



INSTITUTO TECNOLÓGICO DE CD. GUZMÁN

**TITULACIÓN INTEGRAL
TESIS**

TEMA:

**CARACTERIZACIÓN DE UN SISTEMA DE SEIS SERVOMOTORES PARA LA
ESTABILIZACIÓN EN UN PLANO DENTRO DE LOS ÁNGULOS DE
ROTACIÓN Y TRASLACIÓN**

**QUE PARA OBTENER EL TÍTULO DE:
ING. ELECTRÓNICA**

PRESENTA:

SAÚL LARES MUNGUÍA

ASESOR(A):

M.I.E. CARLOS ENRIQUE MACIEL GARCÍA

CD. GUZMÁN JALISCO, MÉXICO, SEPTIEMBRE DE 2020



"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cd. Guzmán, Jal. a **15/octubre /2020**

ASUNTO: Liberación de proyecto
para titulación integral

C. M.I.E. FAVIO REY LUA MADRIGAL
JEFE DEL DEPTO. DE DIV. DE EST. PROF.
PRESENTE

Por este medio informa que ha sido liberado el siguiente proyecto para titulación integral:

Nombre del estudiante y/o egresado:	SAUL LARES MUNGUIA
Carrera:	INGENIERIA ELECTRONICA
No. de Control:	C16290006
Nombre del Proyecto	CARACTERIZACION DE UN SISTEMA DE SEIS SERVOMOTORES PARA ESTABILIZACION EN UN PLANO DENTRO DE LOS ANGULOS DE ROTACION Y TRASLACION
Producto:	TESIS

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestros egresados.

ATENTAMENTE

M.E.H. MARCO ANTONIO SOSA LÓPEZ
JEFE DEL DEPTO. ELÉCTRICA Y ELECTRÓNICA

José Ma Hernández O.

Luis Enrique Salvador Cano

 M.I.E. CARLOS ENRIQUE MACIEL GARCIA	 M.I.E. JOSE MARIA HENANDEZ OCHOA	 M.I.E. LUIS ENRIQUE SALVADOR CANO
Nombre y firma del asesor	Nombre y firma del revisor	Nombre y firma del revisor



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Ciudad Guzmán

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

C.p. Archivo
MASL/adc

Av. Tecnológico No. 100 C.P. 49100



@150

Cd. Guzmán, Jal. Tel. Computador (341) 5752050
www.tecnm.mx | www.itog.edu.mx



Agradecimientos

Me gustaría agradecer primera mente a mis padres que siempre han estado ahí para apoyarme, siempre en las buenas y las malas, apoyándome económicamente y moralmente, siempre queriendo lo mejor para mí, de igual manera a mi novia que era la que tenía que aguantar mi mal genio cuando más estresado estaba por que algún proyecto o trabajo no me salía ya que a veces recalaba con ella, gracias por aguantar todos esos malos momentos, donde no tenía nada que ver pero terminaba perjudicada.

A su vez un gran agradecimiento a los maestros del tecnológico por siempre apoyarnos a todo en nuestros proyectos y cosas externas que se presentaban, como, concursos, eventos varios y excursiones, fueron grandes maestros y serán grandes amigos que uno se lleva del tecnológico.

Resumen

Este proyecto se originó con la idea de aportar algo a las generaciones futuras, y a su vez al laboratorio de electrónica, esto para tener un sistema en el cual se puedan probar diversos controladores a su vez ejecutar la practica en el sistema realizado, este sistema Ball and Plate (bola y plato), en este sistema usaremos 6 ejes los cuales a diferencia del de 3 ejes se obtenga una estabilidad mejor, a su vez usar una plataforma desarrollada en Python para así tener una plantilla genérica de cualquier tipo de Ball and Plate, a esto me refiero a que funciona para cualquier medida del sistema, ya sea un sistema más pequeño o más grande, estas características se ingresan mediante un archivo JSON, ya después dentro del script se realizan los cálculos pertinentes para dibujar tu sistema y poder probarlo, para hacer los cálculos de posicionamiento de los servomotores se realizó mediante cinemática inversa, ya una vez realizado esto podríamos modificar los ángulos (Yaw, Pitch y Roll) y en este caso solo moverlo en el eje z ya que el eje x y y no son necesario moverlos en esta plataforma.

Abstrac

This project originated with the idea of contributing something to future generations, and in turn to the electronics laboratory, this in order to have a system in which various controllers can be tested and in turn execute the practice in the system performed, this Ball system. and Plate (ball and plate), in this system we will use 6 axes which, unlike the 3 axis, will obtain better stability, in turn using a platform developed in Python to have a generic template of any type of Ball and Plate To this I mean that it works for any measurement of the system, be it a smaller or larger system, these characteristics are entered through a JSON file, and then within the script the relevant calculations are performed to draw your system and be able to test it , to make the positioning calculations of the servo motors was done using inverse kinematics, and once this was done we could modify the angles (Yaw, Pitch and Roll) and in this case only move it on the z axis since the x and y axis do not need to be moved on this platform.

Índice

AGRADECIMIENTOS	IV
RESUMEN	IV
ABSTRAC.....	V
ÍNDICE	VI
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS	IX
INTRODUCCIÓN	1
CAPÍTULO I GENERALIDADES DEL PROYECTO.....	4
ANTECEDENTES DEL PROYECTO	5
PLANTEAMIENTO DEL PROBLEMA	6
JUSTIFICACIÓN	6
OBJETIVOS	7
<i>Objetivo General</i>	7
<i>Objetivos Específicos</i>	7
HIPÓTESIS	8
ALCANCES	8
CAPÍTULO II MARCO TEÓRICO	9
ARDUINO	10
<i>Pines digitales</i>	10
<i>Pines analógicos</i>	11
<i>Pines alimentación sensores</i>	11
<i>Microcontrolador de comunicaciones</i>	11
<i>Microcontrolador de programación</i>	12
<i>Botón reset</i>	12
<i>Puerto USB</i>	12
<i>Conector de alimentación</i>	12
SERVOMOTOR.....	13
<i>Especificaciones servo corona DS558HV:</i>	13
PYTHON.....	14
<i>Características</i>	15
CONTROLADOR PCA9685	15
<i>Características</i>	16
TRANSFORMACIONES LINEALES	17
CAPÍTULO III CARACTERIZACIÓN DEL ÁREA DE TRABAJO.....	19
GENERALIDADES DE LA EMPRESA.....	20
ANTECEDENTES DE LA EMPRESA	20
MISIÓN	21
VISIÓN.....	21
OBJETIVOS	21
<i>Académico</i>	21
<i>Vinculación</i>	22
<i>Planeación</i>	22
<i>Administración de Recursos</i>	22
<i>Calidad</i>	22

VALORES.....	22
POLÍTICA DE CALIDAD	23
DESCRIPCIÓN DEL ÁREA DE NEGOCIOS	23
PUESTO ASIGNADO Y FUNCIONES	24
CAPÍTULO IV DESCRIPCIÓN DE LAS ACTIVIDADES DESARROLLADAS	25
ACTIVIDADES DEL PROYECTO	26
<i>Instalación de VSC (Visual Studio Code)</i>	26
<i>Instalación del IDE Arduino</i>	26
<i>Descargas IDLE (Entorno de Desarrollo Integrado para Python) Python 3.7 32 bits</i>	28
<i>Descargar librerías en VSC</i>	30
<i>Reimpresión de las piezas para plataforma</i>	31
<i>Ensamble de las piezas</i>	37
<i>Scripting el código en arduino para el movimiento de los servomotores</i>	42
<i>Conexión del PCA9685</i>	44
<i>Ajuste de los servomotores</i>	45
Cálculo de los 0° - 180° naturales de cada servomotor (rango de trabajo).....	46
Cálculo de los 0° o 180° y 90° horizontales para cada servomotor	48
<i>Scripting el código en Python para leer un archivo JSON</i>	49
<i>Scripting el código en Python para la identificación del sistema y posición de servomotores</i>	52
ACTIVIDADES ADICIONALES	56
<i>Comprensión de la sintaxis para POO en Python</i>	56
CAPÍTULO V RESULTADOS OBTENIDOS.....	58
RESULTADOS.....	59
RESPUESTA A LA HIPÓTESIS	62
CAPÍTULO VI ALCANCES Y LIMITACIONES.....	64
ALCANCES	65
LIMITACIONES	65
CAPÍTULO VII CONCLUSIONES Y RECOMENDACIONES	66
CONCLUSIONES	67
RECOMENDACIONES.....	67
REFERENCIAS BIBLIOGRÁFICAS	68

Índice de figuras

Figura 2. 1 Arduino UNO	10
Figura 2. 2 Servomotor DS558HV.....	13
Figura 2. 3 PCA9685	16
Figura 2. 4 Transformación de rotación	17
Figura 4. 1 Logotipo Visual Studio Code.....	26
Figura 4. 2 Arduino UNO	27
Figura 4. 3 Download Arduino.....	27
Figura 4. 4 Barra de direcciones.....	28

Figura 4. 5 Ventana de comandos.....	28
Figura 4. 6 Inicialización Python en CMD	29
Figura 4. 7 Instalando Numpy	29
Figura 4. 8 Actualización de pip.....	30
Figura 4. 9 Icono Extensions.....	30
Figura 4. 10 Búsqueda de Python.....	31
Figura 4. 11 Extensión Python.....	31
Figura 4. 12 Ball and Plate primera versión ITCG.....	32
Figura 4. 13 Piezas Ball and Plate	33
Figura 4. 14 Pandeo de pieza	34
Figura 4. 15 Impresora 3D Prusa	34
Figura 4. 16 MatterControl	35
Figura 4. 17 Generación de G-code.....	36
Figura 4. 18 Comienzo de impresión.....	37
Figura 4. 19 Base servomotores.....	37
Figura 4. 20 Base ensamblada.....	38
Figura 4. 21 Enumeración de servomotores	38
Figura 4. 22 Conexión PCA	39
Figura 4. 23 Cámara TB-916776	39
Figura 4. 24 Brazos fibra de carbono.....	40
Figura 4. 25 Soporte cámara	40
Figura 4. 26 Montaje soporte para la base del plato	41
Figura 4. 27 Montaje base del plato.....	41
Figura 4. 28 Plataforma Ball and Plate	42
Figura 4. 29 Gestor de librerías	43
Figura 4. 30 Inicialización de servos código arduino.....	43
Figura 4. 31 Fijación del ángulo código arduino.....	44
Figura 4. 32 PCA9685	44
Figura 4. 33 Encontrar Duty código arduino	46
Figura 4. 34 Primera adquisición de rango de trabajo	46
Figura 4. 35 Segunda adquisición de rango de trabajo	47
Figura 4. 36 Ángulo 0° y 180° horizontales	48
Figura 4. 37 Ángulo 90° horizontal.....	48
Figura 4. 38 Características de la plataforma archivo JSON.....	50
Figura 4. 39 Función leer archivo JSON	51
Figura 4. 40 Función para inicializar variables de la fuente JSON	52
Figura 4. 41 Ángulos de rotación.....	53
Figura 4. 42 Clase punto.....	53
Figura 4. 43 Constructor de la clase plataforma	54
Figura 4. 44 Calculo puntos de la base	54
Figura 4. 45 Método para fijar posición	55
Figura 4. 46 Cargar puntos de servos y plato.....	55
Figura 5. 1 Plataforma Ball and Plate	59
Figura 5. 2 Resultado cinemática inversa.....	60

Figura 5. 3 Plot puntos de la base de la plataforma (Naranja).....	61
Figura 5. 4 Plot de la base del plato (Azul)	61
Figura 5. 5 Plot plataforma completa.....	62

Índice de tablas

Tabla 4. 1 Conexión PAC – Servomotores	45
Tabla 4. 2 Valores del rango de trabajo	47
Tabla 4. 3 Adquisición ángulos horizontales	49

Introducción

este referente sistema consta solamente de una bola y un plato, es un sistema muy simple pero a su vez tiene un alto nivel de dificultad, aunque por muy sencillo que parezca el controlar una bola sobre el plato ya sea en el centro, que de igual manera puedes hacer rutinas de movimiento, ya sea calculando una trayectoria ya preestablecida con los servomotores o mover el set point y variar este para crear el movimiento, claro con ayuda de un controlador para que este sea eficiente.

Para realizar todo esto es necesario dos cosas muy importantes, como censar la bola para obtener su posición en el plato y el que posición los motores deben tomar para realizar un movimiento en el plato, el cual se mueve en los ángulos de Euler (Yaw, Pitch y Roll).

Para el cálculo de la posición de los motores se realizan diversos cálculos matemáticos, y para estos utilizamos cinemática inversa, ya que con esta técnica nos permite determinar el movimiento de una cadena de artículos para lograr que un actuador final se ubique en una posición correcta.

Ya una vez se determina esto de primera mano, se necesita el saber donde se encuentra la bola, esto se realiza mediante el uso de una cámara, la cual haciendo el uso de la librería OpenCV y algunas modificaciones puedes tu ingresar el color de la bola que se usara y determinar con coordenadas (x, y) el posicionamiento de la bola. Ya en esto se le pueden aplicar diversos controles a la plataforma, pues ya solo es cuestión de saber dónde queremos posicionar la bola y calcular los ángulos de Euler y mediante una función poder mover el plato.

Esto es aplicable para la prueba de los diversos controladores que se deseen y a su vez ejercer diversas practicas parala reafirmación de los conocimientos adquiridos en cualquier materia de control, y a su vez sintonización de controladores.

En el capítulo I se especificarán las generalidades del proyecto, se explica por qué se realiza este proyecto y como es que surge el problema principal para el cual se realizó este proyecto, así como a su vez los objetivos a cumplir dentro de este lapso que se tiene para el proyecto, todo para llegar a un alcance a llegar en el mismo.

En el capítulo II se encuentra toda la teoría que se necesitó para el proyecto, tomando en cuenta que diversos términos y temas no se pudieron abarcar ya que son conocimientos previos que se requieren tener antes de la realización de este proyecto, por lo cual solo se implementan conceptos nuevos y relevantes a conocer.

En el capítulo III están las características de la empresa en la cual se realiza el proyecto, en este caso es un proyecto del banco de proyectos del ITCG (Instituto Tecnológico de Ciudad Guzmán), por lo cual están sus especificaciones y una explicación del área y puesto asignado donde se estuvo trabajando.

En el capítulo IV se podrá apreciar todas las actividades realizadas en el proyecto, a su vez explicando cada una de ellas de lo que se realizó en cada actividad, tanto como las actividades que se realizan para el proyecto y las que son adicionales que se necesitaron para alguna actividad del proyecto, pero no de manera directa dentro de él.

En el capítulo V están los resultados obtenidos realizando todas las actividades y discutiendo el cómo fue que funciona la plataforma, de la misma manera comprobando si se logró cumplir con los objetivos plasmados al inicio.

En el capítulo VI están los alcances que se esperan obtener los cuales se refieren a los beneficios que la empresa espera tener o llegó a tener con la realización del proyecto, a su vez plasmando las limitaciones u obstáculos que se presentaron en la realización del mismo.

En el capítulo VII está la conclusión general del proyecto después de haberlo realizado, a su vez algunas recomendaciones para que las futuras generaciones que intenten usar este documento para la realización de alguna plataforma nueva puedan contemplar algunas cosas que podrían ayudar para realizarlo con un menor tiempo o evitando el tener tiempos muertos durante el proyecto.

