ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DISEÑO DEL SISTEMA DE CONTROL BASADO EN SOFTWARE LIBRE PARA UN BRAZO ROBÓTICO DE 6 GRADOS DE LIBERTAD CON FUNCIONALIDAD DE MECANIZADO Y PALETIZADO

AUTORES:

EDISON VELASCO

JAVIER MAMARANDI

TUTOR: ING. DAVID RIVAS

COTUTOR: ING. MARCELO ÁLVAREZ

INTRODUCCIÓN

- La necesidad de obtener mayor ventajas y viabilidad en el uso de sistemas robóticos empuja a encontrar maneras de manipular estos sistemas a través del uso de software libre, ya que con el avance de la investigación en lenguajes de programación, hacen posibles el desarrollo de nuevas herramientas, extensiones, librerías, etc. que facilitan un lazo de comunicación entre un robot y una PC por medio de un algoritmo codificado en software libre.

PROBLEMA:

- Todos los sistemas robóticos existentes, tanto en la industria, en la medicina y otros campos en donde es necesaria la interacción de un robot, presentan un problema debido al costo de los software de control, que por ser propietarios requieren de grandes inversiones.

INVESTIGACIÓN

- Debido al problema mencionado anteriormente el presente proyecto se sumerge en la investigación de la comunicación y programación de sistemas robóticos a través de software libre el cual resolvería el inconveniente financiero, además de que abre las puertas a futuras investigaciones en estos campos.

