Baudios.....	22
3. Metodología de diseño.....	24
3.1 Planteamiento de funcionalidades del sistema.....	24
3.2 Diseño del circuito electrónico.....	24
3.3 Programación del microcontrolador.....	26
3.3.1 Configuración comunicación serial UART.....	26
3.3.2 Configuración comunicación serial I2C.....	27
3.3.3 Configuración PCA9685.....	28
3.3.4 Configuración RTC.....	28
3.3.5 Configuración y uso de la pantalla GLCD.....	29
3.3.6 Configuración de teclado matricial.....	30
3.4 Diseño CAD del robot de 6 DOF.....	30
3.5 Aplicación de cinemática inversa en MatLab.....	34
3.6 Diseño de algoritmo de envío, recepción y procesamiento de datos.....	34
3.7 Diseño de interfaz de Matlab.....	38
3.8 Diseño de PCB.....	38
4. Resultados.....	40
4.1 Envío y recepción de datos en el microcontrolador.....	40
4.2 Interfaz gráfica.....	40
4.3 PCB y montaje del circuito.....	42
4.4 Robot.....	43
5. Capítulo (...)	¡Error! Marcador no definido.
6. Conclusiones y recomendaciones.....	52
6.1 Conclusiones.....	52
6.2 Recomendaciones.....	53

Lista de figuras

Figura 1 Microcontrolador PIC.....	17
Figura 2 Protocolo de comunicación serial.....	18
Figura 3 Bus I2C.....	18
Figura 4 Pantalla GLCD.....	19
Figura 5 PWM aplicado a servomotor.....	20
Figura 6 Brazo robótico.....	20
Figura 7 RTC.....	21
Figura 8 PCA9685.....	22
Figura 9 Modulación por ancho de pulso PWM.....	22
Figura 10 Periodo de una onda.....	23
Figura 11 Esquema de comunicaciones del sistema.....	24
Figura 12 Diagrama de conexión I2C.....	25
Figura 13 Conexión del teclado matricial.....	26
Figura 14 Distribución pantalla GLCD.....	29
Figura 15 Diseño CAD.....	31
Figura 16 Planteamiento de fuerzas estático en el diseño CAD.....	32
Figura 17 Espacio de Trabajo.....	34
Figura 18 Lógica de programación MATLAB.....	36
Figura 19 Lógica de programación PIC.....	37
Figura 20 Esquema de conexiones PCB.....	38
Figura 21 Envío y recepción de datos.....	40
Figura 22 Interfaz MATLAB.....	41
Figura 23 Comunicacion interfaz MATLAB con puerto serial.....	42
Figura 24 PCB.....	43
Figura 25 Estructura mecánica.....	46
Figura 26 Posición 0.....	47
Figura 27 Posición A.....	48
Figura 28 Posición elevada.....	49
Figura 29 Posición B.....	50
Figura 30 Posición B con Gripper abierto.....	50

Lista de tablas

	Pág.
Tabla 1: Peso de los componentes de la estructura mecánica.....	31
Tabla 2 restricciones de los servomotores	33
Tabla 3 Parámetros Denavit-Hartenberg	33

1. Introducción

1.1 Antecedentes

El desarrollo de las tecnologías en los últimos años ha creado la necesidad de conocer acerca de distintas tecnologías para implementar y diseñar dispositivos de automatización. Para el desarrollo de estas maquinarias se requiere realizar diseños mecánicos, electrónicos y de software capaces de interactuar entre si con el fin de realizar distintos procesos de forma óptima.

Es así como es necesario escoger y/o diseñar las estructuras mecánicas que van a realizar físicamente los procesos, los componentes electrónicos que van a procesar distintos tipos de datos y señales de entrada y así controlar los actuadores, interfaces gráficas, enviar datos, entre otros. Adicionalmente se realizan algoritmos de programación con la lógica de operación del sistema que se está automatizando.

1.2 Descripción del problema

1.2.1 Pregunta problema

¿Se puede realizar el control de un robot de 6 grados de libertad con un microcontrolador PIC18F4550, un computador y distintos componentes electrónicos, con el fin de mostrar sus capacidades y limitaciones?

1.2.2 Formulación del problema

En las áreas de ingeniería y desarrollo tecnológico se presenta la necesidad de realizar una gran variedad de maquinarias capaces de optimizar los procesos en las empresas, esto se realiza teniendo en cuenta limitaciones económicas y de tiempo.

Empezando por el diseño mecánico, se requiere de un estudio estático y cinemático para conocer las fuerzas que debe tolerar cada una de las partes, así como la fuerza que debe aplicar cada actuador durante el proceso. Para el diseño electrónico se necesita escoger adecuadamente un microcontrolador o sistema embebido capaz de realizar el procesamiento de entradas digitales y análogas sin bloquearse durante la ejecución del proceso, así mismo si es necesario implementar módulos externos se debe analizar la implementación de los distintos protocolos de comunicación que usen estos módulos. Finalmente se realiza la

programación de los controladores de los sistemas embebidos implementados o una computadora en caso de procesos con alta complejidad, para esto es necesario realizar un algoritmo para procesar las variables del sistema con efectividad y finalmente realizar las operaciones necesarias.

Esto requiere que los ingenieros conozcan de las distintas tecnologías para realizar las distintas partes del diseño con el fin de no derrochar los recursos dispuestos y no realizar maquinas que resulten con problemas de eficiencia o inservibles. Esto se puede observar desde tres enfoques; la estructura mecánica, el circuito electrónico y la programación de software.

1.3 Justificación de la investigación

En este proyecto se va a diseñar y construir un robot de 6 grados de libertad con un sistema embebido basado en un microcontrolador PIC18F4550 para el control de los servomotores y ciertos módulos adicionales complementando el funcionamiento y características del sistema. Así mismo se realiza una interfaz en un equipo de cómputo por medio de la aplicación de Matlab con el fin de poder controlar el robot por medio de una interfaz intuitiva.

El diseño de cada una de las partes se realiza buscando métodos que faciliten el desarrollo y permitan optimizar el funcionamiento de cada uno de los sistemas, se empieza por una estructura mecánica genérica, componentes electrónicos con capacidades específicas para adicionar de forma modular y librerías dentro de la programación aprovechando las funcionalidades que traen integradas.

1.4 Alcance y delimitación

Para este proyecto se estipulan ciertas limitaciones desde el enfoque de los diseños mecánico, electrónico, programación del microcontrolador y programación de la interfaz gráfica.

1.4.1 Diseño Mecánico

La estructura mecánica se desarrollará basada en servomotores genéricos de bajo costo y con la capacidad de levantar un peso estipulado que permita demostrar su funcionamiento en ciertas aplicaciones. Así mismo se adquieren piezas realizadas específicamente para los motores escogidos.

1.4.2 Diseño Electrónico

Se parte de las limitaciones y capacidades técnicas del microcontrolador PIC18F4550, se tiene en cuenta la cantidad de entradas y salidas digitales, no se van a utilizar las entradas y salidas análogas. Dentro de los pines digitales se apartan aquellos que se necesiten para las comunicaciones seriales I2C y RS232 debido a que se van a utilizar para enviar y recibir datos a los módulos y el computador.

Finalmente se escogen los componentes y módulos periféricos necesarios, un modulo conversor de comunicación RS232 a USB, el driver modulador de pulsos PWM para servomotores PCA9685, una pantalla LCD graficadora, un reloj en tiempo real y un teclado matricial.

1.4.3 Programación del microcontrolador

El microcontrolador se encargará exclusivamente de tres funcionalidades; la primera es la recepción de una trama datos por comunicación serial RS232 y extraer los valores de los ángulos de los servomotores y las coordenadas en los ejes 'x', 'y' y 'z'. La segunda es la adquisición de la hora del reloj en tiempo real e imprimir dichos datos junto a las coordenadas en la pantalla LCD. Por último, el PIC18F4550 se encarga de leer el teclado matricial y en caso de presionar una tecla se realizará una acción programada según la aplicación que se le quiera dar al robot.

1.4.4 Programación de la interfaz grafica

La interfaz gráfica debe permitir la visualización la posición del robot y sus valores angulares en cada instante, también debe permitir el ingreso de coordenadas para mover el robot y establecer configuraciones adicionales que faciliten su manejo, estas herramientas indican pueden indicar al robot la apertura o cierre del gripper, regreso a su posición inicial y la ejecución de una secuencia programada y definida previamente.

1.5 Objetivos

1.5.1 Objetivo general

Diseñar un brazo robótico de 6 grados de libertad controlado mediante un microcontrolador PIC y comunicaciones seriales UART e I2C

1.5.2 Objetivos específicos

1. Diseñar un robot de 6 grados de libertad por medio de su modelo CAD y determinar los parámetros Denavit-Hartenberg
2. Diseñar una interfaz de usuario para observar en un gráfico la localización del brazo robótico de 6 DOF que permita mover el robot a posiciones específicas.
3. Diseñar un circuito electrónico para adquirir datos y controlar los distintos periféricos del proyectog
4. Realizar pruebas de diferentes secuencias (Pick and place) y posiciones con el brazo robótico de 6 grados de libertad

2. Marco teórico

2.1 Microcontrolador

Los microcontroladores son sistemas compuestos por al menos un microprocesador, una memoria programable, una memoria para datos y entradas y salidas, pueden ser análogas o digitales. Algunos pueden contener más características como temporizadores, contadores, salidas y entradas especiales, estas para algún uso específico, por ejemplo, para comunicarse mediante un protocolo serial o i2c. estos dispositivos son usados en gran cantidad de electrodomésticos y se encuentran principalmente dentro de los sistemas que controlan, razón por la cual también se les conoce como embebidos. [1] véase Figura 1

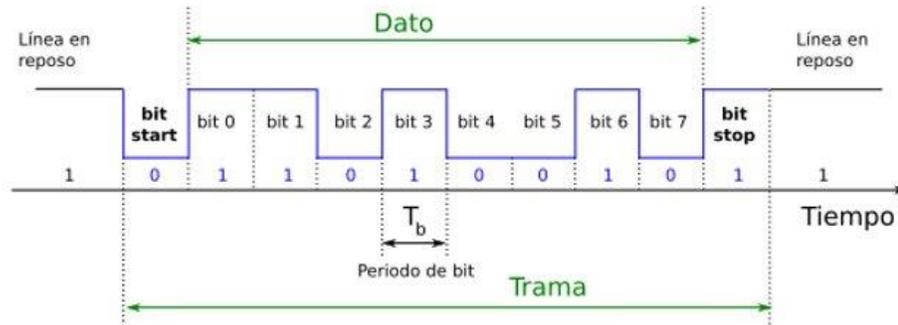
Figura 1 Microcontrolador PIC



2.2 Comunicación Serial UART

La comunicación transmisor receptor asíncrona universal (UART) por sus siglas en inglés es un tipo de comunicación serial usada en su mayoría para comunicar computadores y periféricos, se caracteriza por ser de corta distancia, baja velocidad y bajo costo. utiliza únicamente un transmisor y receptor sin establecer jerarquía entre las partes, y pudiendo ambas recibir y enviar información. Mediante este tipo de comunicación se envían de 5 a 9 bits de información, aunque suelen utilizarse 8 en la mayoría de los casos. La comunicación se conoce como asíncrona debido a que ambas partes carecen de un reloj por lo que se debe establecer la misma velocidad de transmisión y recepción de datos en ambas partes, la velocidad corresponde a la cantidad de bits enviados por segundo, también conocida como Baudios. [2] véase Figura 2