CAPÍTULO I
Generalidades del proyecto

Antecedentes del proyecto

Este Proyecto se comenzó a desarrollar dentro del departamento de ingeniería Eléctrica y Electrónica del instituto tecnológico de Cd. Guzmán, en la carrera de Ing. Electrónica, se comenzó trabajando con el jefe del laboratorio de instrumentación.

Esta idea surgió del jefe de laboratorio, puesto que los instrumentos y equipo del laboratorio son demasiado viejos, ya que estos están desde que el laboratorio fue inaugurado, estos equipos tienen más de 25 años, por lo tanto, no se adaptan a las necesidades de la industria hoy en día, ya que nuestra rama tiende a estar actualizando la tecnología año con año y no se puede estar uno tan retrasado.

Por lo cual en una charla entre el jefe del laboratorio y los alumnos integrantes del club de robótica Adictronics, sugirieron realizar una plataforma en la cual se logre implementar un controlador, ya sea un PID Clásico hasta un Difuso, esta plataforma serviría para diversas materias dentro de las especialidades de Mecatrónica e Instrumentación y Control.

Tomando como base una plataforma ya planteada por otros compañeros de la carrera, de una plataforma Ball and Plate a 6 ejes, así solo se necesitará reimprimir la plataforma y a su vez hacer modificaciones necesarias para el material que se tiene para realizarla, ya una vez esto se trabajará en hacer un script genérico en Python, tanto para si se desea utilizar el procesador de una computadora (con sistema operativo Windows o Linux) y realizando el control de los servomotores mediante un arduino conectado por serial al ordenador o usar una Raspberry y con esa misma realizar el control de los mismos.

Con esto teniendo como resultado que solo usando un archivo de formato JSON se logre ingresar datos de la nueva plataforma a utilizar esto sin necesidad de hardcodear el script o realizarle alguna modificación.

Planteamiento del problema

En los laboratorios asignados a la carrera de Ing. Electrónica por parte del ITCG (Instituto Tecnológico de Ciudad Guzmán) se cuenta con equipo demasiado viejo en el cual es difícil poder generar practicas lo bastante buenas para alcanzar las competencias establecidas a la carrera, ya que en la actualidad la tecnología ha avanzado mucho y los equipos que se usan en la actualidad son diferentes con los que se cuenta, es por esto que se desea realizar una plataforma en la cual se pueda implementar cualquier tipo de controlador, para de esta forma comparar el funcionamiento de cada uno, independientemente los factores.

Justificación

El sistema Ball and Plate por más difícil que pueda llegar a verse al usar varias salidas y solo una entrada no llega a ser tan complejo desde un punto de vista matemático, ya que existen diversas soluciones para que, a lo que se quiera llegar a hacer en el movimiento del mismo sea posible, esto estando dentro del rango de trabajo de los servomotores, ya que se le encontró un buen fin al uso de las plataformas Ball and Plate (en este caso con 6 ángulos de libertad).

Tomando de ejemplo diversas versiones anteriores realizadas por compañeros del tecnológico, se logró identificar diversas fallas y errores en sus versiones, ya que realizaban scripts mal desarrollados, con mucho hardcodeo dentro del mismo y esto solamente hablando en la parte de codificar, ya en la parte del diseño mecánico de la plataforma estaba de buena forma estructurada.

Aprovechando lo que se tenía y con el ámbito de mejorarlo se planteó diseñar un algoritmo de posicionamiento, que sea más eficiente (hablando de los tiempos de respuesta, realización de cálculos y ajuste de motores) para poder obtener el mejor funcionamiento del controlador aplicado a la plataforma, mediante el uso de este script el cual se hará usando el lenguaje de programación Python, dentro de este se

estructuran los cálculos matemáticos respectivos para realizar el posicionamiento e identificación de la plataforma.

A su misma vez se usará POO (programación orientada a objetos) esto para poder aligerar el código y a su vez su eficacia, así de esta manera no saturaremos la memoria de nuestro procesador con información ya no relevante después de su uso, esto suele pasar ya que si se hace de otra manera se estarán realizando los cálculos iniciales una y otra vez siendo que algunos solo necesitan calcularse una única vez, así con esto se aligera el tiempo de respuesta de la plataforma.

Ya hablando de hacer genérico la plataforma es muy factible, ya que se busca de la misma manera hacer este código factible para diversas plataformas que se deseen realizar en un futuro semejantes a esta, hablando de que sea de 6 ejes, y usando un ordenador con sistema operativo Linux o Windows, esto vuelve más comercial el script a realizar, ya que no es para una sola plataforma en específico.

Para concluir esta plataforma es muy factible para el uso en diversas materias de la carrera de Ing. Electrónica, ya que puedes realizar diversas prácticas sobre la plataforma, aplicando controladores y comparándolos, hacer sintonización de los parámetros de ganancias, entre otras prácticas. Esto nos lleva a un resultado de aprendizaje de los alumnos mejor del que se puede llegar ahora usando el equipo que se tiene.

Objetivos

Objetivo General

- Diseño de un algoritmo de posicionamiento para un Ball and Plate mediante el uso de cinemática inversa.

Objetivos Específicos

- Construcción de la plataforma Ball and Plate usando impresión 3D.

- Comprensión de la cinemática inversa.
- Desarrollo del script en Python para la identificación de la plataforma.
- Desarrollo de los cálculos matemáticos para el posicionamiento de los servomotores.
- Calibración de los servomotores para calcular sus puntos de origen (0° horizontales).
- Desarrollo de la plataforma haciendo uso del plot de Matlab.

Hipótesis

Un script de python alimentado por un archivo tipo JSON, con las características de la plataforma, genera un modelado del sistema Ball and Plate, usando cinemática inversa, el cual se genera una función por la cual se ingresen las coordenadas y los ángulos de rotación para que el plato adquiera dicha posición, transmitiendo los datos desde un ordenador por un puerto serie hacia un procesador ATmega el cual manda las señales para el movimiento de los 6 servomotores.

Alcances

El alcance al que se pretende llegar es obtener un script en Python el cual pueda identificar cualquier tipo de plataforma Ball and Plate a 6 ejes, y así generar la identificación del sistema sin necesidad de modificar el código, todo esto ingresando solamente un archivo JSON con las características de la plataforma a trabajar.

CAPÍTULO II
Marco Teórico

Arduino

Arduino es una plataforma de código abierto de prototipos electrónicos que se basa en hardware y software flexibles y fáciles de usar que ponen al alcance de cualquier persona la construcción de circuitos electrónicos/robots. En lo referente a hardware, se basa en placas que se pueden ensamblar a mano o que se pueden comprar directamente preensambladas. Cada una de las placas lleva un microcontrolador en el que se carga el programa software que es necesario desarrollar para “darle vida” a la placa. (Moreno Muño & Córcoles Córcoles, 2017)

En la figura 2.1 se muestra la placa arduino Uno, señalando en esta las partes importantes.

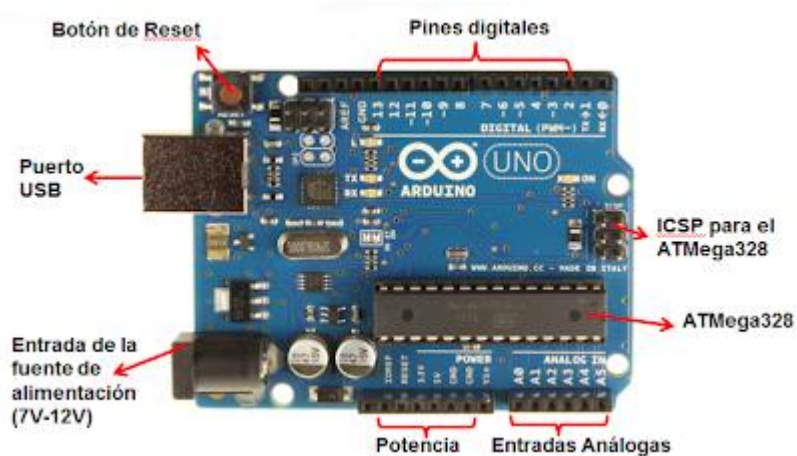


Figura 2. 1 Arduino UNO

Pines digitales.

Los pines digitales son las conexiones digitales de los dispositivos conectados en la placa. La placa de Arduino cuenta con 14 pines digitales, que van del 0 al 13.

Una señal digital solo puede tener dos estados:

- 0 (LOW, bajo, false): Indica 0V de tensión enviados desde la placa.
- 1 (HIGH, alto, true): Indica 5V de tensión enviados desde la placa.

(Moreno Muño & Córcoles Córcoles, 2017)

Por lo tanto, cuando ponemos un pin digital a valor HIGH, la placa suministra 5V de tensión por la salida que hayamos indicado, y si ponemos el valor a LOW suministrará 0V de tensión. (Ojo: Hay que tener en cuenta que los 5V no siempre son 5 ni los 0 siempre son 0). Los pines digitales de Arduino pueden ser usados tanto de entrada como de salida. (Moreno Muñoz & Córcoles Córcoles, 2017)

Pines analógicos

Los pines analógicos pueden medir un rango de valores de voltaje, a diferencia de los digitales que solo entienden dos valores: 0-1, o lo que es lo mismo, 0V o 5V.

Con los pines analógicos vamos a poder leer valores intermedios entre 0V y 5V, representados con un valor entero comprendido entre 0 y 1023, ya que la información se representa en números de 10 bits, y también vamos a poder escribir en los pines valores comprendidos entre 0 y 255, ya que la información se representa en números de 8 bits.

En el punto anterior hemos hablado sobre pines digitales, si te fijas en ellos verás que aparecen algunos con el símbolo “~” en la placa, este símbolo indica que pueden ser utilizados también como pines analógicos. (Moreno Muñoz & Córcoles Córcoles, 2017)

Pines alimentación sensores

Además de los pines de entrada y salida descritos anteriormente, Arduino dispone de pines que nos permiten alimentar componentes externos, concretamente uno con 5V y otro con 3,3V. También dispone de pines de tierra (GND). (Moreno Muñoz & Córcoles Córcoles, 2017)

Microcontrolador de comunicaciones.

El microcontrolador de comunicaciones se encarga de gestionar las comunicaciones con todo lo que se conecta a la placa. (Moreno Muñoz & Córcoles Córcoles, 2017)

Microcontrolador de programación

Este componente de la placa es el cerebro de la misma, es donde la placa almacena el programa que tiene que ejecutar y el que lo ejecuta. El microcontrolador de la placa se programa utilizando el IDE (Entorno de Desarrollo Integrado) de programación gratuito de Arduino. En los apartados siguientes explicamos cómo instalarlo y como ponerlo a funcionar. (Moreno Muño & Córcoles Córcoles, 2017)

Botón reset

El botón Reset permite reiniciar el programa que se ha cargado en el microcontrolador interrumpiendo la ejecución actual. Ten en cuenta que no borra el programa que se ha cargado, únicamente lo reinicia. (Moreno Muño & Córcoles Córcoles, 2017)

Puerto USB

El puerto USB es el puerto mediante el cual nos comunicaremos con la placa de Arduino. Sus funciones principales son:

- Alimentación
- Cargar los programas en el microcontrolador.
- Envío de información desde la placa al ordenador.

Conector de alimentación

Arduino dispone de un puerto de alimentación externo que nos permitirá hacer funcionar la placa sin utilizar un ordenador. Tienes que tener en cuenta el no alimentar la placa con más voltaje del que soporta, ya que podrías dañarla. Lo recomendado es alimentarla entre 7V y 12V. (Moreno Muño & Córcoles Córcoles, 2017)

Servomotor

El servo es un potente dispositivo que dispone en su interior de un pequeño motor con un reductor de velocidad y multiplicador de fuerza, también dispone de un circuito que controla el sistema. El ángulo de giro del eje es de 180° en la mayoría de ellos, pero puede ser fácilmente modificado para tener un giro libre de 360° , como un motor standard. (González, 2013)

Para controlar un servo se debe aplicar un pulso de duración y frecuencia específicas. Todos los servos disponen de tres cables, dos para alimentación Vcc y Gnd (4.8 a 6 [V]) y un tercero para aplicar el tren de pulsos de control, que hace que el circuito de control diferencial interno ponga el servo en la posición indicada, dependiendo del ancho del pulso. (González, 2013)

En la figura 2.2 se muestran la forma física del servomotor DS558HV, a continuación, se enuncian las especificaciones de cada uno de ellos.



Figura 2. 2 Servomotor DS558HV

Especificaciones servo corona DS558HV:

- Voltaje de funcionamiento: 6.0V / 7.4V
- Corriente de funcionamiento: 300mA / 400mA
- Velocidad de funcionamiento: 0.20sec. 60° / 0.18sec. 60°
- Puesto par: 12kg.cm / 14kg.cm
- Tamaño: 20X40X38.5mm
- Peso: 58g

- Banda muerta: $\leq 3\mu\text{seg}$
- Viaje de funcionamiento: 40° / un pulso lado 400US viajar
- Potenciómetro: 5 deslizador / Direct Drive
- Cojinete de bolas: MR106
- Engranaje: Metal
- Alambre de conector: 300mm
- Temperatura de funcionamiento: $-20\text{C} + 60\text{C}$

Python

Python fue creado por Guido van Rossum, un programador holandés a finales de los 80 y principio de los 90 cuando se encontraba trabajando en el sistema operativo Amoeba. Primariamente se concibe para manejar excepciones y tener interfaces con Amoeba como sucesor del lenguaje ABC. (Challenger Pérez, Díaz Ricardo , & Becerra García, 2014)

El 16 de octubre del 2000 se lanza Python 2.0 que contenía nuevas características como completa recolección de basura y completo soporte a Unicode. Pero el mayor avance lo constituye que este comenzó a ser verdaderamente desarrollado por la comunidad, bajo la dirección de Guido. (Challenger Pérez, Díaz Ricardo , & Becerra García, 2014)

El Python 3.0 es una versión mayor e incompatible con las anteriores en muchos aspectos, que llega después de un largo período de pruebas el 3 de diciembre del 2008. Muchas de las características introducidas en la versión 3 han sido compatibilizadas en la versión 2.6 para hacer de forma más sencilla la transición entre estas. (Challenger Pérez, Díaz Ricardo , & Becerra García, 2014)

Características

- Python es un lenguaje muy expresivo, es decir, los programas Python son muy compactos: un programa Python suele ser bastante más corto que su equivalente en lenguajes como C. (Python llega a ser considerado por muchos un lenguaje de programación de muy alto nivel.)
- Python es muy legible. La sintaxis de Python es muy elegante y permite la escritura de programas cuya lectura resulta más fácil que si utilizáramos otros lenguajes de programación.
- Python ofrece un entorno interactivo que facilita la realización de pruebas y ayuda a despejar dudas acerca de ciertas características del lenguaje.
- El entorno de ejecución de Python detecta muchos de los errores de programación que escapan al control de los compiladores y proporciona información muy rica para
 - detectarlos y corregirlos.
 - Python puede usarse como lenguaje imperativo procedimental o como lenguaje orientado a objetos.
 - Posee un rico juego de estructuras de datos que se pueden manipular de modo sencillo.