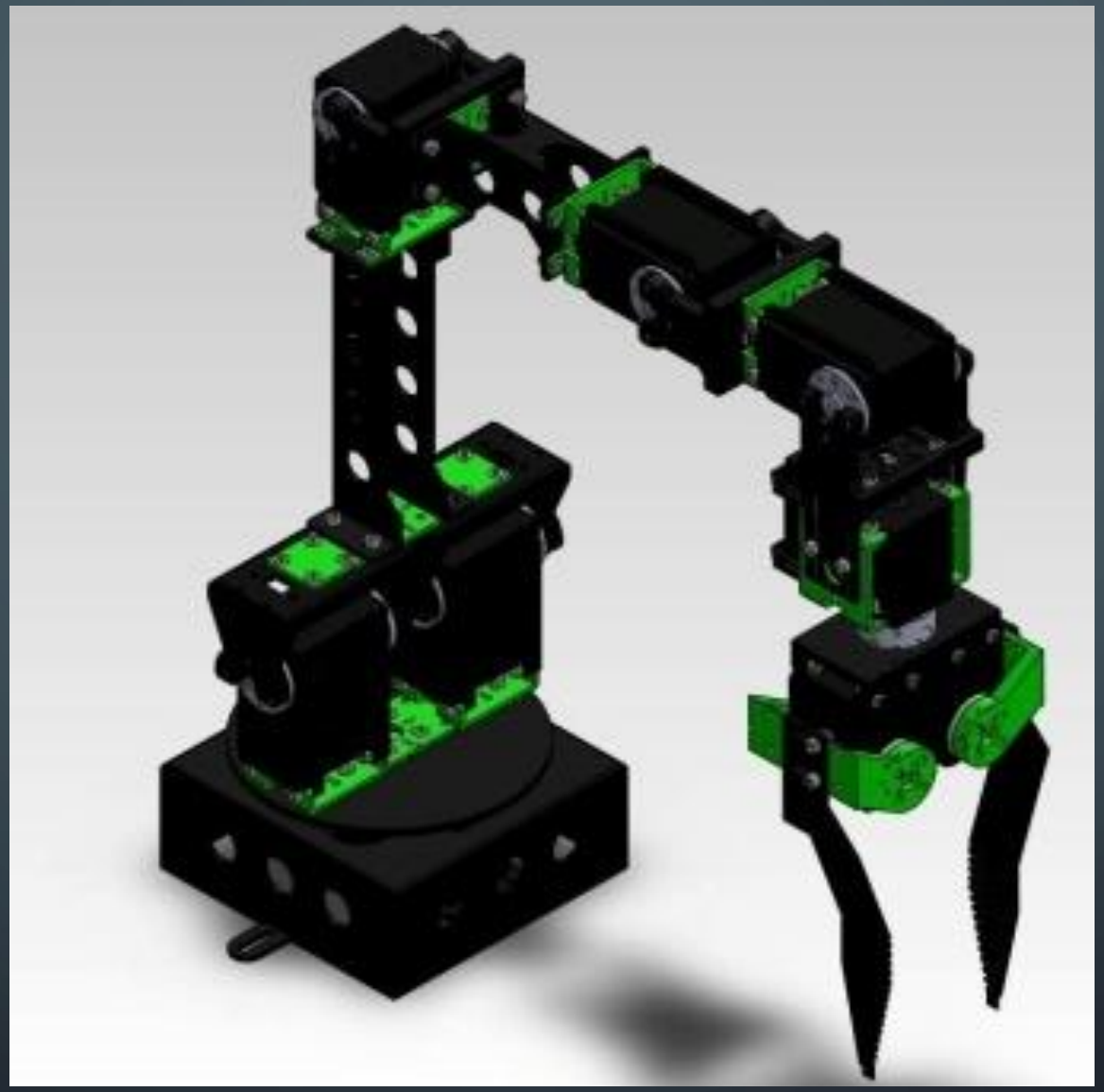
OBJETIVO GENERAL

- Desarrollar el software en Python para el control de un brazo robótico de 6 grados de libertad con funcionalidad de mecanizado y paletizado para el laboratorio de electrónica.

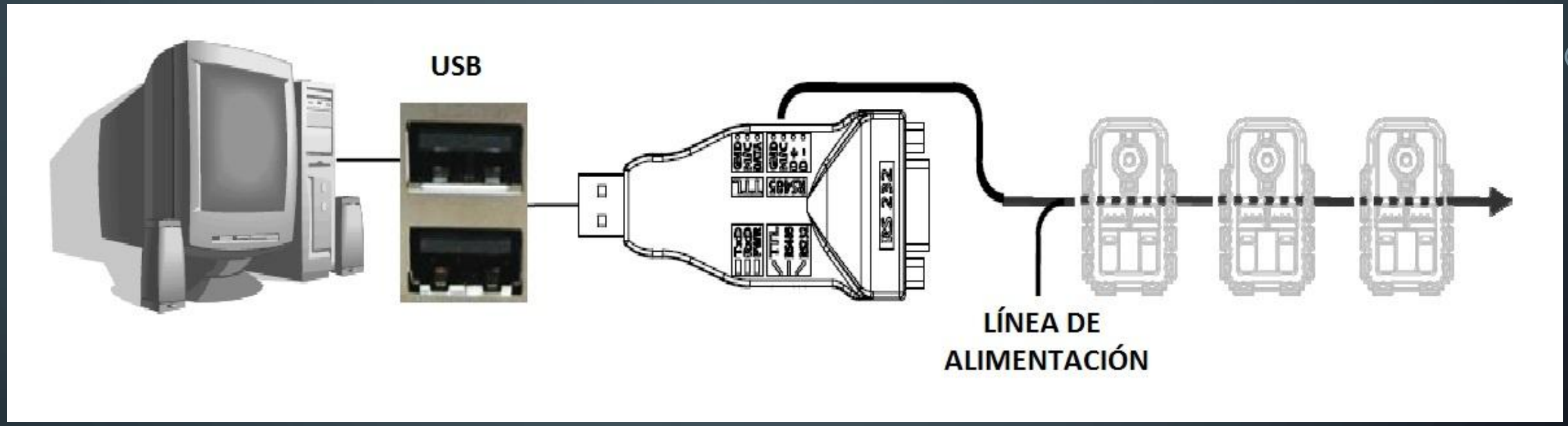
OBJETIVOS ESPECÍFICOS

- Investigar las características técnicas de los servomotores que usa el brazo robótico.
- Conocer el uso del Software Python para el desarrollo del proyecto
- Ensamblar un brazo robótico utilizando servomotores Dynamixel y una estructura diseñada e implementada en la Universidad de las Fuerzas Armadas ESPE Latacunga.