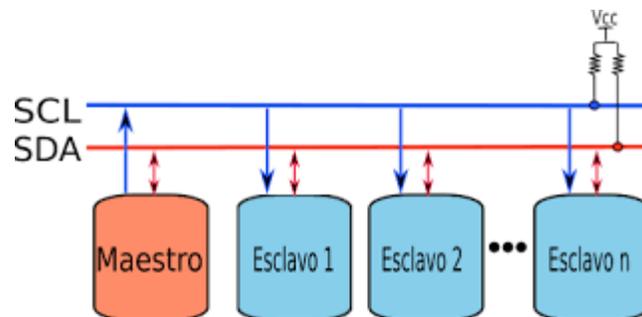
Figura 2 Protocolo de comunicación serial



2.3 I2C

El bus I2C es un estándar de comunicación diseñado por la empresa Philips Semiconductors a principios de los 80 para comunicar circuitos integrados, para ello se utiliza una señal de reloj común para todos los dispositivos conectados y se establece una jerarquía de maestro-esclavo donde el maestro solicita información a los elementos esclavos, cada esclavo contiene una dirección o identificación distinta a la de los demás dispositivos por lo que se establece comunicación con un dispositivo a la vez. En este protocolo los esclavos cuentan con registros direccionados, por lo que el maestro puede leer o reescribir dichos registros para configurar el dispositivo o adquirir información. [3] véase Figura 3

Figura 3 Bus I2C



2.4 Pantalla LCD

Una pantalla LCD o Liquid Crystal Display, por sus siglas en inglés es un dispositivo que utiliza la electricidad para cambiar la posición de pequeños cristales y permiten el paso de luz haciendo posible conformar patrones o formas que interpretamos como información de algún tipo, ya sean palabras, números imágenes o varios de estos elementos. [4]

Estas pantallas poseen distintos tamaños y controladores que modifican su operación, en el caso específico de la pantalla 128x64, utiliza 2 entradas de control CS1 y CS2 que habilitan la escritura en la respectiva mitad de la pantalla, cada mitad consta de 64x64 y estas a su vez se subdividen en 8 filas de 8 pixeles como se evidencia en la **¡Error! No se encuentra el origen de la referencia.:**

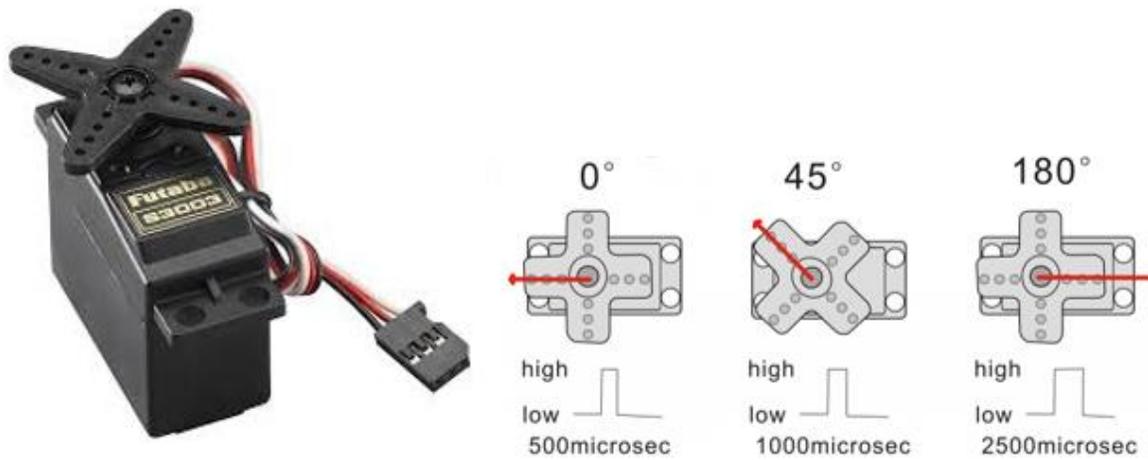
Figura 4 Pantalla GLCD



2.5 Servo Motor

Los servomotores son un tipo de motores que cuentan con un controlador que permite conocer y cambiar su posición angular, esto mediante un rango de frecuencia manipulado a través de un PWM en el cual el ciclo útil determina el ángulo que alcanza el dispositivo, gracias a esto son ampliamente utilizados en robótica, y aplicaciones que requieran control de posición. [5] véase Figura 5

Figura 5 PWM aplicado a servomotor



2.6 Robot

Un robot es un dispositivo con 2 o más grados de libertad que debe ser controlado y programable para realizar alguna tarea en específico, la mayoría de robot industriales son brazos robóticos que realizan operaciones como soldadura, pintura, aunque mayormente operaciones de Pick & Place, ya sea a pequeña o gran escala. [6] véase **¡Error! No se encuentra el origen de la referencia..**

Figura 6 Brazo robótico



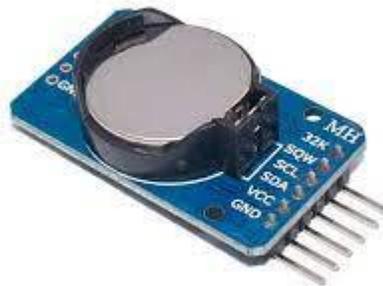
2.7 Cinemática

La cinemática como parte de la mecánica se encarga del estudio de los movimientos de un cuerpo en el espacio, en el caso de la robótica, describe el comportamiento de cada eslabón y articulación que compone un robot y como esto delimita el movimiento que es capaz de realizar. [7]

2.8 RTC (Real Time Clock)

El RTC o Real Time Clock por sus siglas en ingles es un dispositivo utilizado en el control de tiempo, puede comunicarse utilizando el protocolo I2C para configurar una hora y fecha que luego serán consultadas por un dispositivo maestro, además es capaz de mantener el conteo temporal por hasta 10 años sin estar conectado a una fuente de energía. [8] véase Figura 7

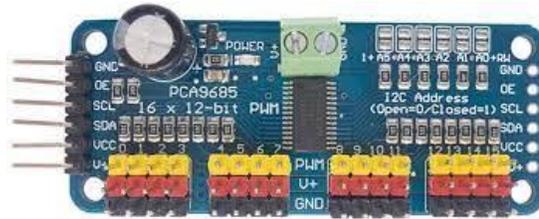
Figura 7 RTC



2.9 PCA9685

El controlador PCA9685 es un dispositivo que permite controlar hasta 16 señales de PWM por medio de comunicación I2C, esto permite ahorrar entradas y salidas de un sistema embebido, también permite utilizar una fuente externa al circuito principal de forma que no interfiera en caso de requerir mayor energía de la que se le podría proporcionar mediante el circuito base. [9] véase Figura 8

Figura 8 PCA9685



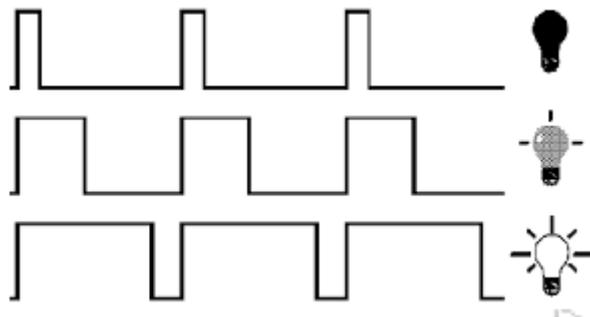
2.10 Baudios

La tasa de baudios de un sistema hace referencia a la cantidad de Bits por segundo que se envían cuando se utiliza un protocolo de comunicación, la velocidad de los baudios se ve afectada por la distancia por lo que se debe considerar este factor antes de ajustar una velocidad específica de baudios en el sistema.[10]

2.11 PWM

La modulación por ancho de pulso o PWM por sus siglas en inglés (Pulse Wide Modulation) es una señal digital que controla la activación de un dispositivo utilizando la frecuencia de una onda en sus ciclos de trabajo positivo y negativo [11] de la siguiente manera: véase Figura 9

Figura 9 Modulación por ancho de pulso PWM



Durante un periodo de la onda, el tiempo que esta en alta en proporción a la duración total se conoce como ciclo de trabajo, en el caso de la primera línea se observa que corresponde a uno bajo por lo que el dispositivo que controla en este caso un LED no emite luz, mientras que en la segunda imagen con un ciclo de trabajo mayor el LED emite luz, pero no una muy brillante, finalmente con un ciclo de trabajo alto se observa como el LED emite una luz brillante. Según el dispositivo a utilizar la frecuencia y el ciclo de trabajo determinan alguna característica del elemento, en el caso del LED su intensidad, en el caso de un servomotor controla la posición.

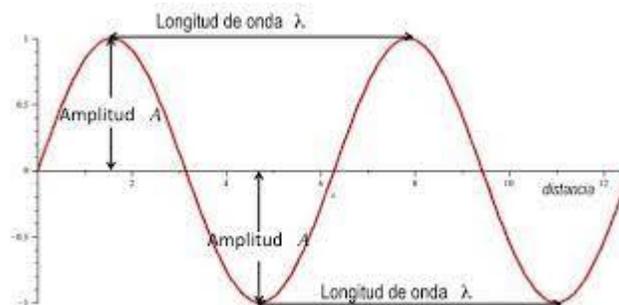
2.12 Frecuencia

La frecuencia se conoce como la cantidad de veces que se repite un evento periódico o repetitivo en una unidad de tiempo, nombrada esta cantidad en Hertzios (Hz) para el caso de la frecuencia temporal.[12]

2.13 Periodo

El periodo de una onda corresponde al tiempo que tarda en completar un ciclo, es por esto que se conoce como el inverso de la frecuencia.[12] véase Figura 10