(Marzal Varó & Gracia Luengo, 2013)

Controlador PCA9685

El Controlador de servos de 16 canales está basado en el chip PCA9685 figura 2.3, que es un generador PWM con reloj interno y resolución de 12 bits, el cual puede ser programado a través de una interfaz I2C. La gran versatilidad de este módulo es que solo requiere de 2 pines para controlar todas las señales PWM y no requiere señal alguna de oscilador o temporización, por lo que funciona de forma totalmente autónoma. (Factory, 2015)

Este módulo es una alternativa más económica al shield controlador de motores de Adafruit, sin embargo, conserva la compatibilidad de software con este, por lo que las librerías de Adafruit funcionan perfectamente con este módulo. (Factory, 2015)

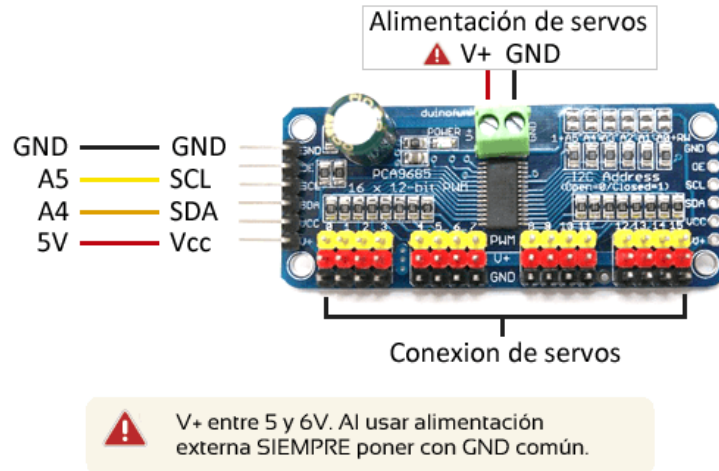


Figura 2. 3 PCA9685

Características

- Controlador de PWM PCA9685 12 bits de resolución
- El controlador de servos es completamente independiente y no requiere señales de reloj externas
- Control de 16 señales PWM a través de 2 hilos I2C
- Salidas configurables “push-pull” o “drenador abierto”
- Habilitación / Des habilitación de salidas para anular rápidamente la señal en todas las salidas
- Bloque de terminales para la alimentación de los servos
- Resistencias de 220 Ohm en serie con cada salida PWM para protección y manejo directo de leds
- Diseño que permite colocar varias tarjetas en el mismo bus para un máximo de 992 salidas PWM independientes
- 6 jumpers “soldables” para configurar la dirección de I2C

(Factory, 2015)

Transformaciones lineales

Sean V y W espacios vectoriales reales. Una transformación lineal T de V en W es una función que asigna a cada vector $v \in V$ un vector único $Tv \in W$ y que satisface, para cada u y v en V y cada escalar α . (I. Grossman, 2015)

$$T(u + v) = Tu + Tv \quad (1)$$

$$T(\alpha v) = \alpha Tv \quad (2)$$

Transformación de rotación.

Suponga que el vector $v = \begin{pmatrix} x \\ y \end{pmatrix}$ en el plano xy se rota un ángulo θ (medido en grados o radianes) en sentido contrario al de las manecillas de reloj. Llame a este nuevo vector rotado $v' = \begin{pmatrix} x' \\ y' \end{pmatrix}$.

Entonces, como se ve en la figura 2.4, si r denota la longitud de v (que no cambia por la rotación) (I. Grossman, 2015)

$$\begin{aligned} x &= r \cos \alpha & y &= r \operatorname{sen} \alpha \\ x' &= r \cos(\theta + \alpha) & y' &= r \operatorname{sen}(\theta + \alpha) \end{aligned}$$

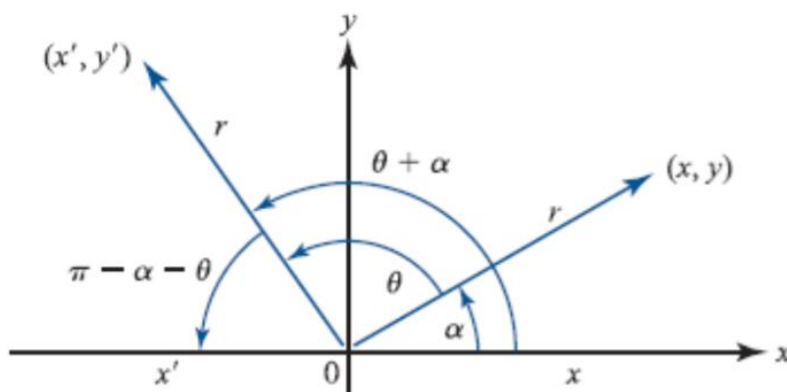


Figura 2. 4 Transformación de rotación

$$x' = x \cos \theta - y \sin \theta \quad (3)$$

Pero $r \cos(\theta + \alpha) = r \cos \theta \cos \alpha - r \sin \theta \sin \alpha$, de manera que

De manera similar $r \sin(\theta + \alpha) = r \sin \theta \cos \alpha + r \cos \theta \sin \alpha$, o sea

$$x' = x \sin \theta - y \cos \theta \quad (4)$$

Sea

$$A_0 = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (5)$$

(I. Grossman, 2015)

Entonces de (3) y (4), se ve que $A_0 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$. La transformación lineal $T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ definida por $T_v = A_0 v$, donde A_0 está dado por (5), se llama transformación de rotación. (I. Grossman, 2015)

CAPÍTULO III
Caracterización del área de
trabajo

Generalidades de la empresa

Avenida Tecnológico #100.

Ciudad Guzmán, Mpio. de Zapotlán el Grande, Jalisco, México

Tel. 01 (341) 575 20 50

Fax. 01 (341) 575 20 74

Web. www.itcg.edu.mx

Antecedentes de la empresa

En mayo de 1972 se hizo la petición al Sr. Presidente de la República, durante una de sus visitas a Cd. Guzmán, de la creación de una Institución Educativa de nivel superior en el Sur de Jalisco, que tuviera como finalidad propiciar el desarrollo cultural, técnico y económico de la región, así como reducir los flujos migratorios de los jóvenes estudiantes hacia las grandes ciudades en busca de su formación profesional. Así mismo, otro de los objetivos consistió en fomentar el arraigo de los egresados en sus lugares de origen.

El Instituto Tecnológico nace el 13 de septiembre de 1972, sobre una extensión de terreno de 26 hectáreas con el nombre de Instituto Tecnológico Regional No.29 de Ciudad Guzmán, Jalisco.

El inicio de sus actividades se hace el 20 del mismo mes y año en las instalaciones del CERETI (hoy CETIS) en la ciudad de Guadalajara, contando con una población de 120 alumnos de nivel licenciatura, 15 docentes y 10 administrativos.

El 22 de marzo de 1973, el CAPECE terminó la construcción de aulas y laboratorios, empezando en ese momento el nivel bachillerato ya en las instalaciones propias y en el mes de junio del mismo año se integraron los estudiantes que se encontraban en el CERETI de Guadalajara, a partir de ese periodo el ITCG ha logrado un desarrollo en todo sentido apegado a los lineamientos emanados de la SEP. Las carreras ofrecidas

por este Instituto actualmente se distribuyen de la siguiente manera: cuatro Licenciaturas y seis Ingenierías.

Misión

Somos una institución perteneciente al Tecnológico Nacional de México, que ofrece servicios de Educación Superior Tecnológica, formando profesionales íntegros, con capacidad científica y tecnológica para contribuir en la mejora de los sectores público, social y productivo, coadyuvando responsablemente en el desarrollo sustentable para favorecer la conformación de una sociedad más justa y humana.

Visión

Ser una Institución de Educación Superior Tecnológica con reconocimiento internacional y programas estratégicos de investigación e innovación tecnológica de vanguardia, acordes a las necesidades de los sectores público, social y productivo, ofreciendo servicios de calidad, producto del desarrollo, participación y superación permanente de nuestro talento humano, fomentando una relación responsable y comprometida con el medio ambiente.

Objetivos

Proporcionar el servicio educativo de calidad, orientado a la satisfacción de sus estudiantes, atención de necesidades y expectativas de sus partes interesadas pertinentes.

Académico

Gestionar los planes y programas de estudio para la formación profesional del estudiante.

Vinculación

Contribuir a la formación integral del estudiante a través de su vinculación con los sectores productivo y de gobierno, la sociedad, la cultura y el deporte.

Planeación

Realizar la planeación, programación, presupuestación, seguimiento y evaluación de las acciones para cumplir con los requisitos del servicio educativo.

Administración de Recursos

Administra los recursos humanos, financieros y materiales del Instituto Tecnológico, así como la prestación de los servicios generales, de cómputo y mantenimiento de infraestructura, para lograr la conformidad con los requisitos del servicio educativo.

Calidad

Medir, analizar y promover mejoras en los procesos del sistema de gestión de la calidad, para lograr la satisfacción de sus estudiantes y partes interesadas pertinentes.

Valores

- Interés público
- Respeto
- Respeto a los derechos humanos
- Igualdad y no discriminación
- Equidad de género
- Entorno cultural y ecológico
- Integridad
- Cooperación
- Liderazgo
- Transparencia
- Rendición de cuentas

Política de calidad

El Instituto Tecnológico de CD. GUZMÁN establece el compromiso de implementar sus procesos, orientados a la atención de los requerimientos del contexto y de los riesgos identificados, para la satisfacción de estudiantes y partes interesadas, cumpliendo con sus requisitos aplicables, mediante la eficacia y mejora de su SGC, sustentado en la Planeación Estratégica, Liderazgo y la Calidad del Proceso Educativo, conforme a la norma NMX-CC-9001-IMNC-2015/ISO 9001:2015.

Descripción del área de negocios

El proyecto fue realizado para el área de instrumentación y control, se estuvo trabajando en el laboratorio de instrumentación, en el cual el encargado jefe de laboratorio es el MIE Carlos Enrique Maciel, el cual nos facilitó mucho las herramientas y material a usar para la realización del proyecto, ya que se tuvo que rehacer una nueva plataforma, por la cual se necesitó el uso de las impresoras 3D que se encuentran dentro del mismo laboratorio, se nos facilitó el uso de nuevos servomotores y partes varias vitales dentro de la plataforma.

Dentro del departamento de ingeniería eléctrica y electrónica, se encuentra un grupo de maestros los cuales realizan investigación, dentro de este se encuentra el MIE Carlos, el cual, al ya concluir con la carga académica respectiva nos facilitó un proyecto en el cual podría ser usado para realizar una investigación y así facilitar las residencias profesionales y a su vez realizando la tesis poder titular al alumno.

El objetivo principal de esto es ayudar a los alumnos a tener más herramientas para aumentar su conocimiento teórico y práctico, en este caso enfocado al diseño y control, ya sea tanto el diseño electrónico o mecánico, y a su vez aplicar el diseño de controladores varios. Logrando dentro de esto el tener alumnos que cumplan con las competencias dictaminadas en el plan de estudio de la carrera de Ingeniería Electrónica.

Puesto asignado y funciones

El puesto que se tuvo asignado fue el de desarrollador de un algoritmo el cual identificara el sistema y creara una función la cual sea capaz de mover los actuadores, en este caso los servomotores, de tal manera que se compensara una ganancia dada para cierto punto, mediante cálculos matemáticos y demás, todo esto para dejar la plataforma lista para que cualquier persona, alumno o profesor, al querer aplicar un algoritmo de control pueda enfocarse solo a eso y olvidarse de la parte de identificación de sistema y diversas cosas, esto para que se logre enfocar más en las competencias necesarias para el área de control.

Los conocimientos requeridos son:

- Álgebra lineal
- Cinemática (mecánica)
- Python scripting
- Protocolos de comunicación (I2C, Serial)
- Diseño mecánico usando SolidWorks
- Uso de impresora 3D

Se comenzará realizando la reimpresión de las piezas para la plataforma (ya que se usará como base un modelo ya trabajado dentro del laboratorio) y a su vez realizando algunas modificaciones en ciertas piezas, una vez impreso y ensamblado todo, se realizará la identificación del sistema usando las medidas de la planta, enseguida se realizan los cálculos usando cinemática inversa para la posición en grados de cada uno de los servomotores para poder llegar a una posición deseada.

CAPÍTULO IV
Descripción de las actividades
desarrolladas

Actividades del proyecto

Instalación de VSC (Visual Studio Code)

Este es el editor el cual se utilizó para realizar los códigos Python para el proyecto. El cual tiene una licencia GNU, además de tener una interfaz de instalación muy amigable con los usuarios, de igual manera de dejare en los anexos un GIT repo en el cual se puede descargar las aplicaciones que son necesarias para el proyecto.

Dentro de la interfaz de instalación se debe aceptar los términos y condiciones, ubicar la carpeta donde se guardarán las carpetas por defecto del VSC y ya adelante se comenzará a instalar, ya una vez terminada la instalación se reinicia el ordenador y ya queda la aplicación lista para usarse.

En la figura 4.1 se puede mostrar el logotipo de aplicación Visual SC.



Figura 4. 1 Logotipo Visual Studio Code

Instalación del IDE Arduino

Para hacer la comunicación entre el ordenador (Computadora) y la plataforma (Ball and Plate), se requiere un dispositivo el cual pueda ser la comunicación y dar las órdenes a la plataforma que reciba por medio del ordenador que funcione como sistema de adquisición de datos (DAC), para esto, se usa un arduino, independientemente el arduino (UNO, Mega, Leonardo, Nano, entre otros) funciona de la misma manera ya que no se requieren demasiadas salidas, para este caso se usa

un arduino Uno como el que se muestra en la figura 4.2, ya que este es el que se tenía a la disponibilidad inmediata.



Figura 4. 2 Arduino UNO

Para hacer el código para el arduino es necesario tener el IDE Arduino para aquí poder hacer el código, compilarlo y subirlo al arduino.

El IDE de arduino al igual que VSC cuenta con una licencia GNU, para descargarlo se puede hacer desde el sitio oficial de arduino, se tienen diversas opciones para descargarlo, en ellas se encuentra la principal como se muestra en la figura 4.3.

A screenshot of the Arduino 1.8.12 download page. On the left, there is a teal circle containing the Arduino infinity logo. To its right, the text reads: "ARDUINO 1.8.12 The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the Getting Started page for installation instructions." On the right side, there are several download options: "Windows Installer, for Windows 7 and up", "Windows ZIP file for non admin install", "Windows app Requires Win 8.1 or 10" with a "Get" button, "Mac OS X 10.10 or newer", "Linux 32 bits", "Linux 64 bits", "Linux ARM 32 bits", and "Linux ARM 64 bits". At the bottom right, there are links for "Release Notes", "Source Code", and "Checksums (sha512)".

Figura 4. 3 Download Arduino

Aquí se puede observar que en la parte izquierda están diversos sistemas operativos para los cuales puedes descargar arduino, en este caso se descarga el que está en la parte superior izquierda que es para Windows versiones 7 hacia adelante.

Para la instalación, este consta con una interfaz de instalación muy amigable con el usuario, para ejecutar el instalador, se da doble click izquierdo sobre el archivo ejecutable .exe, y esto abrirá la interfaz de instalación, se selecciona la carpeta donde se guardarán los archivos predeterminados, ya se comenzará a instalar y una vez instalado estará listo para usarse.

Descargas IDLE (Entorno de Desarrollo Integrado para Python) Python 3.7 32 bits

Para poder hacer diversas pruebas durante la realización del script Python, se utilizó la ventana de comandos CMD en la cual se puede ejecutar códigos de Python, pero para esto es necesario primero instalar el IDLE de Python.

Este IDLE que usa se descarga del sitio web oficial de Python (referencia link) y su instalación intuitiva hace que solo con algunos clicks se pueda obtener el software adecuado de acuerdo a nuestro sistema operativo.