- Realizar pruebas con software probado para certificar el correcto funcionamiento del robot en la parte mecánico-electrónica.
- Probar la red de servomotores.
- Elaborar librerías en lenguaje Python para el uso del brazo robótico
- Diseñar la lógica de control con la cual se pueda tener movimientos deseados del brazo robótico que coincidan con la información contenida por la interfaz gráfica con el fin de ejecutar un mecanizado y paletizado.

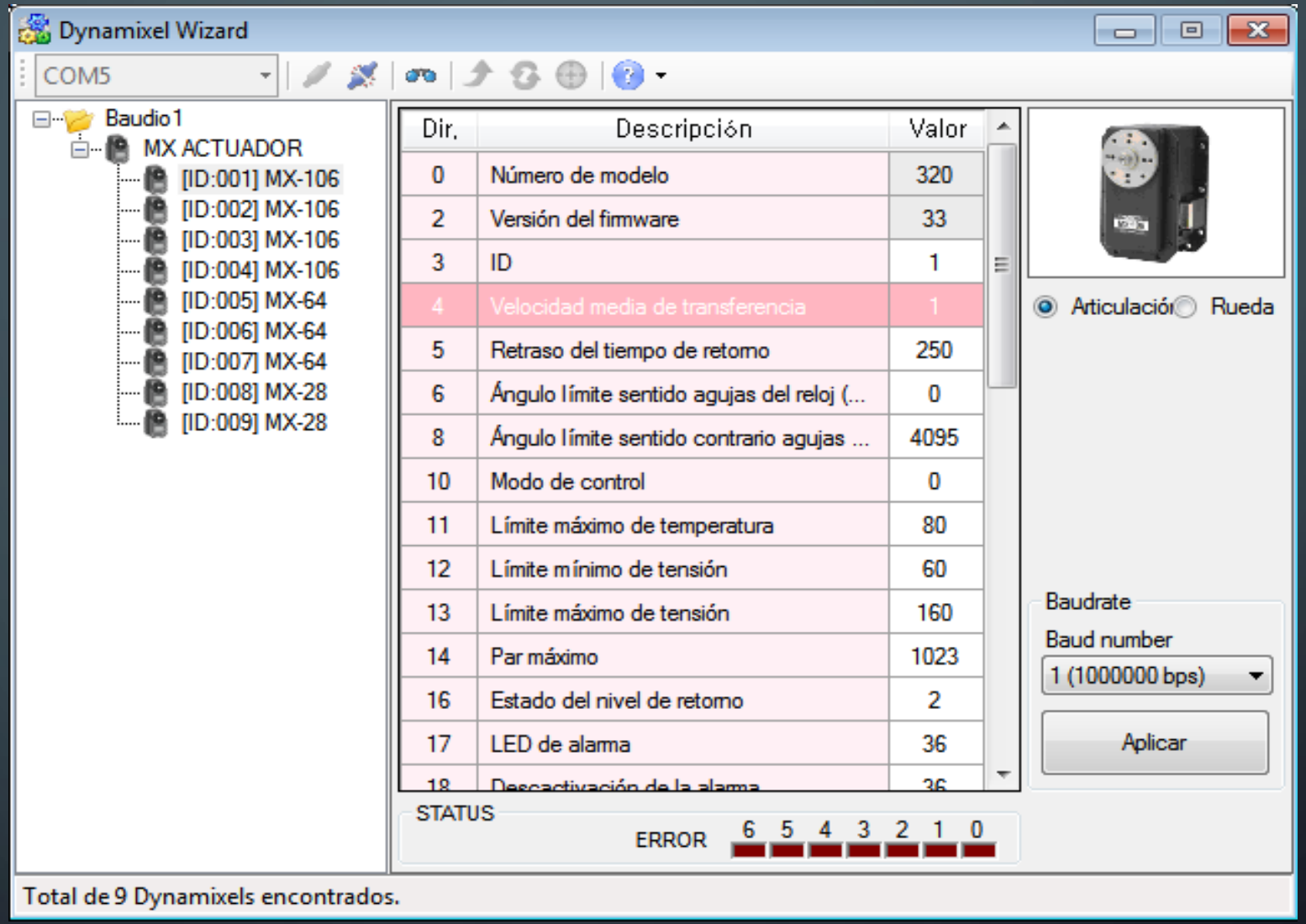


BRAZO ROBÓTICO ENSAMBLADO



COMUNICACIÓN ORDENADOR – INTERFAZ – BRAZO ROBÓTICO

SERVOMOTORES DYNAMIXEL



LECTURA Y COMUNICACIÓN DE SERVOMOTORES

PYTHON

- Lenguaje de alto nivel
- Propósito general
- Multiplataforma (Open Source)
- Sintaxis clara (identada)
- Multiparadigma
- Incrustable

INICIO

COMUNICACIÓN DEL BRAZO CON EL SOFTWARE

LECTURA DE DATOS

PARAMETRIZACIÓN DE DATOS

File Edit Format Run Options Windows Help

```
import time
from math import *
from ctypes import *

libc = windll.dynamixel # doctest: +WINDOWS

art0_cm=11.5
art1_cm=9.5
art2_cm=9.0
art3_cm=9.0
munieca_cm=10.5
pinza_cm=15.5
art4_cm=munieca_cm+pinza_cm

def iniciar():
    #print "Iniciando comunicacion..."
    n = libc.dxl_initialize();
    if n == 0:
        #print "Error al abrir el USB2Dynamixel!"