Figura 10 Periodo de una onda

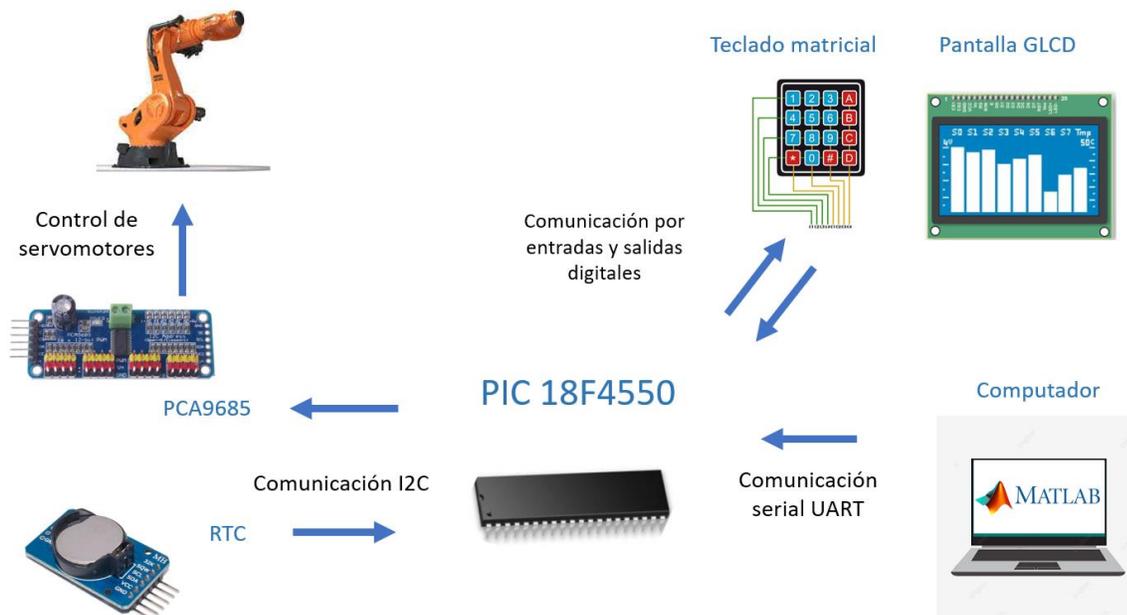


3. Metodología de diseño.

3.1 Planteamiento de funcionalidades del sistema.

Para empezar con el desarrollo del robot de 6 grados de libertad se realiza un esquema (Véase Figura 11) con los componentes que van a intervenir en el funcionamiento del sistema y se representa el sentido del envío y recepción de datos entre los componentes. El PIC18f4550 es el dispositivo central en la comunicación, primero recibe datos desde un computador donde se ingresan y procesan los datos de las coordenadas del efector final y los ángulos de posición de los servomotores del robot, con estos datos el microcontrolador se encarga de imprimir las coordenadas 'x', 'y' y 'z' en la pantalla GLCD y enviar los ángulos al módulo PCA9685 que tiene la función de mover los servomotores a la posición indicada. Adicionalmente se dispone de un reloj en tiempo real (RTC) para que el circuito también imprima la hora y tenga este dato actualizado en caso de que el sistema se encuentre desconectado, adicionalmente se tiene un teclado matricial 4x4 para ciertas acciones que se desean hacer por medio de accionamientos manuales.

Figura 11 Esquema de comunicaciones del sistema



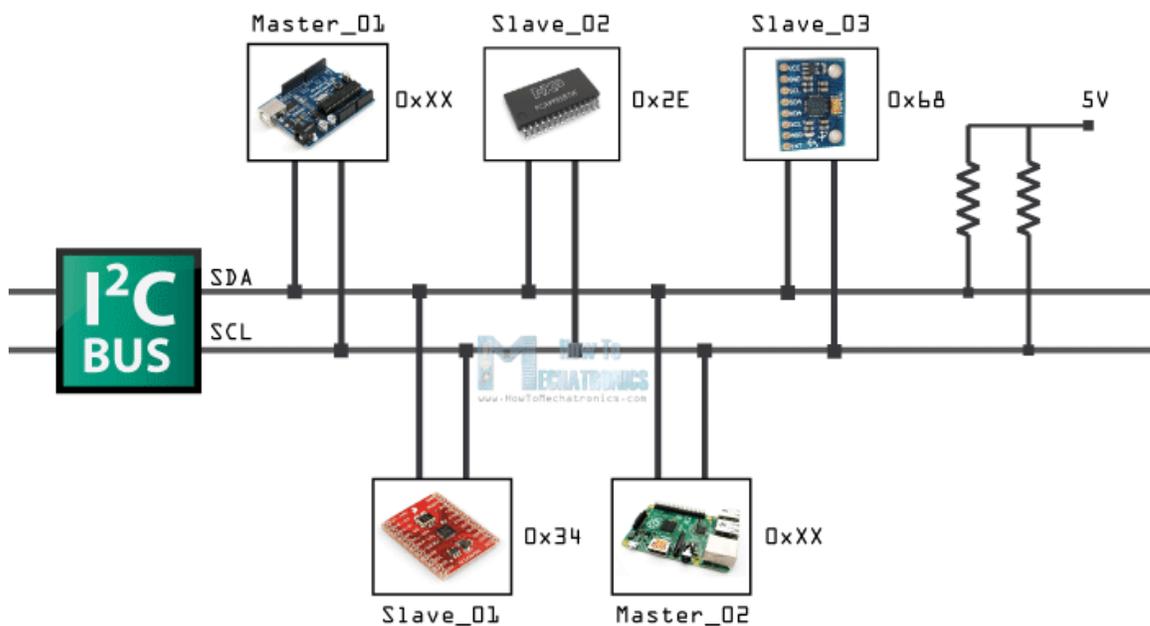
3.2 Diseño del circuito electrónico

Antes de empezar con la programación del microcontrolador es necesario realizar un cableado que permita el uso de los módulos y los componentes electrónicos involucrados en

el funcionamiento del sistema y poder realizar pruebas a lo largo del desarrollo, este montaje se va a hacer inicialmente en protoboard. Para empezar, se realizan las conexiones para energizar el circuito y el microcontrolador, adicionalmente se realiza la conexión del pin MCLR a una resistencia de $10K\Omega$ y a V_{cc} como forma de protección para cualquier reinicio o cambio de señal inesperada.

La conexión de los controladores que funcionan por comunicación I2C se realiza tomando los pines SDA y SCL de cada uno y conectándolos a los pines 33 (SDA) y 34 (SCL) del microcontrolador como se puede observar en la Figura 12.

Figura 12 Diagrama de conexión I2C



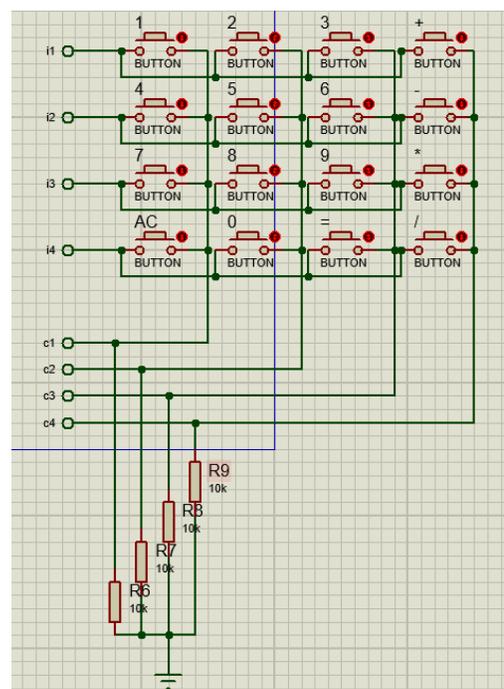
Adicionalmente se realiza la conexión del convertor de comunicación serial RS232 a USB, el módulo convertor tiene un pin de transmisión (TX) y un pin de recepción (RX), la conexión se realiza de forma cruzada con los pines del PIC18F4550; 26 (RX) y 25 (TX) con el fin que los transmisores estén conectados al receptor del otro dispositivo.

El cableado de la pantalla GLCD se inicia realizando las conexiones para energizar el microcontrolador interno de la pantalla y los LEDs del fondo de luz de la pantalla, se conectan los pines de V_{dd} , V_{ee} y A (ánodo) a una señal de 5V y los pines de V_{ss} , RST y K (cátodo) a GND. Para el control del contraste de la pantalla se conecta un potenciómetro al pin V_0 y también se conecta el pin RW a GND para que la pantalla este en estado de escritura (write) y no de lectura (read).

Para el control de la pantalla se conectan los pines D0 a D7 en el puerto D del PIC18F4550 para enviar los datos, el pin E (enable) al pin RE0 para habilitar el envío de datos, el pin RS (register select) al pin R1 para alternar entre el envío de datos y la configuración de la pantalla, finalmente se conectan los pines CS1 y CS2 a los pines RC0 y RC1 respectivamente para el control de la impresión del lado izquierdo y derecho de la pantalla.

Finalmente se realiza la conexión del teclado matricial para ello se conectan 4 de las terminales a los 4 primeros pines del puerto A (RA0 – RA3) y las otras 4 se conectan cada una a una resistencia pull down de 10k Ω y a los últimos 4 pines del puerto B (RB4 – RB7) debido a que es necesario utilizar la interrupción por cambio de estado para el funcionamiento del teclado matricial, en la Figura 13 se observa la conexión.

Figura 13 Conexión del teclado matricial



3.3 Programación del microcontrolador.

3.3.1 Configuración comunicación serial UART

Para realizar la configuración entre el PIC 18f4550 y un computador por medio del protocolo de comunicación UART RS232 se empieza revisando la hoja de datos del microcontrolador para saber que registros necesitan ser configurados. Primero se habilitan los pines de entrada y salida por donde se va a realizar el envío de datos, luego se configuran los registros RCSTA y TXSTA para habilitar el periférico de comunicación de recepción y transmisión

respectivamente teniendo en cuenta que se va a realizar una comunicación asíncrona y de 8 bits. Por último, se configura el registro SPBRG utilizando la

(1 y teniendo en cuenta que la tasa de baudios que se va a utilizar, en este caso se usa el estándar 9600.

$$SPBRG = \frac{F OSC}{DESIRED BAUD RATE * 64} \quad (1)$$

Con la configuración de la comunicación serial se crean 2 funciones para controlar el envío y recepción de los datos dentro de la ejecución del código. Primero se crea la función “receive” como interrupción principal teniendo en cuenta que el funcionamiento principal del sistema depende del envío de datos del computador donde este ejecutándose MATLAB. Para ello se configura el registro RCIE que habilita la interrupción por recepción de datos de comunicación serial UART y el registro RCIP que se utiliza para que esta interrupción tenga mayor prioridad ante cualquier otro proceso que se ejecute.

Adicionalmente se crea una función llamada start que se utiliza para enviar datos hacia el computador con el fin de poder realizar un control de errores durante la ejecución del programa.

3.3.2 Configuración comunicación serial I2C

Para poder configurar y utilizar los periféricos que utilizan la comunicación serial I2C se configuran los registros necesarios para el uso del módulo de comunicación I2C y adicionalmente se crea una serie de funciones para poder optimizar y asegurar el envío de datos. Inicialmente se configuran los registros de control SSPSTAT, SSPCON1 y SSPCON2 que inician el módulo para el protocolo de comunicación y lo colocan en modo maestro debido a que el PIC18F4550 va a ser el encargado de controlar todos los dispositivos periféricos que utilizan dicha comunicación. Finalmente se calcula el valor del registro SSPADD que permite configurar la velocidad de la comunicación por medio de la

(2 teniendo en cuenta que se utiliza una ratio de baudios de 100 KHz.

$$SSPADD = \frac{F OSC}{4 * BAUD RATE} - 1 \quad (2)$$

Primero se declaran dos funciones, writeI2C para el envío de datos y readI2C para la recepción. En la función writeI2C se tienen 4 parámetros de entrada; la dirección I2C del dispositivo a donde se van a enviar los datos, el registro que se quiere afectar dentro del dispositivo periférico, un arreglo con el o los datos que se quieren enviar y finalmente un valor con la cantidad de bytes que se quieren enviar. Para complementar el envío de datos se declara la función sendI2C donde se coloca el valor que se quiere enviar en el buffer

SSPBUF. Para la función readI2C se tienen los mismos parámetros con la diferencia que se solicita al componente periférico la cantidad de datos del cuarto parámetro de entrada y se guardan en el arreglo indicado por el tercer parámetro.

Adicionalmente se declaran otras dos funciones; startI2C y stopI2C, respectivamente se encargan de iniciar y finalizar el proceso de comunicación serial I2C. Finalmente se declaran las funciones repeatedStart y readyI2C para verificar que los buffers de comunicación estén disponibles para el envío y recepción de datos.