Una vez que se instale el IDLE, dentro de la carpeta la cual se localizan todos los archivos, aquí en la barra de direcciones colocamos CMD (comando) como se muestra en la figura 4.4 y se le da enter.

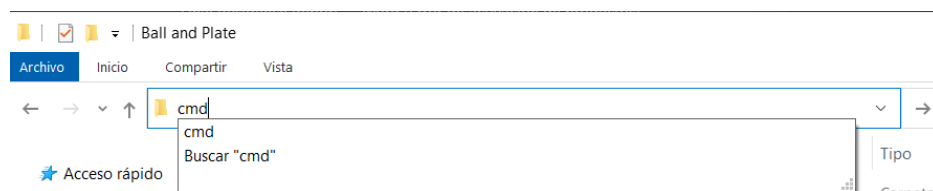


Figura 4. 4 Barra de direcciones

Este nos abrirá la ventana de comandos como se muestra en la figura 4.5.

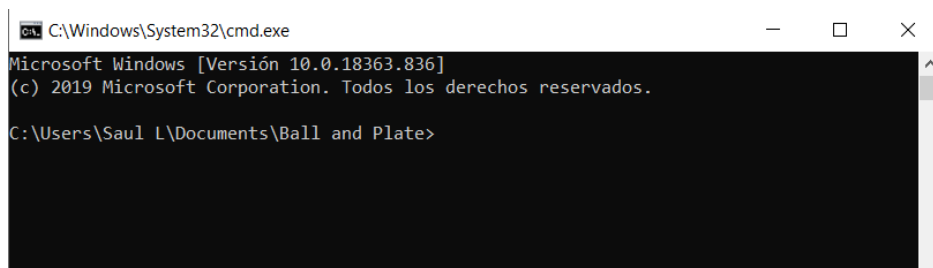
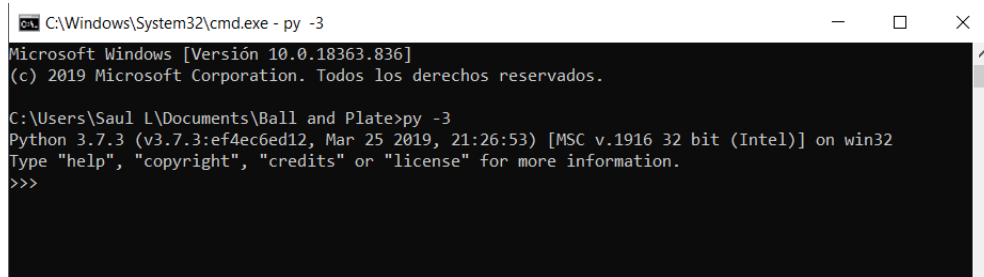


Figura 4. 5 Ventana de comandos

Para comenzar a utilizar Python en la ventana de comandos digita “`py -3`” como se muestra en la figura 4.6, y se logra apreciar cómo se carga Python 3.7.3 ya que esta es la versión que se instaló.



```
C:\Windows\System32\cmd.exe - py -3
Microsoft Windows [Versión 10.0.18363.836]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Saul L\Documents\Ball and Plate>py -3
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figura 4. 6 Inicialización Python en CMD

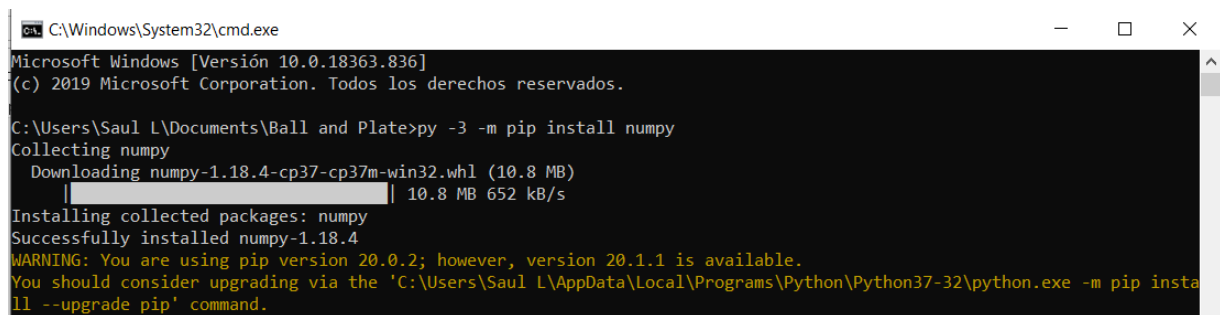
Ya dentro de aquí es necesario instalar diversas librerías para poder correr los scripts.

Las librerías necesarias son los siguientes:

- Numpy
- Pandas
- Matplotlib

Para instalarlas de una manera más sencilla es usando el pip, este es un sistema de gestión de paquetes, que es usado para instalar los package, ya que de esta manera es más amigable con el usuario y a su vez la más rápida.

Para hacerlo desde la ventana de comando es necesario colocar el siguiente comando “`py -3 -m pip install nombreDeLibreria`” al final solo se coloca el nombre de la librería que se quiere instalar. Se tiene que aclarar que no se debe estar dentro de Python para instalar las librerías, esto se realiza directamente en el CMD como se muestra en la figura 4.7 instalando la librería Numpy.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.18363.836]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Saul L\Documents\Ball and Plate>py -3 -m pip install numpy
Collecting numpy
  Downloading numpy-1.18.4-cp37-cp37m-win32.whl (10.8 MB)
    |#####| 10.8 MB 652 kB/s
Installing collected packages: numpy
Successfully installed numpy-1.18.4
WARNING: You are using pip version 20.0.2; however, version 20.1.1 is available.
You should consider upgrading via the 'C:\Users\Saul L\AppData\Local\Programs\Python\Python37-32\python.exe -m pip install --upgrade pip' command.
```

Figura 4. 7 Instalando Numpy

Y como se aprecia de igual manera en la figura 4.7 se tiene un mensaje en el cual se tiene ya disponible una versión más nueva del pip, para esto es solo digitar el comando que ahí nos sugiere para poder actualizar el pip, se puede mostrar cómo se realiza en la figura 4.8.

```
C:\Windows\System32\cmd.exe
C:\Users\Saul L\Documents\Ball and Plate>py -3 -m pip install --upgrade pip
Collecting pip
  Downloading pip-20.1.1-py2.py3-none-any.whl (1.5 MB)
    |-----| 1.5 MB 364 kB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.0.2
    Uninstalling pip-20.0.2:
      Successfully uninstalled pip-20.0.2
  Successfully installed pip-20.1.1
C:\Users\Saul L\Documents\Ball and Plate>
```

Figura 4. 8 Actualización de pip

Descargar librerías en VSC

Dentro de la aplicación VSC se necesita descargar la librería de Python, ya que los scripts son realizados en el mismo, por lo consecuente para instalar su debida extensión, para instalar alguna extensión se puede hacer desde la ventana principal del VSC en la parte izquierda se tiene un menú con diversos iconos dentro de estos se encuentra uno llamado “*Extensions*” o se puede ingresando usando el comando “*Ctrl + Shift + X*”, en la figura 4.9 se puede apreciar la sección donde se encuentra el icono “*Extensions*”.

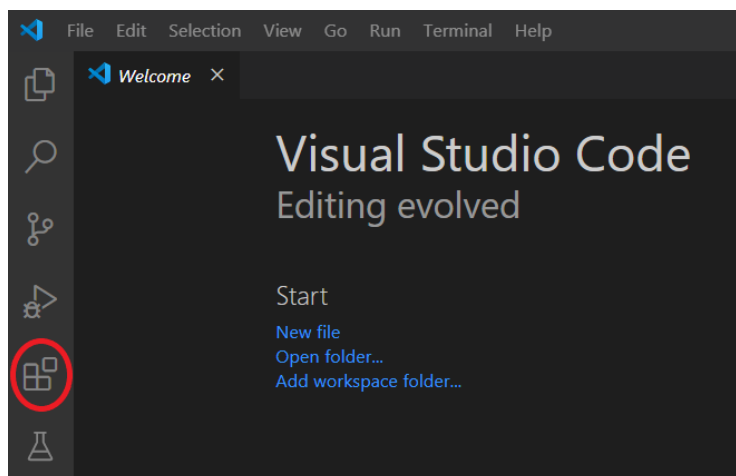


Figura 4. 9 Icono Extensions

Dentro de Extensions se busca Python como se muestra en la figura 4.10, como se puede apreciar, dentro de la búsqueda salen muchas extensiones, pero la que se busca es la marcada en rojo dentro de la misma imagen.

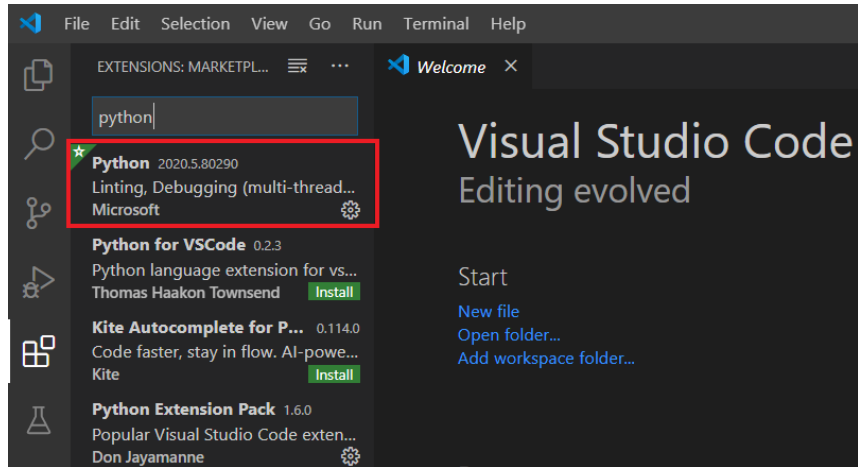


Figura 4. 10 Búsqueda de Python

Dando le click nos abrirá una ventana como se aprecia en la figura 4.11 en la cual se puede ver todas las características de la instalación e información diversa referente a lo mismo.

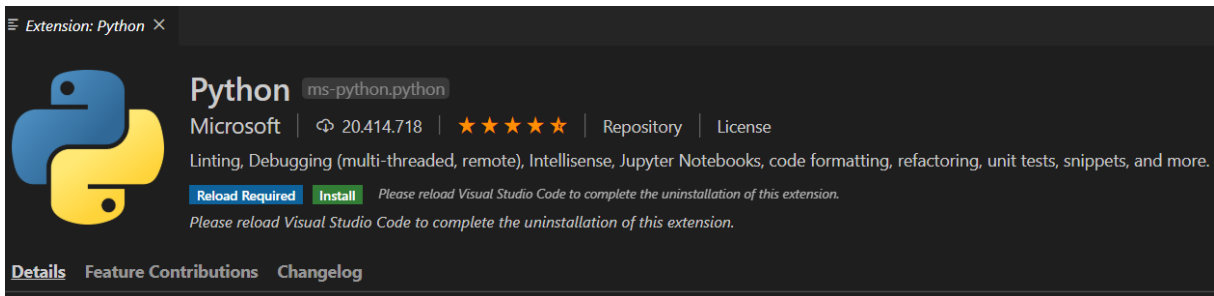


Figura 4. 11 Extensión Python

Para culminar la instalación se debe dar click en el botón verde que dice “Install” y solo comenzara la instalación y una vez instalado se puede comenzar a usar.

Reimpresión de las piezas para plataforma

Para la realización de esta nueva plataforma se basó la estructura en una plataforma ya antes diseñada, dentro del laboratorio del tecnológico en el cual sustentan el por qué es mejor una plataforma a 6 ejes que a 3 ejes, por lo cual se realiza una segunda plataforma basada en esta.

La plataforma es como se muestra en la figura 4.12 en la cual se logra apreciar la estructura completa, con todos los componentes necesario que son la cámara, el PCA9685, Step Down, Servomotor, Bola y plato. Dentro del mismo hay otras que no son indispensables como una pantalla Oled, control con joystick entre otros.



Figura 4. 12 Ball and Plate primera versión ITCG

Dentro del mismo se aprecia que se cuenta con una Raspberry Pi 4 la cual fue utilizada para el control de la plataforma.

Las piezas se realizaron utilizando el uso de SolidWorks, esta es una herramienta de diseño mecánico a 3D en la cual se pueden realizar diversas piezas basadas en

dibujos 2D, de los cuales una vez realizada cada pieza se realiza un ensamble para poder mostrar más a detalle como es el armado de algún sistema a realizar, en este caso la plataforma Ball and Plate.

De este esquema se toman las piezas como se muestra en la figura 4.13 a imprimir para la nueva plataforma, entre ellas las bases para la cámara, la base de los servomotores, la base del plato, entre otras diversas piezas que complementan el diseño.

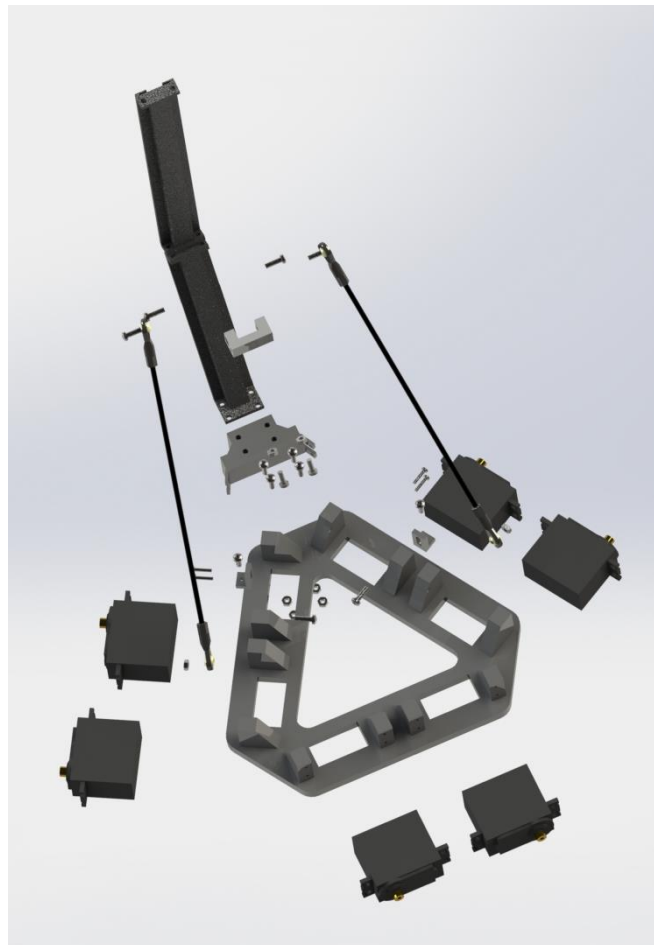


Figura 4. 13 Piezas Ball and Plate

Para la impresión de las piezas se utiliza una impresora 3D Prusa como se muestra en la figura 4.14, la cual se encuentra dentro del mismo laboratorio, mediante esta impresora se imprimieron todas las piezas necesarias para la nueva plataforma, estas impresiones ya que llevan una alta calidad, esto implica el grosor y la cantidad de filamento a usar, además de el detalle del terminado, lleva mucho tiempo imprimirlo.