        exit();
        #print"USB2Dynamixel abierto satisfactoriamente!"

def home(evt):
    ejecutar_mov([[90, 225, 100, 175, 100, 180, 120]],10,0,0);

def led(ID,estado):
    if estado==0:
        libc.dxl_write_byte(ID,25,0);
    if estado==1:
        libc.dxl_write_byte(ID,25,1);

def carga_direccion(ID):#direccion=0 antihorario ;direccion=1 horario ;
    carga_dato=libc.dxl_read_word(ID,40)
    if carga_dato>1023:
        direccion=1
        porcentaje=(carga_dato-1024)*100/1024
    else:
        direccion=0
        porcentaje=(carga_dato)*100/1024
    return(porcentaje,direccion)
```

Ln: 4 Col: 0

INICIAR/FINALIZAR

INICIA Y FINALIZA LA COMUNICACIÓN ENTRE LA PC Y EL BRAZO ROBÓTICO

HOME

LLEVA AL BRAZO A UNA POCISIÓN PREDETERMINADA

```
def iniciar():
    #print "Iniciando comunicacion..."
    n = libc.dxl_initialize();
    if n == 0:
        #print "Error al abrir el USB2Dynamixel!"
        exit();
        #print"USB2Dynamixel abierto satisfactoriamente!"

def home(evt):
    #print "Posicion Inicial"
    ejecutar_mov([[90, 225, 100, 175, 100, 180, 120]],10,0,0);
```

BASE, ART1, ART2, ART3, ART4, MUÑECA, PINZA

PARA TODAS LAS ARTICULACIONES EXCEPTO LA PINZA SE INGRESA EL VALOR DEL MOVIMIENTO EN GRADOS, PARA LA PINZA EL RANGO ES ENTRE 0 Y 100