3.3.3 Configuración PCA9685

Para poder utilizar adecuadamente el módulo PCA9685 es necesario realizar unas configuraciones iniciales acerca del modo de trabajo y de las frecuencias de trabajo del controlador interno que tiene integrado, así como la frecuencia del pulso que se va a utilizar para los servomotores. Primero se configura la frecuencia de trabajo de los servomotores, para ello se revisa la hoja de datos del controlador de servomotores para saber que registros se tienen que afectar en dicha configuración y también se revisa las hojas de datos de los servomotores genéricos que se van a utilizar, se toma el ancho de pulso de trabajo de los servomotores, el periodo es de 20ms y el tiempo de duración del pulso efectivo se maneja entre mínimo 1ms y máximo 2ms que equivalen respectivamente al valor mínimo y máximo que alcanza el servomotor.

Teniendo en cuenta las frecuencias de trabajo se aplica la (3) y tomando la frecuencia de oscilación como 25MHz que es la frecuencia de trabajo del módulo PCA9685.

$$Prescale\ Value = round\left(\frac{OSC\ Clock}{4096 * Update\ Rate}\right) \quad (3)$$

Finalmente se crea una función map para escalar el valor de los ángulos de posicionamiento de los servomotores con el valor correspondiente para el registro del módulo. Con este dato se envía a los registros LEDx_ON_L cambiando el valor de la x por el número del servomotor que se quiere mover.

3.3.4 Configuración RTC

Para configurar el módulo de tiempo real se tiene en cuenta que se va a utilizar la hora en un formato de 24 horas y no se va a utilizar la función de la fecha. Parte de las características del uso del reloj en tiempo real es que se configura una primera vez configurando la hora y el formato, de ahí en adelante solo es necesario leer los datos de la fecha y hora dependiendo de la aplicación que se le dé.

arreglo específico en un nivel especificado por los parámetros de la función. Adicionalmente se crea una función que imprima una imagen digitalizada con anterioridad, permitiendo imprimir logos o cualquier tipo de imagen que la resolución permita imprimir.

Para realizar la impresión de la pantalla se crea una función interrupción que realice la impresión de los datos cada cierto tiempo, se utiliza el timer 1 que tiene incorporado el microcontrolador asegurando la impresión en un tiempo específico. En las configuraciones iniciales se coloca el registro TMR1IE en el valor lógico 1 para habilitar la interrupción y el registro TMR1IP en 0 para que la interrupción sea de baja prioridad. Por último, se crea una función interrupción con el parámetro `low_priority` indicando que las interrupciones manejadas dentro de la misma tienen baja prioridad, adicionalmente se coloca un condicional `if` con el registro TMR1IF que maneja la bandera de esta interrupción.

3.3.6 Configuración de teclado matricial

Para el teclado es necesario habilitar la interrupción de cambio de estado en el puerto B, para ello se coloca el registro RBIE en 1. Adicionalmente se configura el registro RBIP en 0 para que la prioridad sea baja a la hora de activarse y no interfiera con la interrupción de comunicación serial que tiene mayor prioridad. Para el funcionamiento de la interrupción se coloca un condicional en la función con el indicativo `low_priority`, el activador va a ser el registro RBIF con la bandera de la interrupción del cambio de estado en el puerto B.

Finalmente se realiza una función llamada `key` con un ciclo `for` que realiza un barrido a lo largo de los puertos que se utilizan en el teclado y obteniendo la tecla que se presionó y activó la interrupción, este dato se retorna para poder utilizarlo en la función interrupción donde se realizan los procesos destinados para cada tecla.

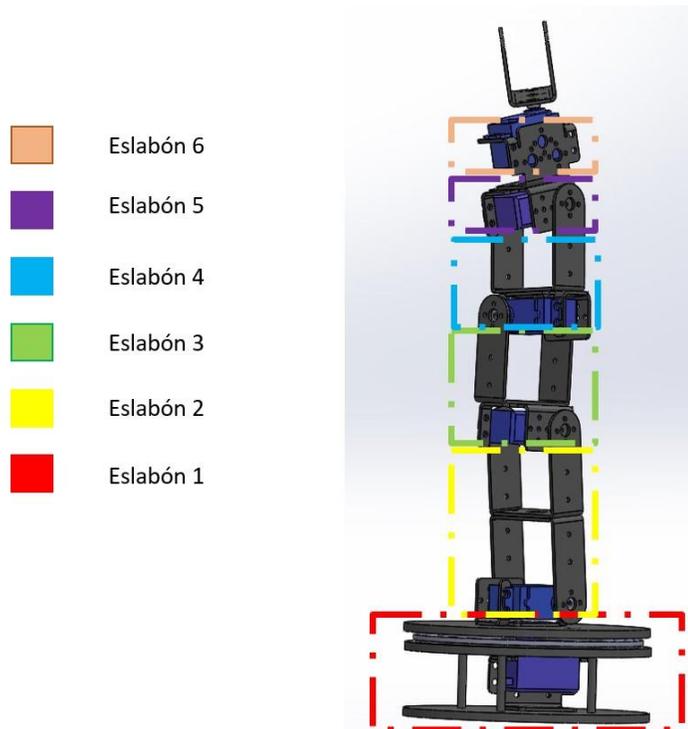
Teniendo la configuración finalizada se asignan inicialmente 3 teclas para configurar la hora, los minutos y los segundos del RTC en caso de que se descargue la batería del mismo o sea cambiado. Se configura una tecla para colocar el robot en el punto 0 en caso de que se requiera forzar dicha posición, adicionalmente se agrega una función para realizar una operación de Pick and Place entre 2 puntos específicos.

3.4 Diseño CAD del robot de 6 DOF

Para el diseño del robot de 6 grados de libertad se inicia planteando las limitaciones y alcances que va a tener la estructura, se parte de una arquitectura con 6 actuadores de tipo rotacional y se opta por el uso de servomotores y partes genéricas para su construcción facilitando la compra de materiales y el ensamble de la estructura. Se escogieron Brackets multipropósito de aluminio para los eslabones debido a la practicidad que presentan para construir robots de este tipo, adicional se consigue una base giratoria con rodamiento ya

diseñada para disminuir la carga de la primera junta, teniendo esto en cuenta se realiza un modelo en CAD de las piezas y se ensamblan para tener un diseño claro de la estructura del robot (Ver Figura 15).

Figura 15 Diseño CAD



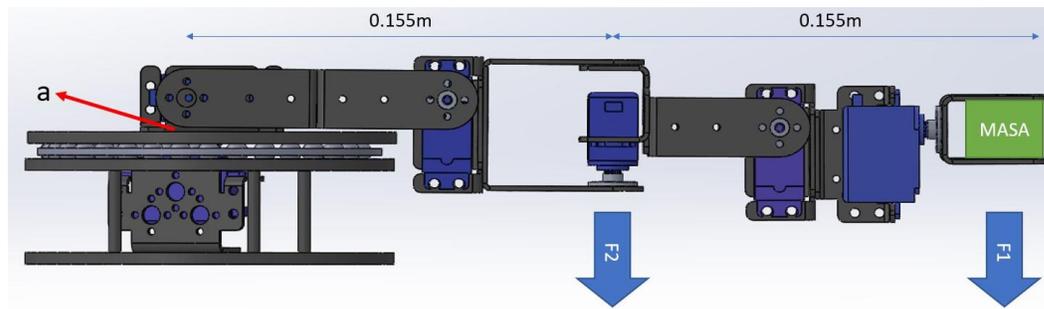
Inicialmente se plantea que el robot pueda soportar una carga de 0.5 Kg, con esto se realizaron los cálculos necesarios para escoger el torque requerido en cada uno de los actuadores para poder mover todas las juntas sin presentar dificultad con una carga del peso escogido. Se tiene en cuenta el peso promedio de un servomotor estándar y de los Brackets en aluminio que se van a utilizar ver tabla 1.

Tabla 1: Peso de los componentes de la estructura mecánica.

Elemento	Peso en gramos
Servomotor	71
Brackets servomotor	15
Brackets U	25
Gripper	68

En este proceso se tomó en cuenta el peso del objeto a cargar (F_1) y el peso de los eslabones y articulaciones del robot (F_2), así como las distancias de cada una de estas fuerzas al eslabón de mayor esfuerzo, en este caso el eslabón N°1 y específicamente en su eje de rotación (a) véase figura Figura 16.

Figura 16 Planteamiento de fuerzas estático en el diseño CAD



Partiendo del planteamiento anterior se aplican las ecuaciones 4 y 5.

$$F_1 = M * g = 9.8 \text{ m/s}^2 * 0.5 \text{ Kg}$$

(4)

$$F_1 = 4.9 \text{ N}$$

(5)

Se realiza el mismo procedimiento para F_2 que se ubica en el centro de masa del robot, para ello se aplican las ecuaciones

$$F_1 = M * g = 9.8 \text{ m/s}^2 * 1.03 \text{ Kg}$$

(6)

$$F_1 = 10.094 \text{ N}$$

(7)

Luego se obtiene la distancia entre la F_1 y el punto de rotación a, de igual manera para F_2 . seguido se obtiene el momento para cada fuerza, esto con el producto cruz de los vectores fuerza X distancia ver ecuaciones

(8,

(9,

(10 y

(11.

$$M1 = F1 \times D = [0.31i \ 0j \ 0k] \times [0 \ 0 \ 4.9] \quad (8)$$

$$M1 = 1.519 \text{ N} * m \quad (9)$$

$$M2 = F2 \times D = [0.155i \ 0j \ 0k] \times [0 \ 0 \ 10.094] \quad (10)$$

$$M2 = 1.514 \text{ N} * m \quad (11)$$

Finalmente se suman los 2 momentos para obtener la fuerza necesaria para el actuador en ese punto ver ecuaciones (12 y (13.

$$M1 + M2 = 1.519 \text{ N} * m + 1.514 \text{ N} * m \quad (12)$$

$$M1 + M2 = 3.083 \text{ N} * m = 30.93 \text{ Kg} f * cm \quad (13)$$

Con los cálculos realizados se buscan los motores con las características que se evidencian en la **¡Error! No se encuentra el origen de la referencia..**

Después de analizar la estructura diseñada y la capacidad de giro de los servomotores se establecen límites de forma que no interfieran con la estructura durante el movimiento, las restricciones de cada servomotor se muestran en la Tabla 2 restricciones de los servomotores

Tabla 2 restricciones de los servomotores

Servo	Rango	Torque (Kg/cm)
1	0 – 180°	35
2	0 – 180°	35
3	0 – 180°	20
4	0 – 120°	20
5	0 – 120°	20
6	0 – 180°	20

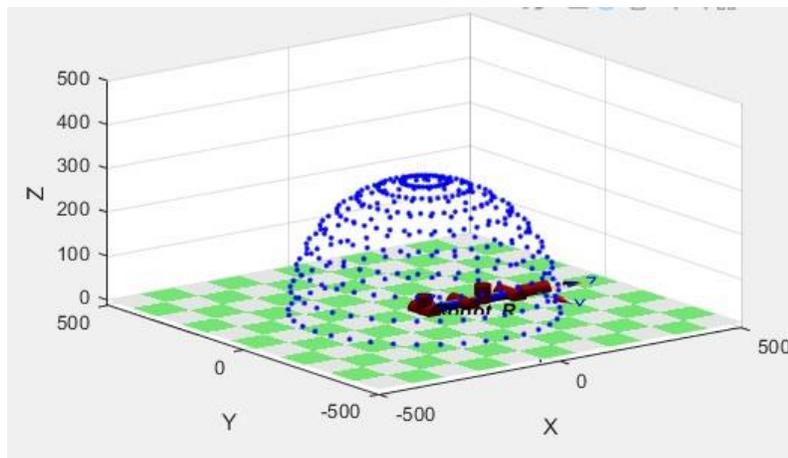
Gracias al diseño CAD también se conocen las medidas del robot y con ello el espacio de trabajo que corresponde a una semiesfera de radio de 329.09mm como se observa en la Figura 17, y su configuración en los parámetros D&H modificados presentados en la **¡Error! No se encuentra el origen de la referencia..**