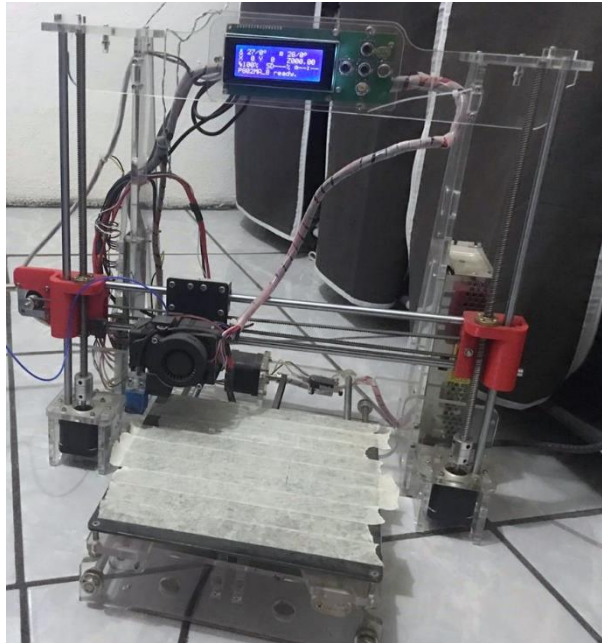


Figura 4. 15 Impresora 3D Prusa

Incluyendo en cada pieza una cama del mismo filamento ya que la impresora cuenta con algunas deficiencias en su cama, puesto a que por tanto desgaste térmico llego a un punto de pandeo en el cual llega a afectar a las piezas, por lo cual se llega a la conclusión de colocar una cama del mismo filamento la cual ayuda a que las piezas no terminen pandeadas como se muestra en la figura 4.15.



Figura 4. 14 Pandeo de pieza

La impresora 3D tiene un sistema guiado por coordenadas, esto significa que la impresora tiene 3 ejes de movimientos (x, y, z) en la cual el plano (x, y) son para el movimiento horizontal y vertical sobre la cama y el eje (z) es para la profundidad entre el extrusor y la cama.

El tipo de código que usa la impresora 3D es un G-code, este es un lenguaje mediante el cual las personas pueden decir a una máquina que hacer y cómo hacerlo, estas

instrucciones están definidas mayormente por instrucciones sobre a donde moverse, cuando moverse y que trayectoria seguir.

Este archivo es generado usando el software MatterControl el cual nos ayuda a importar la pieza realizada en SolidWorks, dentro de esta se colocan las especificaciones para las piezas tanto como la cantidad de filamento por capa y que tipo de relleno se le dará, la calidad, entre otras especificaciones más.

Ya una vez colocadas las especificaciones se genera una simulación de cómo se realiza la pieza ya en la impresora y a su vez se genera un archivo G-code el cual es el que se exporta para introducirlo en la impresora 3D, esto se aprecia como se muestra en la figura 4.15 cargando una pieza dentro del MatterControl.

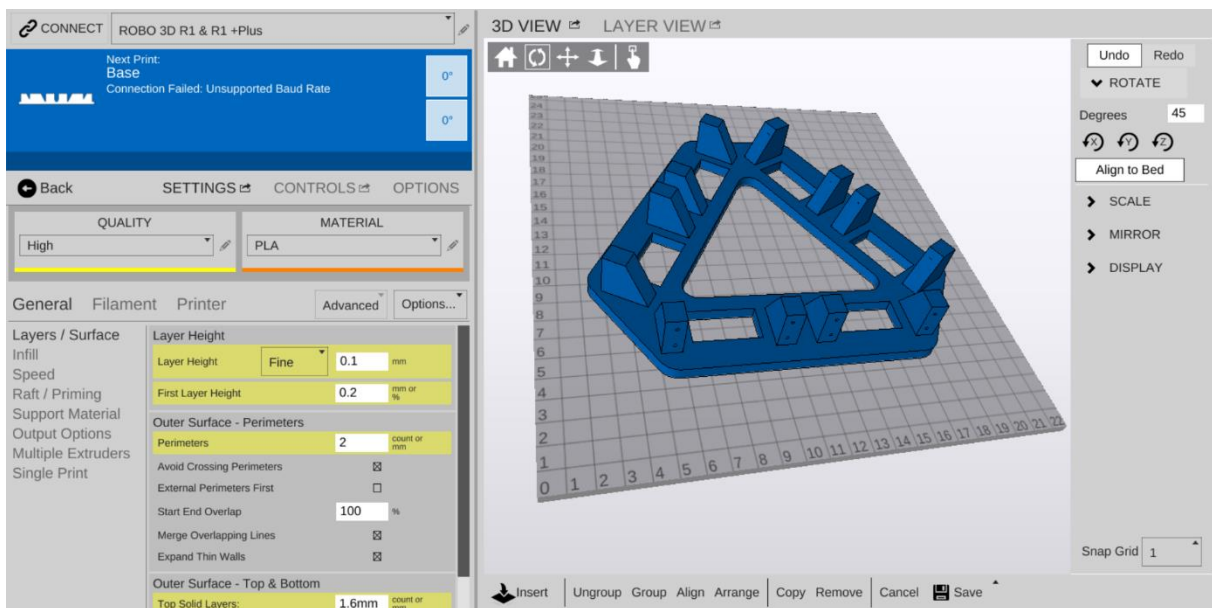


Figura 4. 16 MatterControl

Cómo se puede apreciar en la figura 4.16 se encuentran en la parte izquierda todas las características generales que tendrá la pieza, las velocidades de impresión, superficies, camas y demás. Dependiendo estas especificaciones influirá mucho en el tiempo de impresión y la cantidad de filamento. En la parte derecha se tienen herramientas para editar algunas propiedades de la pieza como la escala o si se quiere aplicar un efecto espejo, además de poder girarla en cualquier eje.

Ya una vez especificadas las diversas especificaciones necesarias para la pieza, se dirige hacia la pestaña Layer View, en la cual se generará el G-code, así como se muestra en la figura 4.17.

Dentro de la figura 4.17 se pueden apreciar diversas características que se tendrán al momento de la impresión, tanto como el tiempo de impresión, la longitud del filamento a utilizar, el peso de la pieza final, entre otras. De la misma manera se puede apreciar que esta pieza en particular tendrá 259 capas y a su vez la velocidad en cada capa y poder observar como la impresión ira imprimiendo en cada una de las capas.

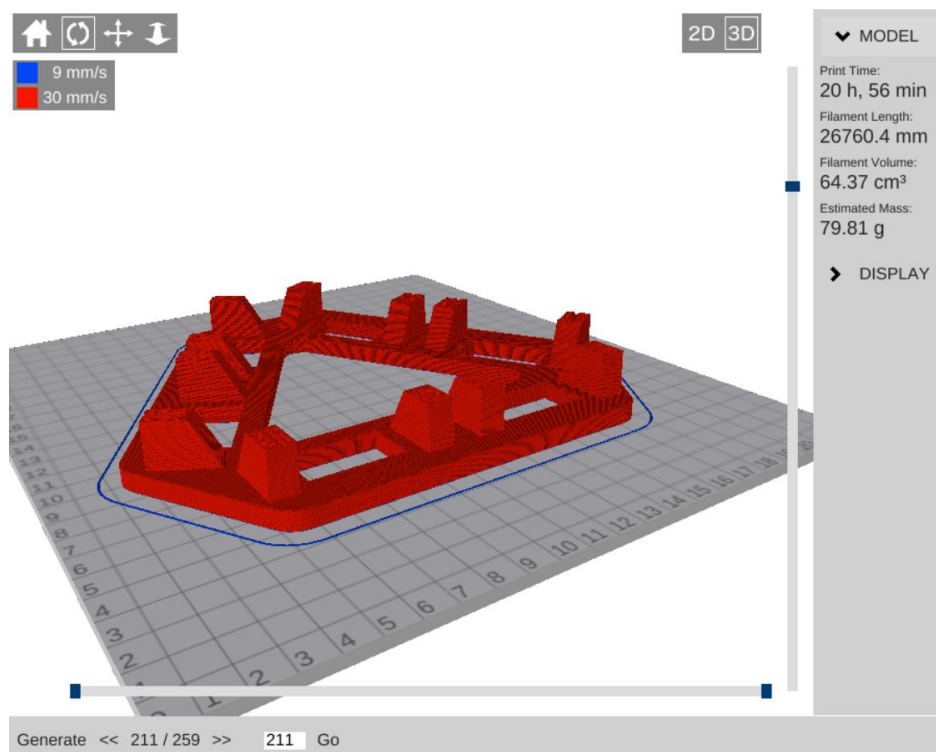


Figura 4. 17 Generación de G-code

Esta es una herramienta muy útil ya que de esta manera se programa el orden de las impresiones, aprovechando en lo absoluto el día y la noche, esto quiere decir que se puede dejar imprimiendo la impresora mientras se duerme sin necesidad de tener que despertar para colocar la pieza siguiente, a su vez a qué hora se deberá comenzar a imprimir la siguiente pieza, esto sin pérdidas de tiempo.

Ya una vez generado el G-code se extrae en una memoria microusb y esta es colocada en la impresora la cual comenzará a imprimir la pieza como se muestra en la figura 4.18, una recomendación al momento de imprimir es que se coloque una capa usando cinta de papel (preferentemente) esto para que el filamento tenga mejor agarre con la cama y a su misma vez para retirar la pieza de la cama sea de una manera más sencilla.

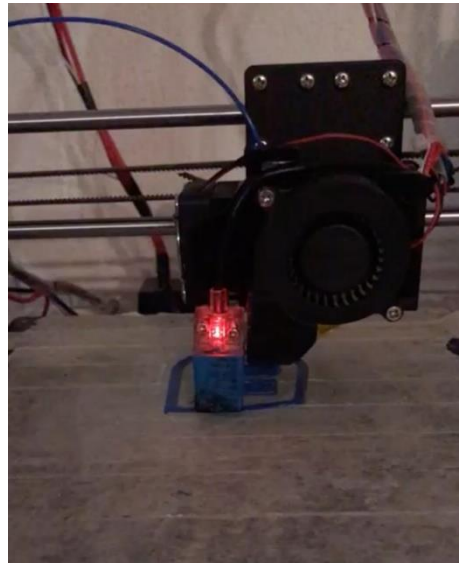


Figura 4. 18 Comienzo de impresión

Este proceso se realiza para todas las piezas necesarias para la construcción de la plataforma.

Ensamble de las piezas

Ya teniendo todas las piezas impresas es momento de comenzar el ensamble, para esto se comienza colocando los servomotores en la pieza que se muestra en la figura 4.19.



Figura 4. 19 Base servomotores

Al tiempo en que se colocan los motores se debe colocar las piezas complementarias en las cuales se montaran el PCA, pantalla led, control, base para la cámara y una base para las bolas que se llegaran a usar, esto lo puedes apreciar en la figura 4.20, estos se colocan a la par ya que van sujetos con el mismo tornillo y así se evita trabajo doble, esto se muestra con los círculos rojos, aquí es donde se agarran los motores y las piezas extras a la base.

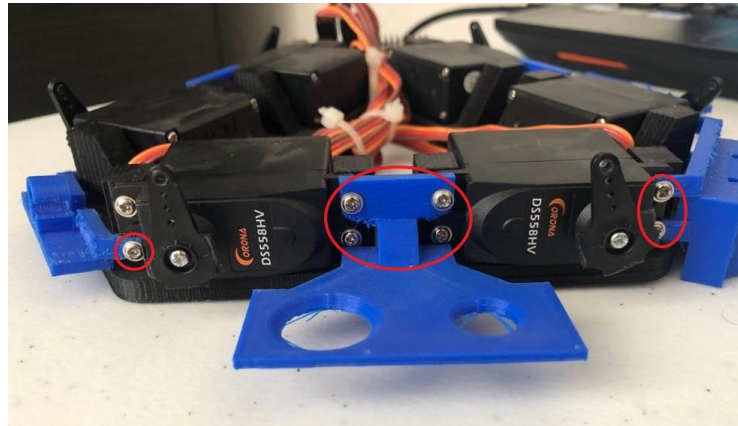


Figura 4. 20 Base ensamblada

Lo consiguiente es la conexión de los servomotores, en la cual se sugiere colocar un numero a cada servomotor como se muestra en la figura 4.21, para poder identificarlos al momento de querer hacer las pruebas y el script para su mismo movimiento.

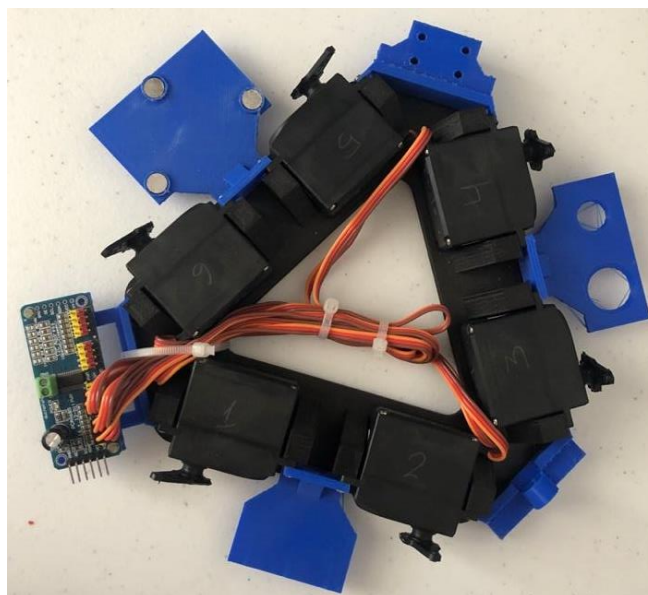


Figura 4. 21 Enumeración de servomotores

Estos servomotores se conectan en la PCA figura 4.22, que tienen 16 canales para 16 servos, para estos se debe identificar en que puerto se conecta cada servomotor y tenerlo considerado para el momento de la realización del código el cual se encargara del movimiento de los servomotores.

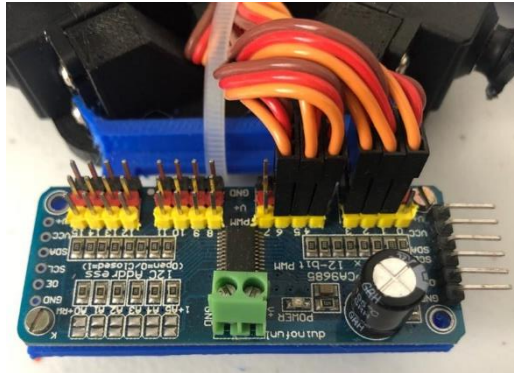


Figura 4. 22 Conexión PCA

Lo que sigue es armar la base para la cámara en la cual en este caso se usa una cámara diferente a la usada en la versión posterior, en este caso se usa una cámara TB-916776 figura 4.23, por diversos percances se opta usar esta sin importar tanto su baja calidad, se recomendó hacer el uso de un módulo de cámara de alta definición pero por cuestiones de la pandemia los envíos desde China estaban cancelados y no se logró obtener la misma.



Figura 4. 23 Cámara TB-916776

La base de la cámara se muestra en la figura 4.24, consiste de 5 brazos los cuales se dividieron en esas secciones ya que la impresora no lograba imprimir piezas tan grandes, de igual manera se imprime una extensión extra a las de la versión anterior, esto porque la cámara tiene diferente enfoque y ocupa más altura. De igual se incluyen un soporte para compensar el peso de la cámara y por último un soporte para la cámara y poder ensamblarla en la base de la cámara.



Figura 4. 25 Soporte cámara

Para lo consiguiente se usan brazos de fibra de carbono figura 4.25 los cuales ayudaran para hacer el movimiento del plato usando los servomotores, estos brazos son de 100 mm de longitud, son resistentes e indispensables para lograr el mejor movimiento del plato, que por consecuente llega a dar una buena estructura de la plataforma.