Tabla 3 Parámetros Denavit-Hartenberg

DH	α	a	d	Offset
Eslabón 1	0°	0	0	0°
Eslabón 2	90°	0	0	0°

Eslabón 3	0	106	0	-90°
Eslabón 4	90°	54.63	0	-60°
Eslabón 5	-90°	74.68	0	-150°
Eslabón 6	-90°	0	64	0

Figura 17 Espacio de Trabajo



3.5 Aplicación de cinemática inversa en MATLAB.

Conociendo las medidas del robot se empieza a modelar su estructura por medio de los parámetros Denavit-Hartenberg y se utilizan dichos datos en las funciones y herramientas que presenta el toolbox de Peter Corke, gracias a estas herramientas MATLAB es capaz de calcular la cinemática inversa y proporcionar la matriz de posición para una coordenada especificada. Teniendo esto en cuenta se ingresan las coordenadas 'x', 'y' y 'z' del efector final con respecto al punto cero del robot, para obtener los valores angulares de cada servomotor y alcanzar el punto deseado en el espacio de trabajo.

Además de esto se realiza una interpolación entre la posición actual del robot y la posición objetivo, para hacer el movimiento más suave y fluido.

3.6 Diseño de algoritmo de envío, recepción y procesamiento de datos.

Para el diseño del algoritmo primero se modelan y parametrizan los datos que se necesitan enviar desde la aplicación al microcontrolador; los datos son los ángulos de posición de cada servomotor y las coordenadas en 'x', 'y' y 'z' del efector final, debido a que los servomotores se mueven de 0° a 180° se puede enviar cada posición dentro de un byte de información. En cambio, las coordenadas se manejan en milímetros y el robot es capaz de

llegar a más de 300 milímetros, para enviar las coordenadas se necesita enviar carácter a carácter la posición en el eje coordenado.

Debido a que la comunicación es serial se necesita manejar un sistema de banderas para identificar el dato que se está enviando, para dichas banderas se define una letra o carácter especial para cada función; la 's' indica que el próximo valor va a ser el número del servo al que se quiere cambiar el valor del ángulo, la 'a' es el indicativo para que el siguiente byte que se envíe se almacene como el ángulo del servomotor que se configuró anteriormente con la 's', para las coordenadas se envía una 'p' indicando que los siguientes nueve bytes se almacenen como las coordenadas en 'x', 'y' y 'z', cada grupo de 3 caracteres es el valor de las coordenadas respectivamente, adicionalmente se configura un condicional en caso de recibir un '-' indicando que dicha coordenada es negativa. En la Figura 18 se observa el diagrama de funcionamiento de la interfaz de MATLAB, y en la Figura 19 se observa el correspondiente diagrama para el PIC.

Figura 18 Lógica de programación MATLAB

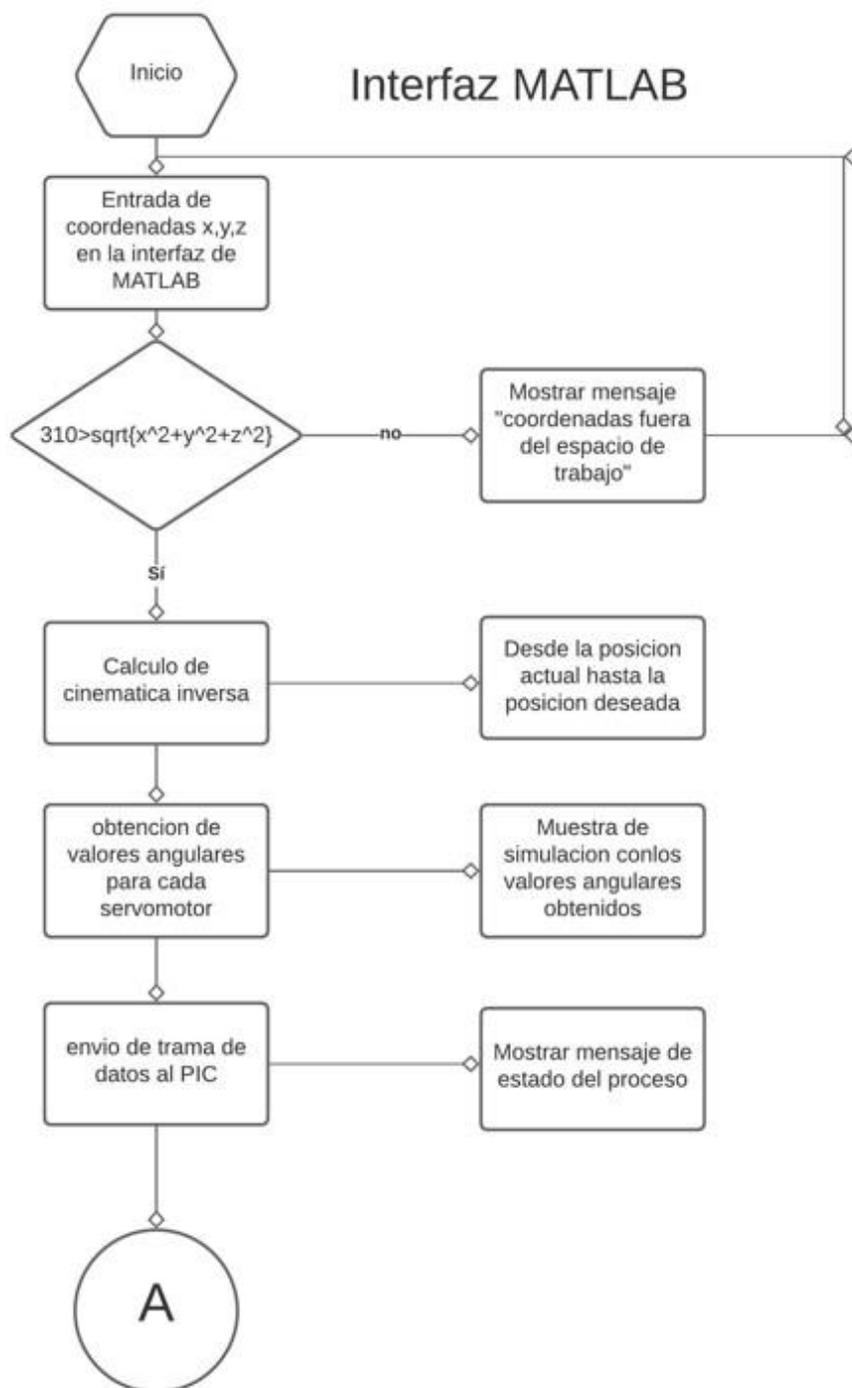
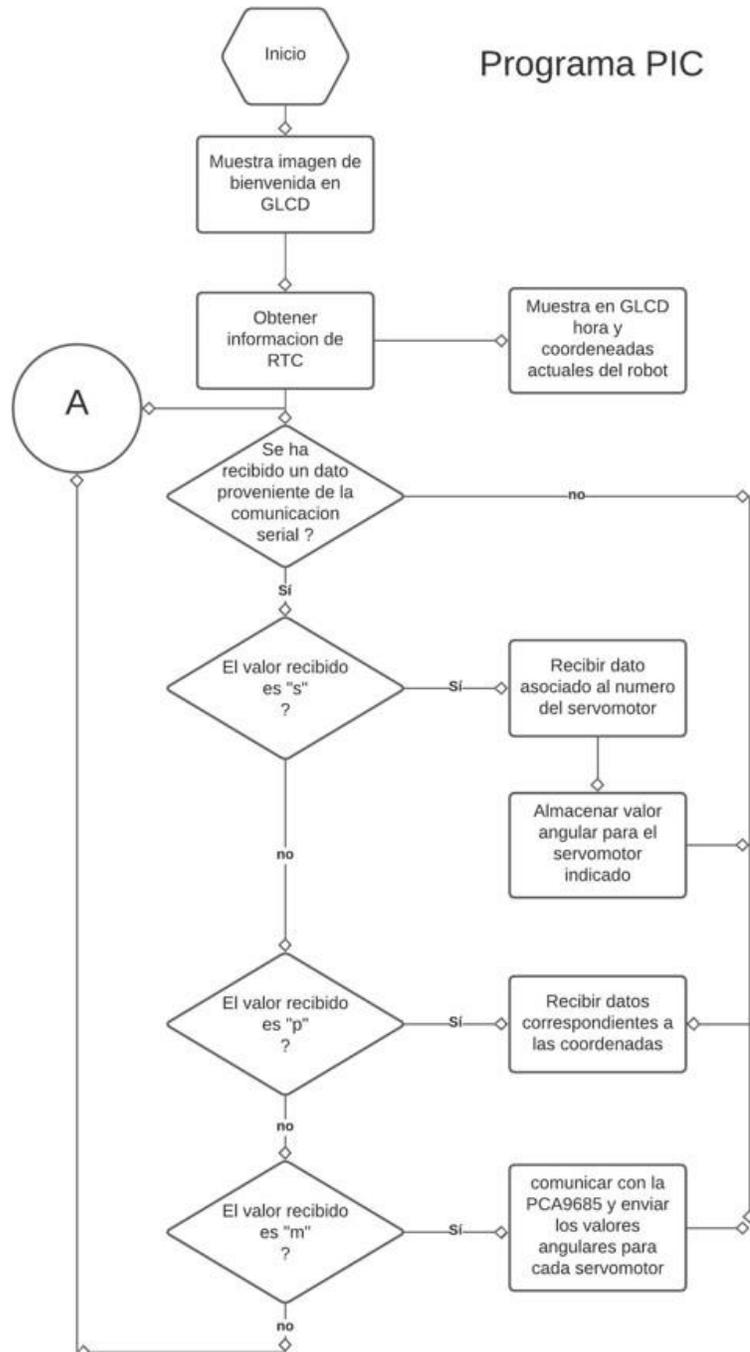


Figura 19 Lógica de programación PIC



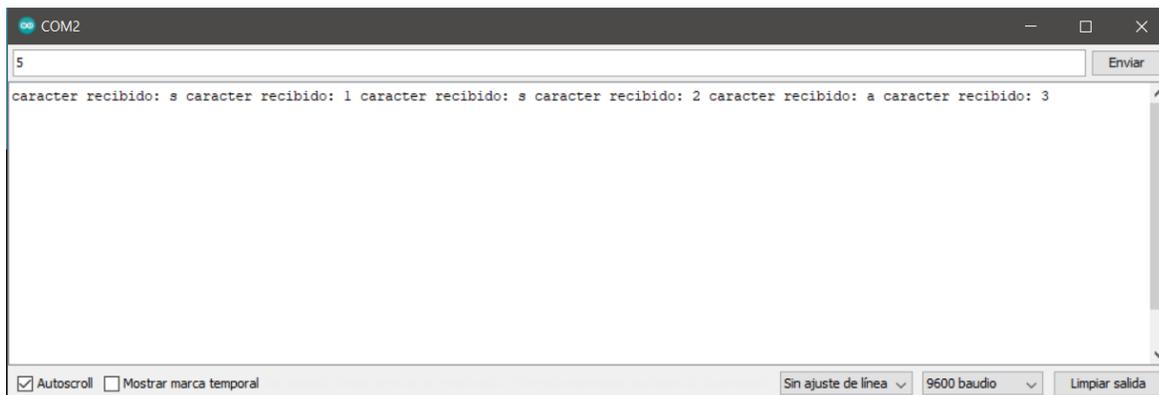
verifica que las medidas de los diseños sean correctas y se procede a ubicar los componentes de la forma en la que se desea que estén en la disposición final de la tarjeta. Primero se ubica el microcontrolador en el centro debido a que la mayoría de las conexiones se centralizan en él.

4. Resultados

4.1 Envío y recepción de datos en el microcontrolador

Después de realizada la programación del microcontrolador se compila el programa y se realizan pruebas verificando el correcto funcionamiento de las comunicaciones. Se envían datos desde un monitor serial en un computador simulando carácter a carácter, esperando una respuesta por cada dato enviado. En la Figura 21 se evidencia la respuesta según los datos enviados por el puerto serial, así como la respuesta por parte del microcontrolador por cada carácter enviado.

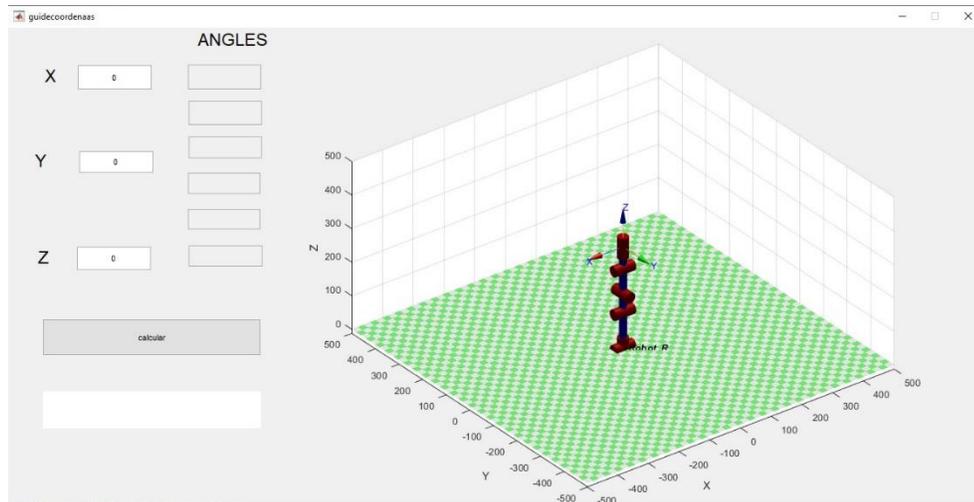
Figura 21 Envío y recepción de datos



4.2 Interfaz gráfica

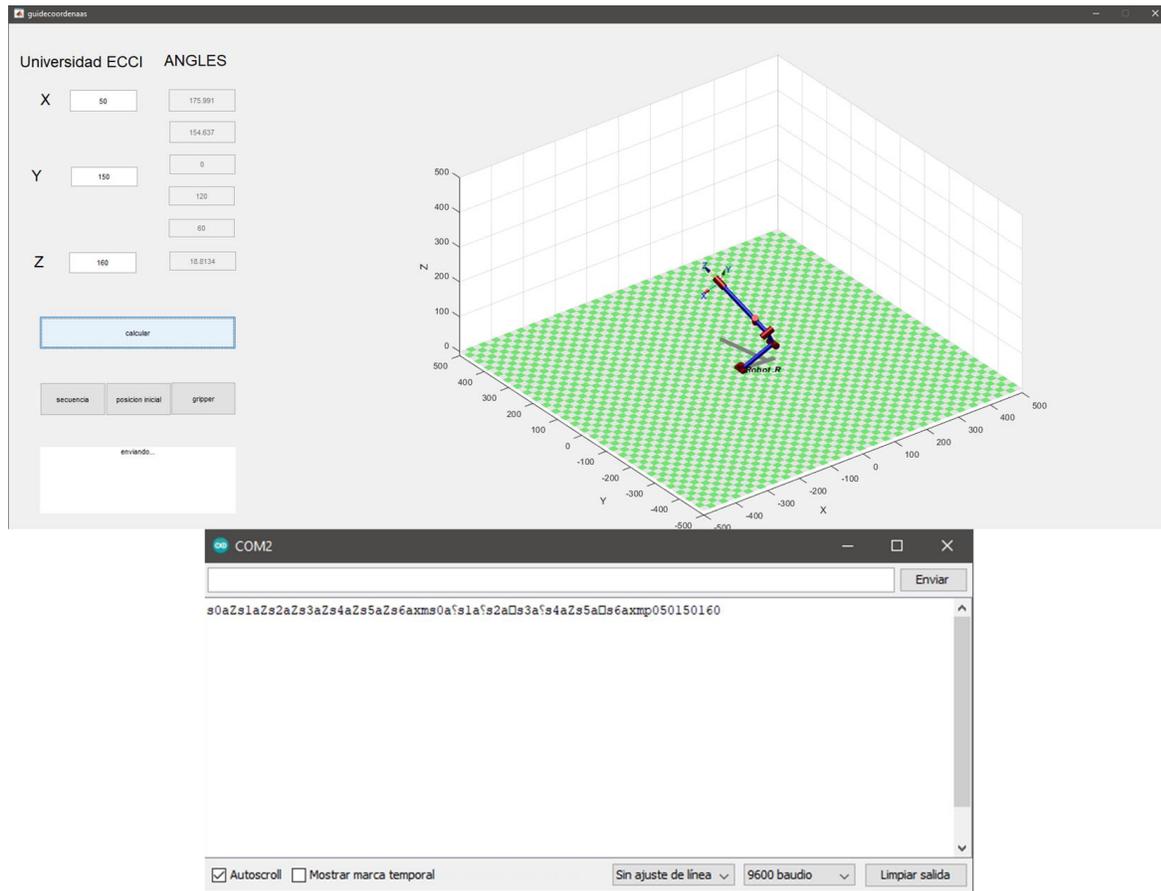
Se puede observar la interfaz gráfica resultante en la Figura 22, lo primero que se puede observar en la interfaz es la simulación de la posición del robot en el espacio de trabajo de acuerdo a la posición actual o a la posición que se está moviendo. En la parte izquierda se observan 3 cuadros para ingresar las coordenadas 'x', 'y' y 'z', abajo de esto se encuentra un botón para indicar el inicio del proceso. Finalmente se encuentra un cuadro para mostrar mensajes del estado actual del proceso.

Figura 22 Interfaz MATLAB



Finalmente se prueba el envío de datos de la interfaz colocando la cantidad de puntos de interpolación en 2 y se envían 3 coordenadas, en la Figura 23 se puede evidenciar las coordenadas enviadas y el resultado en la terminal serial de Arduino. Este resultado muestra que la trama para el envío de datos se aplica correctamente.

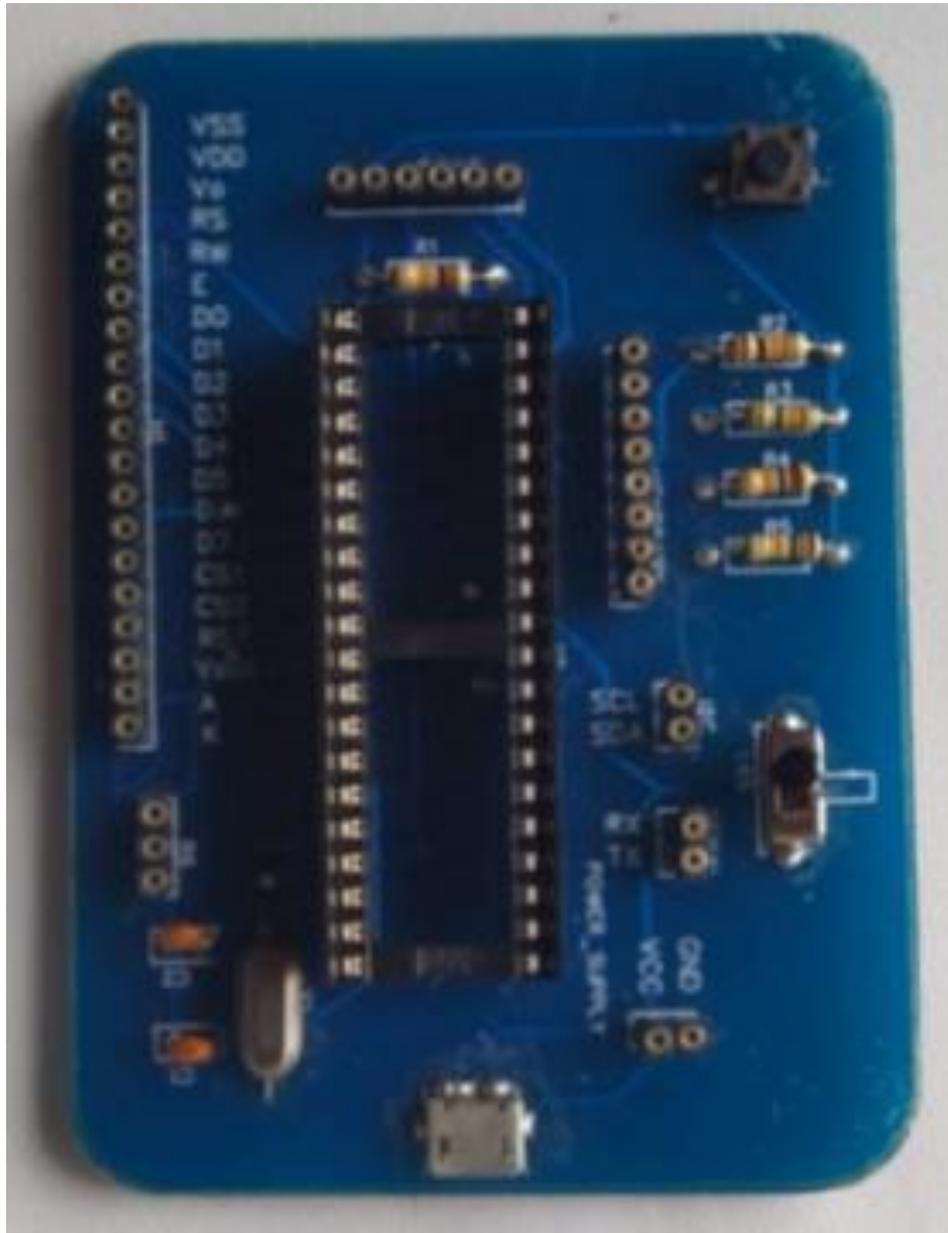
Figura 23 Comunicacion interfaz MATLAB con puerto serial



4.3 PCB y montaje del circuito

El circuito impreso enviado a fabricación se puede evidenciar en la Figura 24, se comparan las dos caras del circuito con la simulación realizada y se prueban las conexiones midiendo continuidad para verificar las conexiones. Se puede evidenciar el circuito montado en la PCB en la figura.