Figura 4. 24 Brazos fibra de carbono

Los brazos se colocan en cada servomotor unidos con una pija, esta no impide el movimiento del brazo ya que cuenta con un rodamiento y esto ayuda en el libre movimiento, del otro lado del brazo se coloca la base del plato, ya que este nos

ayudará a sujetar el plato y será la pieza en la cual el movimiento será reflejado y se lo transmitirá al plato donde se coloca la bola.

En la figura 4.25 se muestra cómo queda ya ensamblada la pieza.

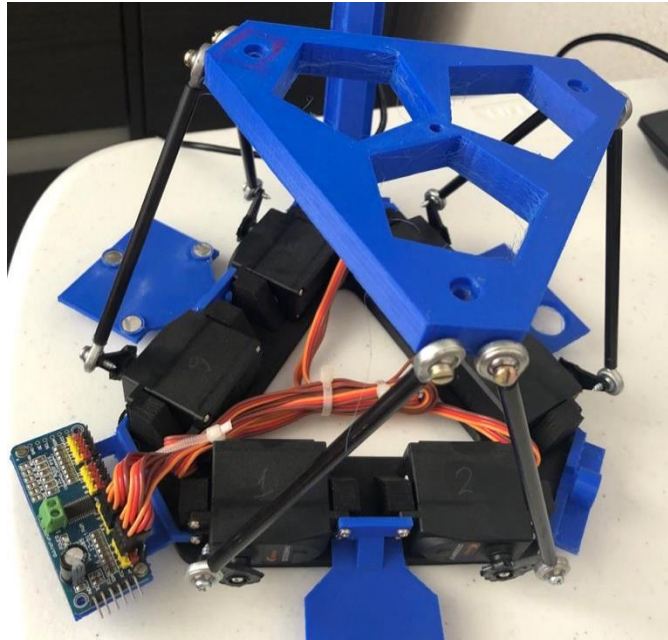


Figura 4. 26 Montaje soporte para la base del plato

Sobre esta misma se coloca una base figura 4.26 en la cual se montará el plato, es del mismo material ya que es ligero y a su vez fácil de montarle más piezas, en este caso soportes para que el plato quede bien ajustado y no le afecte alguna perturbación.

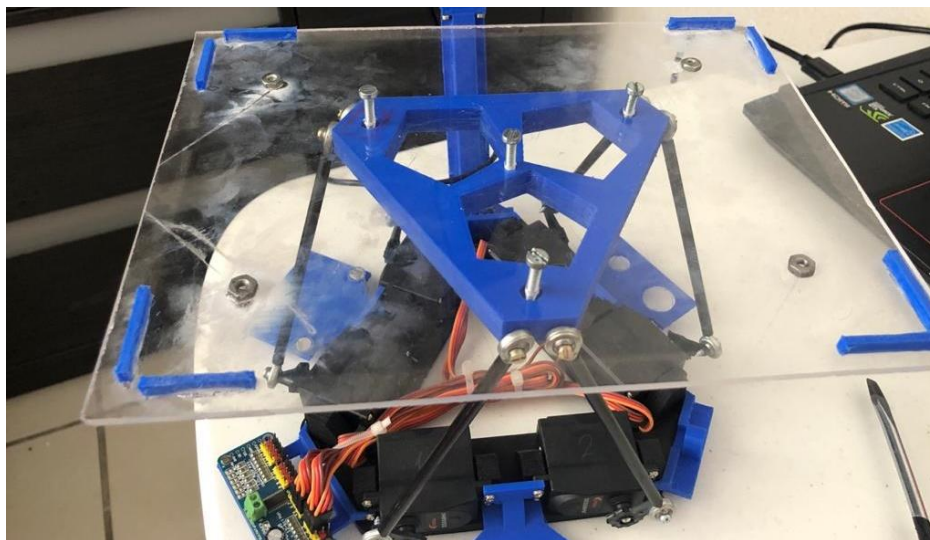


Figura 4. 27 Montaje base del plato

Una vez montado todo es cuestión de colocar el plato sobre su base y ya estará lista la plataforma, está terminada se puede observar en la figura 4.28.

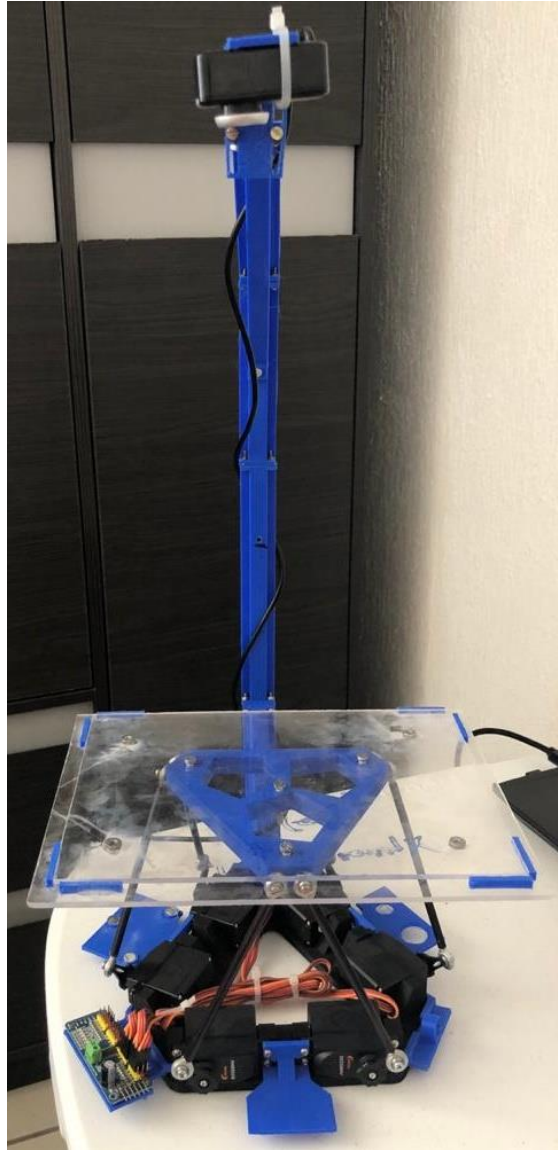


Figura 4. 28 Plataforma Ball and Plate

Scripting el código en arduino para el movimiento de los servomotores

Para la realización de este código es necesario considerar diferentes variables que serán de suma importancia tales como:

- Ancho de pulso para posición 0° servomotor
- Ancho de pulso para posición 180° servomotor
- Numero de servos

- Array especificando puertos del PCA con cada servomotor
- Array posición inicial de los servomotores para que la plataforma este completamente horizontal

Estas variables son las más críticas para lograr un buen posicionamiento de los servomotores y lograr los ángulos del plato deseados.

Inicialmente se comienza por instalar la librería que se necesitará llamada “*Adafruit PWM Servo Driver By Adafruit*” se ocupa ir al gestor de librerías siguiendo el path “programa > incluir librería > gestionar librerías” y abrirá el siguiente gestor como se muestra en la figura 4.29, aquí se coloca el nombre de la librería a buscar y se da click en instalar.

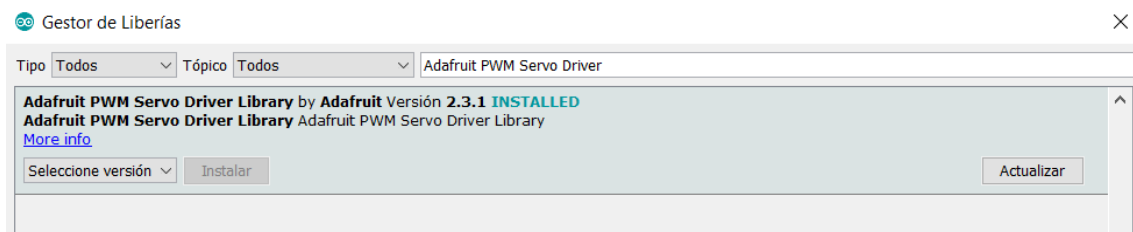


Figura 4. 29 Gestor de librerías

Dentro del código se tienen dos funciones las cuales son las que ejecutan dos funciones específicas una es para el inicializar los servomotores y la segunda es para fijar un el servomotor a un ángulo definido.

En la figura 4.30 se aprecia el código para inicializar los servos, para esto se usa un ciclo for para poder inicializar cada uno de los 6 servomotores que se tienen, teniendo una espera de 500 ms por servo.

```

void servos_init(void) {
    for(int i=0; i<=5; i++){
        setServo(pca_channels[i], servo_home[i]);
        delay(500);
    }
}

```

Figura 4. 30 Inicialización de servos código arduino

En la figura 4.31 se tiene el código en el cual se puede fijar cualquier servo a una posición indicada ya en ángulos no indicando el pulso, para esto se usa un mapeo en el cual se ingresan tres datos, en el primer dato se ingresa la variable de referencia que contiene el ángulo deseado, enseguida van los datos dentro los cuales se tiene referenciado el ángulo, un mayor y un menor (es el rango del ángulo de trabajo) y por

```
void setServo(uint8_t n_servo, int angulo) {  
    int duty;  
    duty=map(angulo,0,180,pos0, pos180);  
    servos.setPWM(n_servo, 0, duty);  
}
```

Figura 4. 31 Fijación del ángulo código arduino

último se cuenta con los datos reales de los pulsos (rango de trabajo del servomotor) de esta manera no se llegaría a tener parámetros fuera de rango.

En la parte del void setup es solamente inicializar las variables e iniciar los servos y su frecuencia de trabajo, ya el resto del código son diversos casos por los cuales se realizan diferentes tareas, así se recibirán datos y el arduino podrá realizar diversas tareas dependiendo el comando ingresado, haciendo una función de DAQ.

Conexión del PCA9685

Este dispositivo es muy importante ya que se es capaz de controlar 16 servomotores y para hacer el control de los mismos se utiliza una comunicación I2C por la cual solo es necesario dos canales a él una con los datos (pulsos mandados desde el arduino) y otro con un reloj, dando cuenta que también se debe conectar una alimentación para alimentar el circuito.

Para los servomotores se tiene una alimentación externa ya que requiere una corriente y un voltaje mayor, como se aprecia en la figura 4.32 se tienen diversas entradas por



Figura 4. 32 PCA9685

varios pines al lado izquierdo de la figura, aquí se conectan el SDA y SCL al igual que ambas alimentaciones de voltaje Vcc es el voltaje de la placa de 5v y en V+ un voltaje de 6v (este ya que es el voltaje nominal de los servomotores).

La conexión de este con el arduino es usando los siguientes puertos

- Vcc > +5v
- GND > GND
- SDA > Analog 4
- SCL > Analog 5

De esta manera y con la librería ya descargada de Adafruit hace la conexión entre ambos y hace la comunicación de lo que deseamos por medio del protocolo ya establecido de uso (I2C). Cabe mencionar que la forma de conexión de los servomotores se muestra en la tabla 4.1.

Numero de servomotor	Salida PCA
1	1
2	2
3	3
4	4
5	5
6	6

Tabla 4. 1 Conexión PAC – Servomotores

Ajuste de los servomotores

Este paso es importante para el cálculo de dos parámetros a utilizar en los scripts a realizar, estos parámetros se pueden obtener utilizando uno de los casos realizados en el código de arduino para el movimiento de los servomotores.

El código es el que se muestra en la figura 4.33. En este caso Duty es el nombre que se le da al valor del ciclo de la frecuencia dada para generar un respectivo movimiento en un sentido indicado.

```
case 'd': //Find Duty using serial port
  if(bufferSize == 2){
    number_servo=Numbers[0];
    int duty=Numbers[1];
    servos.setPWM(number_servo, 0, duty);
    UART_PORT.print("Servo: ");
    UART_PORT.print(number_servo);
    UART_PORT.print(" Duty: ");
    UART_PORT.println(duty);
  }
  break;
```

Figura 4. 33 Encontrar Duty código arduino

Para estar mandando los datos para realizar las pruebas y calcular el Duty de cada servomotor se utiliza el monitor de comandos de arduino, esto ya que el código recibe los datos mediante comunicación serial, el comando a utilizar será “dx,z” donde la letra d es para hacer mención al caso que se usara, x es el número del servomotor a probar y z es el Duty que se probará. Ya que estos servomotores no cuentan con una referencia se buscarán los parámetros a manera de prueba y error.

Cálculo de los 0° - 180° naturales de cada servomotor (rango de trabajo)

Primeramente, se comienza obteniendo el Duty para obtener los ángulos 0° y 180° dependiendo del motor (para esto se habla de 3 secciones en la base de la plataforma, cada una con dos servomotores, uno denominado izquierdo y otro derecho) como se muestra en la figura 4.34 se observa cómo está posicionado el servomotor derecho a sus 0° y el servomotor izquierdo a 180°, esta diferencia de los grados es ya que los



Figura 4. 34 Primera adquisición de rango de trabajo

motores se encuentran volteados uno del otro, por eso esta diferencia de 180° entre ambos servomotores, esto tomando en cuenta que serían los puntos de origen o de partida para los servomotores.

Una vez ya tenido anotado el cálculo anterior se realizará un movimiento para encontrar el otro punto del rango de trabajo que sería en el caso como se muestra en la figura 4.35 en el servomotor derecho 180° y en el servomotor izquierdo 0° .



Figura 4. 35 Segunda adquisición de rango de trabajo

Una vez ya realizados estos cálculos se logra apreciar que en todos los servomotores el rango de trabajo es el mismo, no varía o difiere mucho entre todos los resultados, por lo consecuente en la tabla 4.2 se muestran los valores del rango de trabajo.

Duty	Valor
Duty para ancho de pulso de 0°	130
Duty para ancho de pulso de 180°	510

Tabla 4. 2 Valores del rango de trabajo

Cálculo de los 0° o 180° y 90° horizontales para cada servomotor

Para calcular estos puntos se utiliza un comando diferente que el usado en la parte anterior, aquí se usa el comando `s` y ya no será el valor de Duty a utilizar si no que esta vez se utilizarán grados del rango de trabajo del servomotor.



Figura 4. 36 Ángulo 0° y 180° horizontales

Como se aprecia en la figura 4.36 se calculan los ángulos 0° y 180° pero referentes a la plataforma (ángulo horizontal) el ángulo de 0° lo representa el servomotor de la derecha y el de 180° el servomotor de la izquierda.

Ya lo consiguiente es sacar el ángulo de 90° que para ambos será el mismo respectivamente hablando de la posición no como en el caso anterior, así como se aprecia en la figura 4.37.



Figura 4. 37 Ángulo 90° horizontal

Los ángulos adquiridos se muestran en la tabla 4.3 los cuales serán necesarios para diversas partes del script a desarrollar, se marca con una X en la casilla donde no se logra este valor.

Numero de servomotor	Ángulo 0° horizontal	Ángulo 180° horizontal	Ángulo 90° horizontal
1	X	140°	56
2	30°	X	120
3	X	140	56
4	40°	X	130
5	X	152	62
6	24°	x	114

Tabla 4. 3 Adquisición ángulos horizontales

Scripting el código en Python para leer un archivo JSON

Un archivo JSON (JavaScript Object Notation) es un formato estándar abierto basado en texto que se utiliza para serializar y transmitir datos estructurados entre servidores, este es fácil de leer y escribir.