Figura 24 PCB



4.4 Impresión de datos en la pantalla GLCD

4.4.1 Impresión de imágenes

Con la pantalla lo primero que se puede observar es la capacidad de imprimir imágenes que se hallan digitalizado anteriormente. Al encender el circuito lo primero que se puede

observar es la impresión del logo de la universidad ECCI en la pantalla como indicativo de que las configuraciones iniciales se realizaron adecuadamente. En la figura \ref se puede observar la impresión del logo adecuadamente, se ve de baja calidad debido a la resolución de la pantalla, pero se imprime adecuadamente.

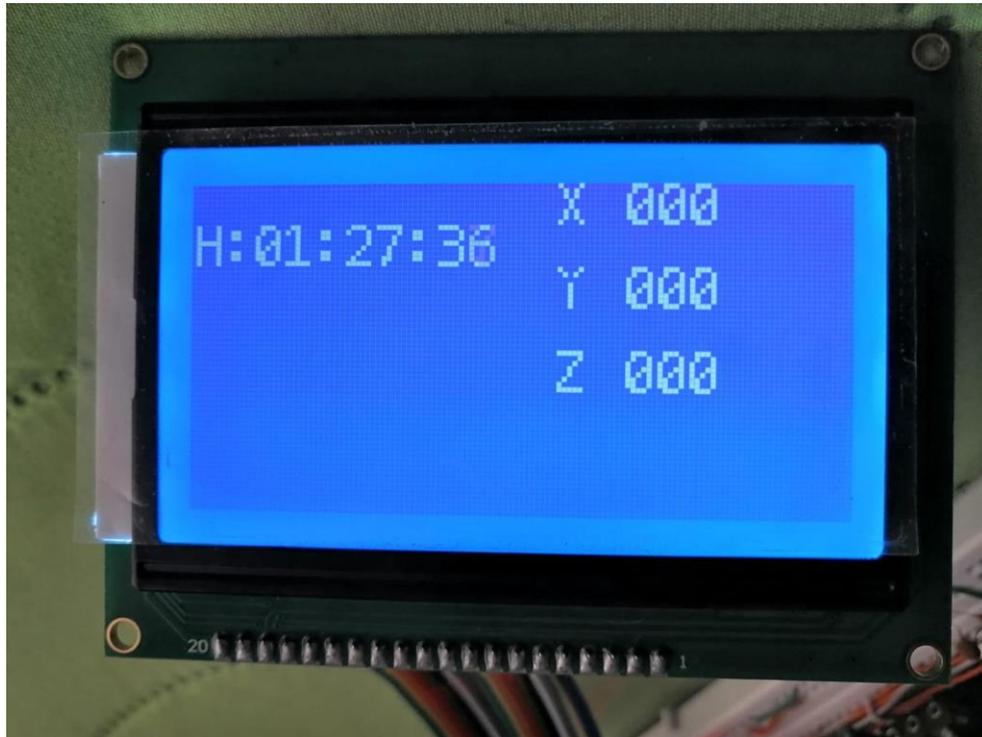
Figura 25 Impresión de imágenes en la pantalla GLCD



4.4.2 Impresión de datos numéricos

Tan pronto se inicia el robot se empiezan a imprimir datos en la pantalla, al principio se imprime la hora adquirida del reloj en tiempo real y las letras 'X', 'Y' y 'Z' con un campo vacío al frente de cada letra debido a que todavía no hay datos acerca de estas coordenadas. Tan pronto se reciben datos por medio de la comunicación UART se imprimen las coordenadas enfrente de las letras correspondientes a cada eje \ref.

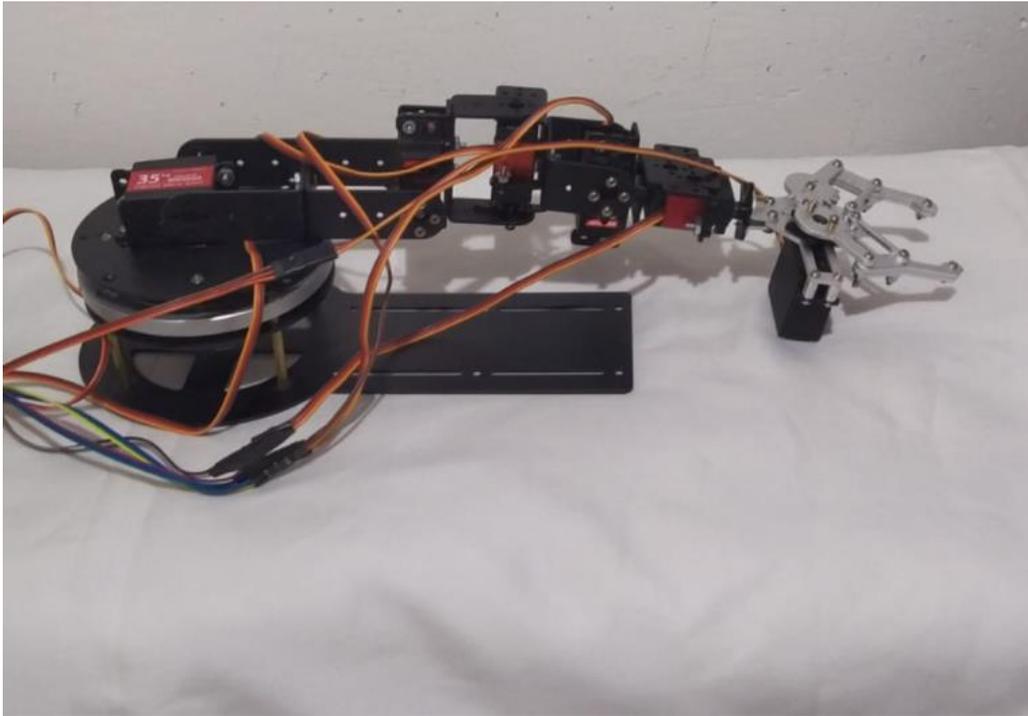
Figura 26 Impresión de datos numéricos



4.5 Estructura mecánica del Robot

Con el robot ensamblado en la Figura 27 se evidencia que cada eslabón sostiene adecuadamente la estructura sin evidenciar desgaste o fracturas en la posición horizontal que es donde se presenta mayor estrés sobre las piezas.

Figura 27 Estructura mecánica



4.6 Movimiento del robot

Al energizar y encender el circuito de control del robot lo primero que se realiza es mover los servomotores a la posición 0 como se puede ver en la Figura 28, evidenciando que el robot es capaz de llegar a dicha posición con facilidad sin importar la posición en la que este la estructura.

Figura 28 Posición 0



4.6.1 Secuencia Pick and place

Al presionar la tecla del teclado matricial que activa la esta secuencia el robot se mueve inicialmente a la posición 0 e inicia la secuencia para mover piezas de un punto A (Véase Figura 29) a un punto B (Véase Figura 31). El robot realiza adecuadamente la secuencia en 4 partes, primero va al punto A donde toma la pieza que se va a mover, luego eleva el objeto hasta el punto más alto que alcanza el robot, luego se lleva a el punto B donde se suelta la pieza que se tomó, finalmente se regresa a la posición 0.