Este archivo está orientado a objetos por lo cual se puede colocar un objeto o varios dentro de un archivo JSON, de igual manera se puede colocar varios atributos dentro de cada objeto, en el caso de los nombres o ID se colocan primero y entre comillas y después del doble punto se añade el valor de la variable (boolean, string o numeric), en el caso del string se utiliza comilla doble, en la figura 4.38 se muestra el archivo JSON con todas las características de la plataforma.

```

1  {
2    "Platform":{
3      "base_length":92.4,
4      "servo_links":[17,115],
5      "scrap_x":15,
6      "scrap_y":15,
7      "centroid_dist": 59.1,
8      "axis":{
9        "x":110,
10       "y":110,
11       "z":220
12     },
13     "home":{
14       "translation":{
15         "x":0,
16         "y":0,
17         "z":117
18       },
19       "angles":{
20         "yaw":0,
21         "pitch":0,
22         "roll":0
23       }
24     }
25   },
26
27   "Servos":{
28     "min_servo_signal": [0,66,0,64,0,70],
29     "max_servo_signal": [108,180,106,180,104,180],
30
31     "min_signal_degree": [12,83,7,85,5,77],
32     "max_signal_degree": [97,170,95,173,90,157],
33
34     "pca_channels": [4,5,6,7,0,1]
35   }
36 }
37 }

```

Figura 4. 38 Características de la plataforma archivo JSON

Se opta por hacer uso de esta herramienta ya que al desear dejar el código cerrado para no realizar modificaciones en el, era necesario optar por una manera externa para ingresar los datos de la plataforma de manera indirecta, es por esto que la mejor opción encontrada es usar este tipo de archivos.

Para realizar la adquisición de datos se debe leer el archivo JSON primero para esto se usa el código de la figura 4.39, se aprecia que se manejan excepciones en la cual si el path del archivo JSON no es válido este generara un error y terminara el programa, pero si se ingresa el correcto leerá todos los datos en él y los guardara en un diccionario separando los del objeto Platform y Servos.

```
def read_keys(self):
    """
    JSON File
    """
    if _os.path.exists(self.json_path):
        json_file = open(self.json_path)
        platform_json = _js.load(fp=json_file)
        self.platform_dic = dict(platform_json["Platform"])
        self.servos_dic = dict(platform_json["Servos"])
    else:
        raise Exception("Path Not Valid")
```

Figura 4. 39 Función leer archivo JSON

Y para poder guardar cada una de las variables del archivo JSON se genera otro método como se muestra en la figura 4.40, en esta se toma cada diccionario y de comienza a desglosar guardando cada una de las variables por separado, es por esto que este método suele ser extenso dependiendo del número de variables o parámetros que se ingresen.

Se dividió los datos que se ingresan por el archivo JSON en dos secciones, una para todos los datos de la plataforma, ya sean algunas medidas importantes a considerar durante los cálculos.

En la segunda parte se creó un apartado para todos los datos de los servomotores, tales como sus rangos de trabajo y los ángulos de operación mínimos y máximos.

```

def init_variables(self):

    self.base_length = self.platform_dic["base_length"]
    self.servo_links = self.platform_dic["servo_links"]
    self.scrapt_x = self.platform_dic["scrapt_x"]
    self.scrapt_y = self.platform_dic["scrapt_y"]
    self.centroid_dist = self.platform_dic["centroid_dist"]
    self.axis_x = self.platform_dic["axis"]["x"]
    self.axis_y = self.platform_dic["axis"]["y"]
    self.axis_z = self.platform_dic["axis"]["z"]
    home_translation = self.platform_dic["home"]["translation"]
    home_angles = self.platform_dic["home"]["angles"]
    self.home_translation = [
        home_translation["x"],
        home_translation["y"],
        home_translation["z"]
    ]
    self.home_angles = [
        home_angles["yaw"],
        home_angles["pitch"],
        home_angles["roll"]
    ]
    self.min_servo_signal = self.servos_dic["min_servo_signal"]
    self.max_servo_signal = self.servos_dic["max_servo_signal"]
    self.min_signal_degree = self.servos_dic["min_signal_degree"]
    self.max_signal_degree = self.servos_dic["max_signal_degree"]
    self.pca_channels = self.servos_dic["pca_channels"]

```

Figura 4. 40 Función para inicializar variables de la fuente JSON

Scripting el código en Python para la identificación del sistema y posición de servomotores

Esta es la parte importante donde se realiza un script en Python el cual será capaz de identificar el sistema, y calcular la posición de los servomotores dependiendo de un ángulo deseado, estos ángulos a utilizar son los ángulos Euler o ángulos de rotación, además de usar los ángulos de traslación.

En la figura 4.41 se muestra el diagrama de ambos tipos de ángulos los 3 más conocidos son (x, y, z) estos están de colores (rojo, amarillo, azul) respectivamente,

estos parámetros nos dan a conocer las dimensiones que puede tomar un objeto, así como su posición en el espacio.

Los otros 3 ángulos son muy utiliza dentro del código ya que serán los 3 ángulos a estar trabajando durante el control de la bola, estos son (Yaw, Pitch, Roll) de colores (naranja, azul agua, verde) respectivamente, estos ángulos sirven para especificar la orientación del sistema de referencia.

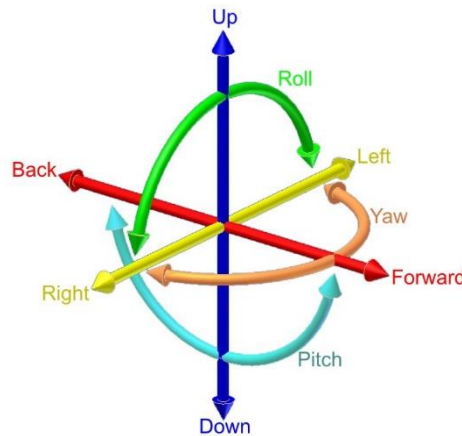


Figura 4. 41 Ángulos de rotación

Una vez teniendo en cuenta el cómo funcionara el sistema es momento de iniciar el script, para esto sabemos que usaremos muchos puntos los cuales tendrán las mismas características unos a los otros, las características principales es que todos los puntos

```
class Point:
    def __init__(self,cordinates):
        if isinstance(cordinates,list):
            if len(cordinates) == 3:
                self.x = cordinates[0]
                self.y = cordinates[1]
                self.z = cordinates[2]
            else:
                raise Exception("Must have 3 items")
        else:
            raise Exception("Must be a List")

    def get_list(self):
        return [self.x,self.y,self.z]
    def __str__(self):
        return "Point(x=%d, y=%d, z=%d)\n"%(self.x,self.y,self.z)

    def __repr__(self):
        return str(self)
```

Figura 4. 42 Clase punto

se rigen por coordenadas, así que es la propiedad en común, para esto se crea una clase en la cual se agregan excepciones de error si es que la coordenada no está dentro de una array o no cuenta con los 3 puntos, esta clase se encuentra en la figura 4.42.

Ahora se crea una clase tipo plataforma, que se quiere decir con esto, que dentro de esta clase se hará toda la identificación del sistema, y se tendrán todos los atributos de la caracterización de la plataforma, para comenzar se tiene el constructor de la clase en la cual se inicializan todas las variables a utilizar y a su vez se puede agregar una condición de si se quiere o no hacer el plot para poder observar una ventana grafica generada por Matlab para Python, este se muestra en la figura 4.43.

```
class Platform_config:
    def __init__(self,json_path,plot=False):
        self.json_path = json_path
        self.read_keys()
        self.init_variables()
        self.base_points = calculate_base_points(self.base_length)
        self.plot = plot
        if plot:
            plt.ion()
            self.fig = plt.figure()
            self.ax = self.fig.add_subplot(111,projection='3d')
            self.draw_axis()
            self.set_platform(self.home_angles,self.home_translation)
```

Figura 4. 43 Constructor de la clase plataforma

Dentro de este se hace el primer calculo que es el cálculo de los 6 puntos de cada servomotor (cálculo de los puntos de la base), esto se realiza con el método que se muestra en la figura 4.44, en la cual se usa solo un dato que es la longitud de la base, que esta es el radio, de una semi circunferencia que se traza, ya es cuestión de rotar dentro de un ciclo for los ángulos en los que se encuentran cada uno de el origen del movimiento del servomotor, el acomodo de cada ángulo debe ser tomando de base cual es el frente de la plataforma, y tomando en cuenta que numero de servomotor

```
def calculate_base_points(radius):
    points = []
    for angle in [240,300,0,60,120,180]:
        points.append(Point([radius*_math.cos(_math.radians(angle)),radius*_math.sin(_math.radians(angle)),0]))
    return points
```

Figura 4. 44 Calculo puntos de la base

estas usando como fue la numeración de los servomotores (1-6) para que esta coincida en el cálculo de los puntos.

Ya una vez teniendo esto se realiza un home en la plataforma, esto quiere decir que cuando se inicialice cualquier plataforma tome esta posición, para esto dentro del constructor se hace un set de la plataforma en la cual se le dan los ángulos de rotación y traslación del home.

```
def set_platform(self, angle, translation):
    rc = self.update_servo_angle(angle, translation)
    if self.plot and rc:
        self.draw_axis()
        draw_by_points(self.base_points, self.ax, self.fig, 'orangered')
        draw_by_points(self.plate_points, self.ax, self.fig, 'dodgerblue')

        draw_servo(self.base_points, self.plate_points, self.servo_links[0], self.servo_angles_1, self.ax, self.fig)
        self.fig.canvas.draw()
```

Figura 4. 45 Método para fijar posición

Ahora hablando de la función que hace el set de la plataforma figura 4.45, dentro de este se realizan diversos cálculos, el primero es cargar los datos del ángulo de rotación y traslación que se desean para poder realizar cada operación y obtener las coordenadas de los puntos del plato y después obtener los ángulos necesarios para cada servo.

Iniciando con todo se realiza el cargar los ángulos del servo, para esto se usan los siguientes métodos de la figura 4.46, de los cuales nos regresaran un valor final de verdadero si nuestra plataforma es capaz de optar en esa posición (que los servomotores puedan adoptar los ángulos calculados) y si esta es falsa nos arrojará un mensaje el cual nos dirá que este ángulo a adoptar no es posible.

```
def update_plate_points(self, angle, translation):
    self.plate_points = plate_points(self.centroid_dist, self.scrapt_x, self.scrapt_y, angle, translation)

def update_servo_angle(self, angle, translation):
    self.update_plate_points(angle, translation)
    try:
        self.servo_angles_1, self.servo_angles_2 = servo_angle(self.plate_points, self.servo_links, self.base_points)
        return True
    except ValueError:
        print("It's not possible to set current position on this plataform")
        return False
```

Figura 4. 46 Cargar puntos de servos y plato

El primer cálculo a hacer es el de las coordenadas del plato, en este se realiza una traslación, esto para poder hacer un movimiento desde el centro hasta los dos puntos de agarre que van sujetos a los brazos, ya después se realiza una rotación para la cual dependiendo los ángulos Euler que deseemos es la posición que tomara, esto recordar que es por cada uno de los puntos.

Ya una vez teniendo estos puntos, el punto de los servomotores y los puntos del plato, junto con las medidas de los dos brazos es momento de con la ayuda de la cinemática inversa calcular los dos ángulos theta el cual uno se usará para el posicionamiento de los servomotores.

Esto es todo lo que realiza la plataforma, son los respectivos cálculos necesarios, ya con el uso de este método dentro de la clase, se puede colocar en la posición deseada la plataforma, la cual será usada para poder aplicar controles, ya que esta cumplirá con la tarea de mandar la indicación al sistema para que este de como resultado lo que se espera.

Actividades adicionales

Comprensión de la sintaxis para POO en Python

La sintaxis que usa Python para la programación orientada a objetos (POO) no es tan compleja, pero si un poco tediosa las primeras veces que se trabaja con ella. Para comenzar lo principal que se necesita saber es que POO es una estructura con la cual se pueden realizar algoritmos muy extensos en códigos demasiados simples y de menor líneas de código, esto comienza con la realización de una clase la cual contendrá todos nuestros objetos los cuales pertenezcan a esta clase, en este caso una plataforma Ball and Plate.

Lo básico que se debe comprender es el cómo funciona la parte más importante de una clase, esta es el constructor, en el constructor de una clase lo que se realiza es inicializar las variables o atributos de la clase, aquí es donde se toman los valores

iniciales o se realizan diversas operaciones las cuales se deben realizar de primera mano, como un ejemplo, en la plataforma dentro del constructor se realiza la lectura del archivo JSON y este guarda todas sus variables dentro de dos diccionarios para después inicializar independientemente cada una de las variables.

Para identificar el constructor en Python se genera como una función, salvo que el nombre de esta función es `__init__`, este nombre nos indicara al crear un objeto de la respectiva clase entrara a este método, que nos creara el objeto ya con sus variables inicializadas.

Otra de las cosas importantes es que para cada variable o atributo que pertenezca a la clase se debe dar como referencia al inicio de la variable el comando "self." Y agregar después del punto el nombre de la variable, así de esta manera la variable quedara guardada y se podrá acceder a ella en cualquier método dentro de la clase, de una manera quedara como variable global pero dentro de la clase.

Al igual que las variables dentro de cada método o comportamiento, se debe agregar self entre paréntesis, para hacer referencia a que este estará siempre presente dentro de la clase, esta es una de las reglas a seguir para usar POO en Python.

Estas son las características más relevantes dentro de POO en Python las cuales nos ayudan a poder generar un buen script y a su vez entendible y funcional para cualquier tipo de adecuación futura que se desee realizar.

CAPÍTULO V
Resultados Obtenidos

Resultados

La construcción de la plataforma fue simple, esto pues, se contó con un video del ensamble de las piezas y como resultado del ensamble se obtuvo la plataforma ya construida que se muestra en la figura 5.1, a esta se le adaptó una nueva cámara TB-916776, ya que la base de la cámara era para una cámara diferente, por lo cual solo se le adaptó una pieza extra ya que la resolución de esta era diferente. De lo demás todo quedó bien fijado y sin necesidad de hacer ninguna modificación.

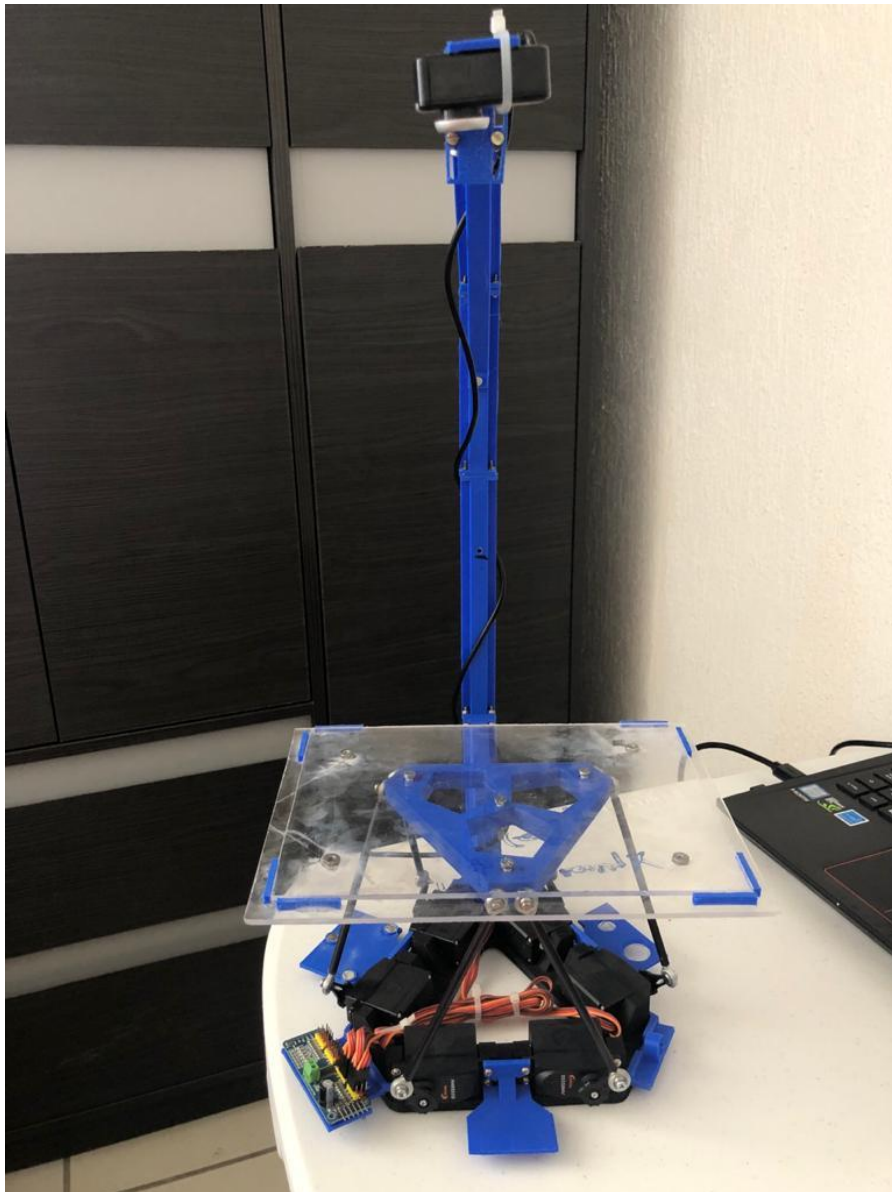


Figura 5. 1 Plataforma Ball and Plate

La cinemática inversa es la técnica usada para realizar los cálculos pertinentes para saber los ángulos requeridos de los servomotores para poder llevar al plato a una posición deseada, independientemente que es un cálculo complejo ya que consiste en la resolución de una serie de ecuaciones cuya solución normalmente no es única, esto nos dice que existen soluciones múltiples para que los brazos de la plataforma tomen las coordenadas indicadas y se obtenga la posición deseada con el movimiento de todos los servomotores.

Para las múltiples formas de llegar a la posición deseada en este caso ya que son solo dos grados de libertad se pueden tener dos posibles resultados como se muestra en la figura 5.2 y para la selección de estos se opta dependiendo el servomotor a utilizar y si le es posible a nuestro servomotor llegar a este punto, para esto se toma el punto P1 y se saca el ángulo que debe tomar.

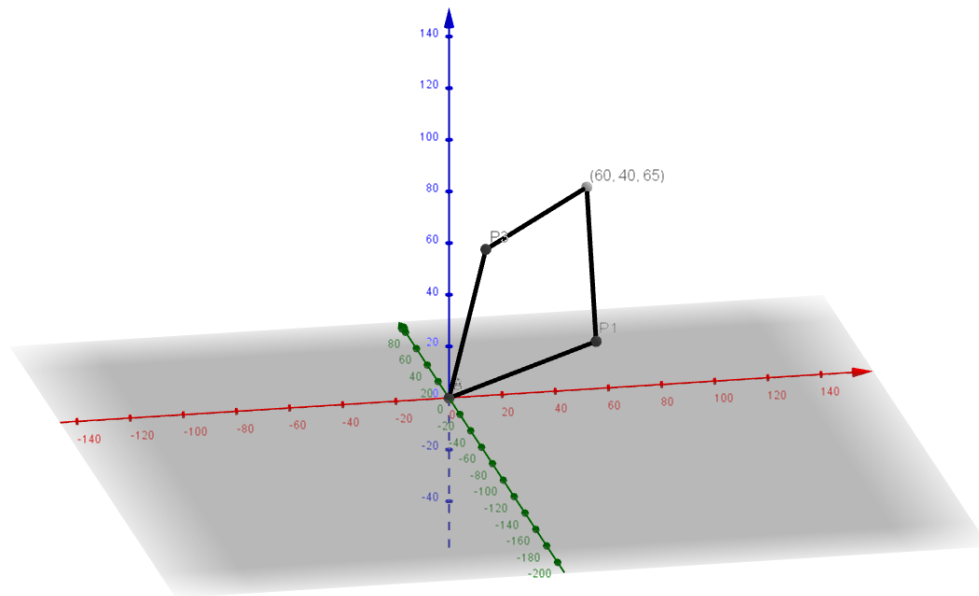


Figura 5. 2 Resultado cinemática inversa

Esto es por lo cual la cinemática inversa es muy útil dentro de la realización del script ya que es un principio de la robótica.

Este conocimiento de cómo realizar los cálculos usando cinemática inversa es importante conocerlas para la realización del script en Python ya que se deben recrear todas las fórmulas dentro de un método el cual realiza las operaciones para los puntos

deseados y consecuente nos arroje los valores (ángulos) que deben tomar los servomotores para poder colocar el plato de la plataforma en la posición indicada. En la realización del script de Python el primer cálculo realizado es el de los puntos de la plataforma, estos puntos se logran dibujar en un plot de Matlab el cual muestra los puntos y es capaz de dibujarlos como se muestra en la figura 5.3 así de esta manera se puede apreciar el cómo el código está calculando los puntos y de esta manera validar si los cálculos están correspondiendo a las especificaciones que se le están dando.

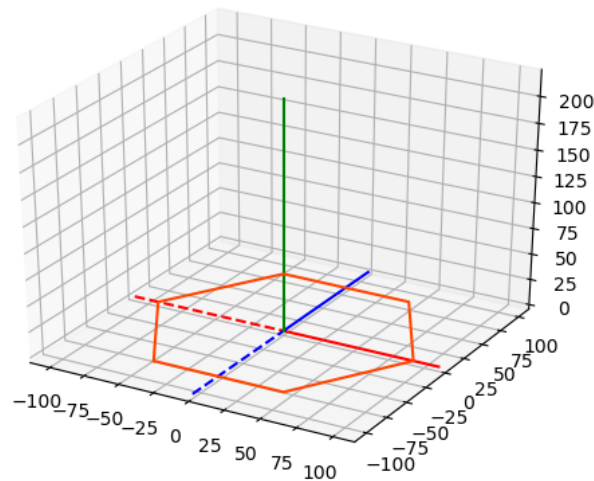


Figura 5. 3 Plot puntos de la base de la plataforma (Naranja)

Ya para la base del plato se aplican transformaciones lineales de rotación y traslación para hacer el cálculo de la posición de los puntos, en el punto donde se hace la unión entre la base del plato y los brazos que están sujetos a cada servomotor, como se

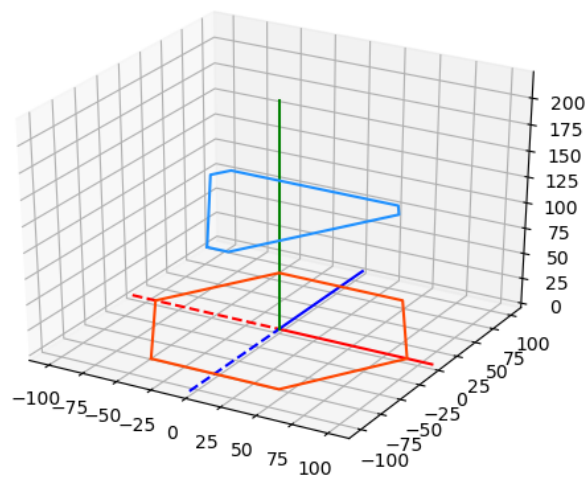


Figura 5. 4 Plot de la base del plato (Azul)

aprecia en la figura 5.4 al calcular cada uno de los puntos se realiza un mapeo en el cual nos mostrara como es que queda el plato, este de color azul.

Una vez que se adquieren estos dos puntos, el del origen del servo y el de la base, junto con los largos de las dos articulaciones a usar para el movimiento, se puede realizar por medio de la cinemática inversa el cálculo del ángulo al que ocupa estar el servo para que el plato tome los valores que se le seleccionaron dentro de la función creada `set_platform()`, una vez generado el objeto plataforma este se inicializa con los valores de home que se agregan dentro del archivo JSON, como se muestra en la figura 5.5 ya los ángulos mapeados y calculados los ángulos que deben tomar los servomotores para que la plataforma este con los valores de home asignados.

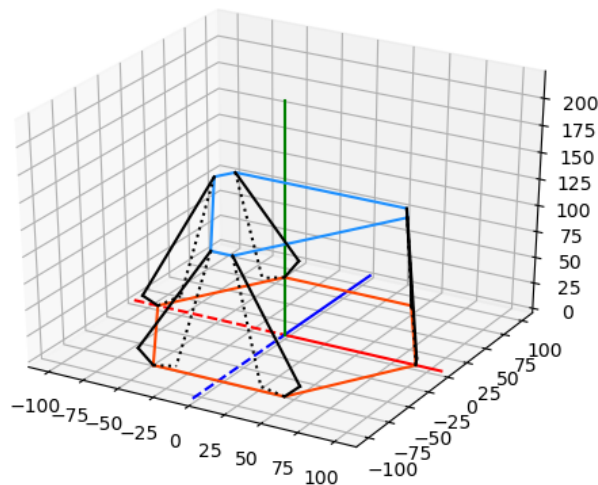


Figura 5. 5 Plot plataforma completa

Ya con este modelado de la plataforma se puede asignar cualquier valor de los ángulos que se desean obtener y con este la plataforma hará los diversos caculos y se obtendrán los valores que los servomotores deben de tomar para que la plataforma real tome esta posición.

Respuesta a la hipótesis

Hacienda el uso de la cinemática inversa se es posible hacer la planificación de movimientos de un brazo robótico, mediante el uso de las ecuaciones cinemáticas de un robot, dentro del script se realizaron todos los cálculos necesarios para obtener las

posiciones de los 6 servomotores esto para llegar a la posición indicado, esta proporcionada de la función en la cual funge el rol de entrada de variables en el script.

Para la realización de los cálculos se realiza en tres planos, un plano para cada par de servomotores, por lo cual se ocupó aplicar una rotación del plano para realizar el calculo de los 3 planos generado, y un calculo en la traslación para conectar los tres puntos de agarre del plato y conectarlo con cada eje de cada brazo, así de esta manera las ecuaciones son más simples y fáciles de procesar por un ordenador.

Mandando las posiciones de los servomotores por serial hacia un microprocesador ATmega el cual tiene un alto poder de procesamiento por lo cual no tiene problemas para manipular los servomotores.

Agregando que haciendo el uso de un archivo JSON para la proporción de datos del sistema Ball and Plate, no se es necesario hacer modificaciones del script por lo cual se puede encapsular y hacerlo genero para cualquier tipo de sistema Ball and Plate con 6 servomotores.

Con la conclusión que después de todo el análisis realizado y el desarrollo del proyecto, se cumple la hipótesis antes planteada.

CAPÍTULO VI
Alcances y limitaciones

Alcances

El beneficio que obtuvo en este caso el tecnológico, es el tener un equipo el cual se pueda usar para la realización de prácticas de control, tanto para la prueba de diferentes controladores, como la sintonización de los mismos, e igual ya tener un script genérico por si en un futuro se necesita hacer más plataformas a 6 ejes, se pueda hacer uso del mismo script solo necesario modificar las especificaciones de la plataforma si es que tiene diferencias con la anterior.

Limitaciones

Para la realización del proyecto se encontró con el primer inconveniente que fue la falta de una plataforma Ball and Plate, por consecuente se tuvo que hacer una reimpresión basándose en una que ya se había trabajado anteriormente dentro del laboratorio, esto tomo algo de tiempo hacer ajustes de las piezas ya que no se usaron los mismos servomotores, cámara, entre otros materiales.

Ya enseguida, una vez ensamblado todo, se presentó el problema con las Raspberry 3 a utilizar ya que no se contaba con el suficiente conocimiento del sistema operativo Linux, ya que este se trabaja mediante comandos, entonces se realizó un cambio a usar un ordenador normal con el sistema operativo Windows, una vez esto un poco de dificultad al hacer el script en Python ya que no se tenía el conocimiento suficiente para realizarlo usando POO, se tenía un conocimiento usando Java pero la sintaxis es un poco diferente entre ambos lenguajes.

Por último, se presentó la pandemia del COVID-19 la cual nos impedía poder desplazarnos con mejor facilidad hacia los puntos de encuentro con personas que nos estaban asesorando ya sea maestros o compañeros los cuales nos ayudaban en la parte de codificar tanto como en la redacción del presente documento.

CAPÍTULO VII

Conclusiones y recomendaciones

Conclusiones

Dentro de la realización de este proyecto me lleve muchas grandes sorpresas que no esperaba, al inicio de todo se tenía un planteamiento de un proyecto distinto a este, pero una vez que me fui adentrando al mismo comencé a notar muchas deficiencias que tenía una plataforma previa que fue diseñada, entonces ya que se tuvo que realizar una reimpresión basándonos en la anterior aproveche para hacer unas modificaciones y hacer cambio por unos mejores motores, algo tra cámara, ya que no contaba con una cámara como la que se usó anteriormente, por ende se tomó esa decisión.

Hablando en el tema de la codificación primero se tuvo que comprender más a fondo la cinemática inversa, analizar y comprender como no ayudaría para nuestra plataforma, una vez se comprendió fue turno de pasar todo esto a un código y crear un script que realizara todos los cálculos automáticamente y poder calcular la posición de los servomotores, fue de primero algo difícil, pero se logró hacerlo.

Dentro de la codificación me fue un poco difícil adaptarme a la sintaxis de Python ya que yo últimamente he estado trabajando con java, este lenguaje es uno de los lenguajes que está orientado a objetos por lo cual no me fue difícil comprender como funciona POO, ya solo fue comprender como es la estructura en Python de POO y ya lo demás no fue tan complicado.

Recomendaciones

De las principales recomendaciones que les daría es que de preferente se monte la plataforma en una base de madera o alguna mesa, para de esta manera tener mejor estabilidad ya que suele haber mucho movimiento con el plato por lo consecuente la cámara llega a desestabilizarse y la imagen se distorsiona y por consecuente el control no llega a ser muy bueno, de igual manera colocar mejores soportes en las impresiones que se utilizan para el soporte de la cámara.

Referencias bibliográficas

Challenger Pérez, I., Díaz Ricardo , Y., & Becerra García, R. A. (2014). El lenguaje de programación Python. Ciencias Holgín.

Factory, G. (14 de Abril de 2015). Geek Factory. Obtenido de http://www.nxp.com/documents/data_sheet/PCA9685.pdf

González, A. (Diciembre de 2013). Universidad Técnica Federico Santa. Obtenido de <http://www2.elo.utfsm.cl/~mineducagv/docs/ListaDetalladadeModulos/servos.pdf>

I. Grossman, S. (2015). Matemáticas 4: Álgebra lineal. McGraw-Hill.

Kumar Saha, S. (2010). Introducción a la robótica. McGraw-Hill.

Moreno Muñoz, A., & Córcoles Córcoles, S. (2017). En Aprende Arduino en un fin de semana (págs. 19-22).