Figura 29 Posición A

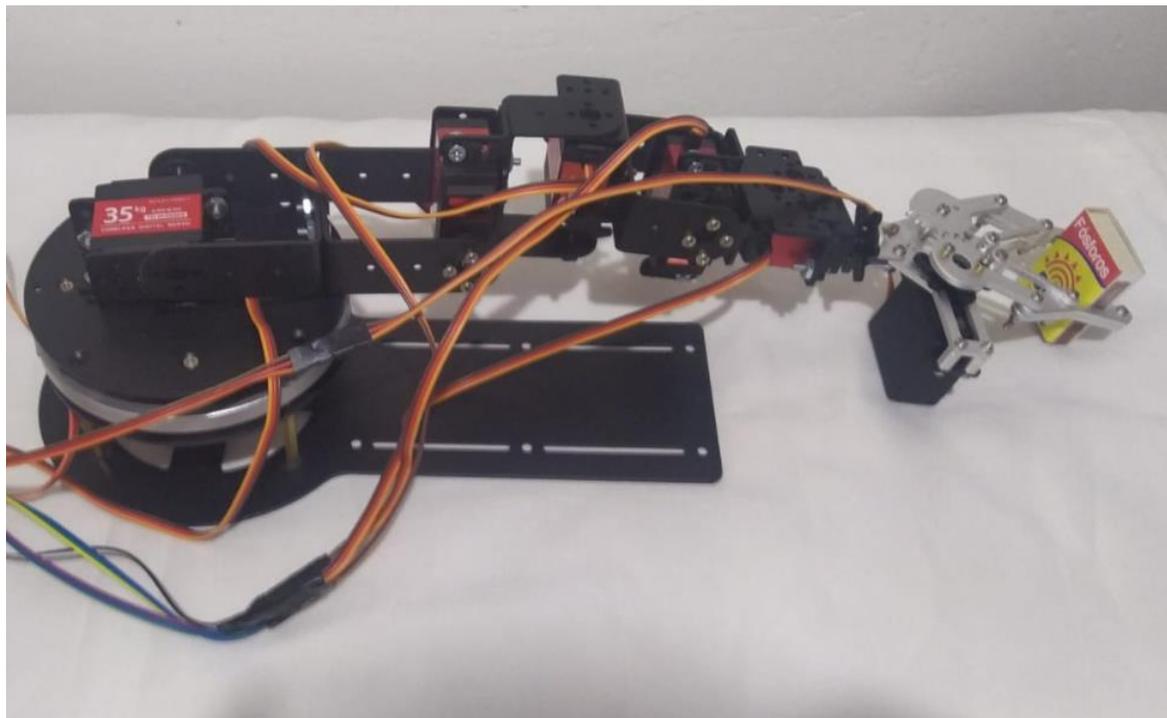


Figura 30 Posición elevada



Figura 31 Posición B

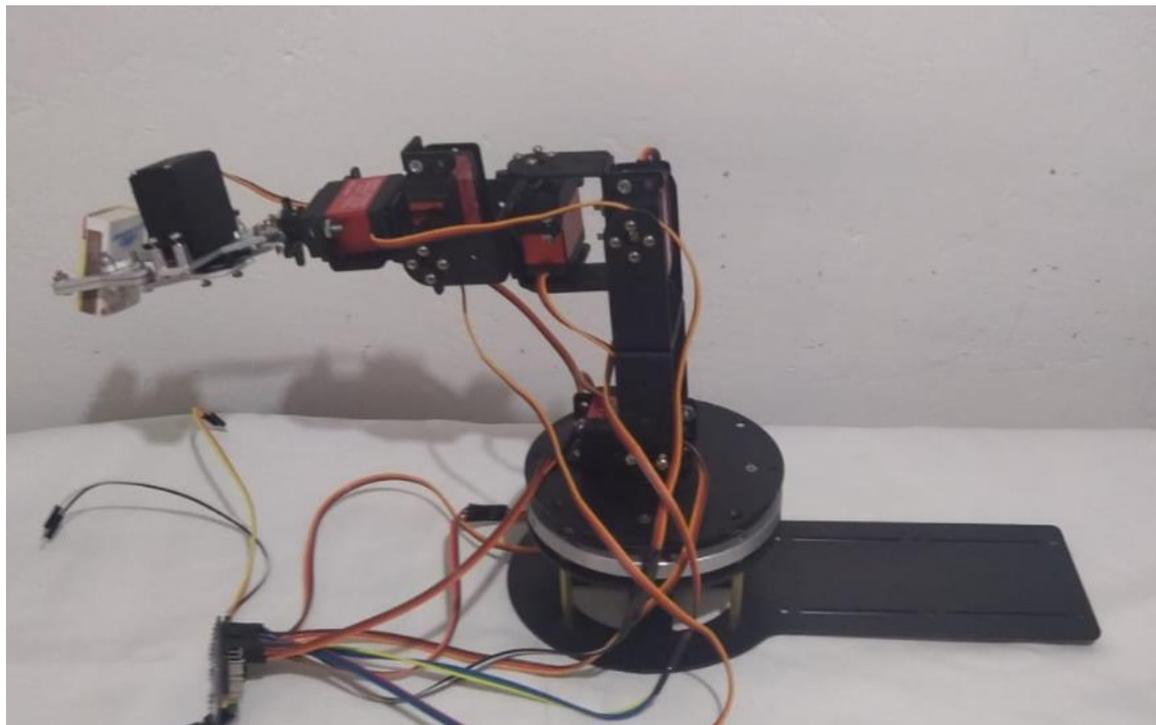
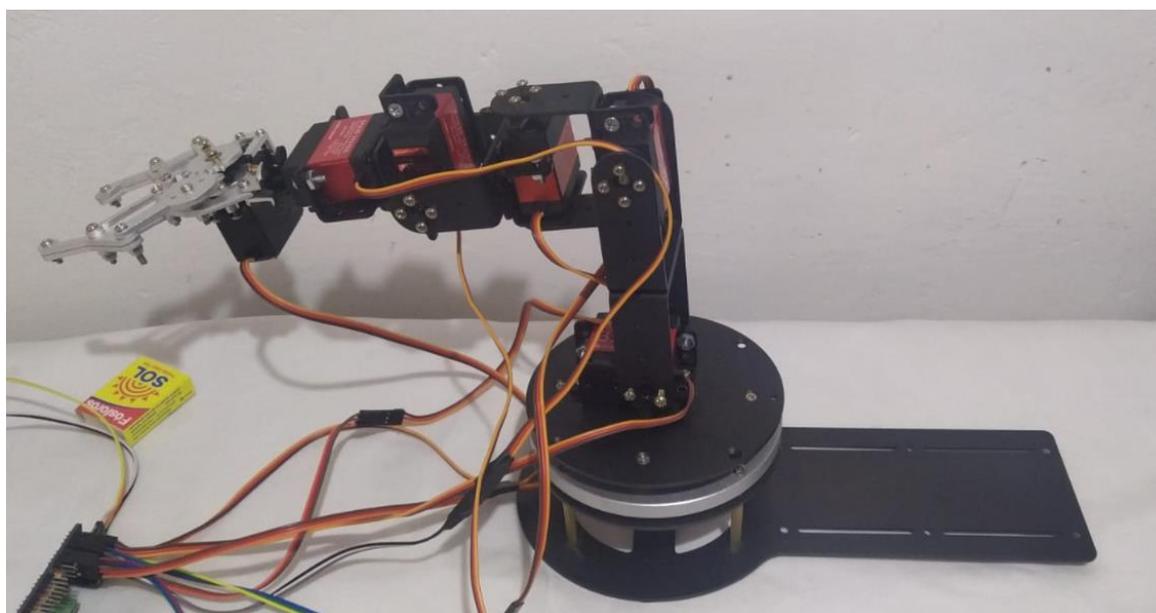


Figura 32 Posición B con Gripper abierto



4.6.2 Movimiento por medio de comandos enviados desde el computador

El funcionamiento del envío de los comandos y la interpretación de los comandos por parte del microcontrolador se pueden evidenciar al usar las 4 funcionalidades para mover el brazo robótico que ofrece la interfaz de usuario. La primera es la de ingresar 3 coordenadas y hacer que el robot llegue a dicha posición con su efector final, la segunda es el de realizar una secuencia Pick and Place preestablecida, la tercera es llevar el robot a su posición inicial y finalmente un botón que abre y cierra el manipulador. Los resultados se pueden observar el video montado en la plataforma de Youtube que se puede observar en el siguiente enlace <https://youtu.be/Ltl0onBhn8k>.

5. Conclusiones y recomendaciones

5.1 Conclusiones

Después de realizar las pruebas necesarias y con los resultados obtenidos en el desarrollo del proyecto se puede concluir lo siguiente; El uso de distintos componentes y periféricos electrónicos permite desarrollar proyectos o dispositivos complejos de forma sencilla e intuitiva. Inicialmente se puede observar la efectividad del microcontrolador PIC18F4550 para el manejo de varios periféricos y las comunicaciones utilizadas para el desarrollo del robot, su papel como componente central se desarrolló con éxito, teniendo en cuenta las limitaciones de este.

A pesar del correcto funcionamiento del microcontrolador para manejar una gran variedad de componentes por medio de sus periféricos presenta ciertas limitaciones en la velocidad de procesamiento de los datos recibidos y enviados, se utilizan pausas en el código de envío de MATLAB para evitar bloqueos en el funcionamiento del microprocesador.

El controlador PCA9685 permite controlar y manejar hasta 16 servomotores dando una gran flexibilidad para controlar robots o dispositivos que tengan 16 o menos motores, a su vez separa la fase de potencia del circuito de control de los motores, facilitando el montaje físico y su programación, debido a que solo se necesita realizar las conexiones adecuadamente y programar por medio de funciones los comandos y datos necesarios para su funcionamiento. Si se realiza el manejo de PWM con los módulos PWM del microcontrolador sería necesario adaptar una fase de potencia y realizar un circuito aparte para esto.

Para el manejo del robot es necesario realizar los cálculos cinemáticos para hallar las coordenadas del efector final y la posición de los actuadores para llegar al punto en el espacio, este procedimiento se facilita por medio del toolbox de Peter Corke, este complemento contiene funciones que piden parámetros específicos de entrada para realizar los procedimientos matemáticos necesarios y devuelve los datos necesarios listos para enviarlos por medio del puerto USB al microcontrolador.

La creación de la interfaz gráfica también se facilita por el uso la herramienta GUIDE de Matlab, aquí se diseña ubicando los componentes para interactuar con el programa y facilitar el manejo y control del robot por parte del usuario final. Se agregan objetos para controlar como campos de texto o botones y objetos para visualizar datos como gráficos, letreros, tablas, etc. Con esto Matlab genera un código backend para poder realizar los cálculos y procedimientos necesarios según la lógica de procesamiento del proyecto.

El modelado 3d y el diseño mecánico se facilitó debido al uso de brackets de aluminio de diseño genérico y servomotores que se puedan acoplar fácilmente. Realizando correctamente el análisis estático y seleccionando adecuadamente el torque de los servomotores este tipo de estructuras funcionan adecuadamente en un ambiente no industrial y que este dentro de las tolerancias de los brackets de aluminio.

5.2 Recomendaciones

Después de terminar con el desarrollo de este proyecto realizamos unas sugerencias para realizar desarrollos de este tipo; inicialmente se recomienda para usos no industriales o didácticos el diseño de mecanismos y brazos robóticos como el que presentamos anteriormente, debido a su practicidad en el ensamblaje y el control de los servomotores permite que se pueda crear con facilidad un dispositivo robótico funcional y con las capacidades necesarias dependiendo del estándar de servomotores.

A la hora de realizar la programación aconsejamos realizar una copia de seguridad de versiones de los compiladores y todos los entornos de desarrollo que se utilicen, esto debido a que una actualización puede afectar la forma en la que se compila una versión del código. Así mismo es necesario crear un repositorio GIT o un sistema de guardado de versiones en caso de realizar cambios que puedan afectar la funcionalidad conseguida hasta un punto y sea necesario devolverse a una versión anterior.

Para el circuito electrónico se recomienda realizar un montaje inicial en protoboard y mantener este prototipo por si ocurre algún error de montajes posteriores o de fabricación en las PCB.

Se aconseja conocer cada uno de los componentes electrónicos que se vayan a utilizar, verificando su forma de funcionamiento, los protocolos de comunicaciones que maneja, el rango de voltajes en los que trabaja, el cableado o montaje necesarios, entre otros, esto con el fin de evitar cortocircuitos y problemas en el desarrollo de la lógica de programación.

6. Bibliografía

- [1] D. Ibrahim, *Advanced PIC Microcontroller Projects in C: from USB to RTOS with the PIC18F*, vol. 53, no. 9. 2008.
- [2] R. Toulson and T. Wilmschurst, “Starting with Serial Communication,” *Fast Eff. Embed. Syst. Des.*, pp. 135–169, Jan. 2017, doi: 10.1016/B978-0-08-100880-5.00007-4.
- [3] E. Carletti, “Comunicación-Bus I2C.”
- [4] M. Clary, “Interfacing to an LCD Screen Using an Arduino,” pp. 1–9, 2015.
- [5] S. H. HosseinNia, I. Tejado, and B. M. Vinagre, “Fractional-order reset control: Application to a servomotor,” *Mechatronics*, vol. 23, no. 7, pp. 781–788, Oct. 2013, doi: 10.1016/J.MECHATRONICS.2013.03.005.
- [6] C. D. Toth, J. O’Rourke, and J. E. Goodman, “Robotics,” pp. 1343–1376, Nov. 2017, doi: 10.1201/9781315119601-51.
- [7] C. L. Tascon, “Una revisión sobre los fundamentos generales: Primero y segundo problemas de la cinemática,” *MOMENTO*, vol. 0, no. 6, pp. 61–70, Jan. 1992, Accessed: Oct. 14, 2021. [Online]. Available: <https://revistas.unal.edu.co/index.php/momento/article/view/35207>.
- [8] D. De Postgrado and C. Contreras, “UNIVERSIDAD NACIONAL EXPERIMENTAL DEL TÁCHIRA VICERRECTORADO ACADÉMICO INTRODUCCIÓN A LOS MICROCONTROLADORES Parte II Diciembre 2007.”
- [9] K. U. Ariawan, G. S. Santyadiputra, and I. W. Sutaya, “Design of Hexapod Robot Movement Based on Arduino Mega 2560,” in *Journal of Physics: Conference Series*, 2019, vol. 1165, no. 1, doi: 10.1088/1742-6596/1165/1/012011.
- [10] M. Neha, R. Laddha, and A. P. Thakare, “Implementation of serial communication using UART with configurable baud rate,” *Int. J. Recent Innov. Trends Comput. Commun.*, pp. 4–263, Accessed: Oct. 14, 2021. [Online]. Available: <http://www.ijritcc.org>.
- [11] J. Luis Díaz Rodríguez and A. Pardo García Ing Edwin Yesidh Ríos Contreras, “OPTIMUM PWM MODULATION IMPLEMENTATION FOR A THREE PHASE INDUCTION MOTOR CONTROL USING THE-TMS320C6416 DSK-CARD IMPLEMENTACION DE UNA MODULACION PWM OPTIMIZADA PARA EL CONTROL DE UN MOTOR TRIFASICO DE INDUCCION USANDO LA TARJETA DSP-TMS320C6416 DSK.”
- [12] G. T. Herman, “Geometry of digital spaces,” p. 216, 1998.