TEMPERATURA

LEE EL VALOR DE TEMPERATURA EN EL SERVOMOTOR SELECCIONADO

```
def art_4(grados_art4):
    if(grados_art4>290):
        grados_art4=290
    if(grados_art4<74):
        grados_art4=74
    posicion_art4= int(round(grados_art4/ 0.087890625))
    libc.dxl_write_word(6,30,posicion_art4)
    #print"Servos_art_4: "+str(grados_art4)

def munieca(grados_munieca):
    if(grados_munieca>360):
        grados_munieca=360
    if(grados_munieca<0):
        grados_munieca=0
    posicion_munieca= int(round(grados_munieca/ 0.087890625))
    libc.dxl_write_word(7,30,posicion_munieca)
    #print"Servos_munieca: "+str(grados_munieca)

def pinza(cierre):
    if(cierre>100):
        cierre=100
    if(cierre<0):
        cierre=0
    pinza_servo_1=int(round(5.5*cierre)+660)
    pinza_servo_2=int(round(-5.45*cierre)+1400)
    libc.dxl_write_word(9,30,pinza_servo_2)
    libc.dxl_write_word(8,30,pinza_servo_1)
    #print"Pinza cierre del "+str(cierre)+"%"

def temperatura(ID):
    return(libc.dxl_read_word(ID,43))
```

LEER SERVO

LEE EL VALOR DE POSICIÓN DEL SERVO QUE SE REQUIERA

```
def leer_servo(articulacion):#articulaciones con datos de 0 a 1023 escalados
    if articulacion=="Base":
        b=round((libc.dxl_read_word(1,36)*0.087890625),2)
    elif articulacion=="Art_1":
        b=round((libc.dxl_read_word(2,36)*0.087890625),2)
    elif articulacion=="Art_2":
        b=round((libc.dxl_read_word(4,36)*0.087890625),2)
    elif articulacion=="Art_3":
        b=round((libc.dxl_read_word(5,36)*0.087890625),2)
    elif articulacion=="Art_4":
        b=round((libc.dxl_read_word(6,36)*0.087890625),2)
    elif articulacion=="Munieca":
        b=round((libc.dxl_read_word(7,36)*0.087890625),2)
    elif articulacion=="Pinza":
        pinza_servo1=round((0.1818*libc.dxl_read_word(8,36)-120),1)
        pinza_servo2=round((-0.19417*libc.dxl_read_word(9,36)+271.844),1)
        b=round((pinza_servo2+pinza_servo1)/2,2)
        b=pinza_servo2
        if(b>100):
            b=100
        elif(b<0):
            b=0
    return(b)
```

PAR ON/OFF

ACTIVA/DESACTIVA EL TORQUE DE LOS SERVOMOTORES DE LA ARTICULACIÓN INDICADA

```
def parOn(articulacion):
    if(articulacion=="Base"):
        libc.dxl_write_word(1,24,1)
    elif(articulacion=="Art_1"):
        libc.dxl_write_word(2,24,1)
        libc.dxl_write_word(3,24,1)
    elif(articulacion=="Art_2"):
        libc.dxl_write_word(4,24,1)
    elif(articulacion=="Art_3"):
        libc.dxl_write_word(5,24,1)
    elif(articulacion=="Art_4"):
        libc.dxl_write_word(6,24,1)
    elif(articulacion=="Munieca"):
        libc.dxl_write_word(7,24,1)
    elif(articulacion=="Pinza"):
        libc.dxl_write_word(8,24,1)
        libc.dxl_write_word(9,24,1)
    #print'Torque '+str(articulacion)+' Activado'

def parOff(articulacion):
    if(articulacion=="Base"):
        libc.dxl_write_word(1,24,0)
    elif(articulacion=="Art_1"):
        libc.dxl_write_word(2,24,0)
        libc.dxl_write_word(3,24,0)
    elif(articulacion=="Art_2"):
        libc.dxl_write_word(4,24,0)
    elif(articulacion=="Art_3"):
        libc.dxl_write_word(5,24,0)
    elif(articulacion=="Art_4"):
        libc.dxl_write_word(6,24,0)
    elif(articulacion=="Munieca"):
        libc.dxl_write_word(7,24,0)
    elif(articulacion=="Pinza"):
        libc.dxl_write_word(8,24,0)
        libc.dxl_write_word(9,24,0)
    #print'Torque '+str(articulacion)+' Desactivado'
```

VELOCIDAD

AJUSTA LOS SERVOMOTORES DE LA ARTICULACIÓN INDICADA A UNA VELOCIDAD INGRESADA EN RPM

```
def velocidad(articulacion,rpm): #INGRESO EN RPM, de 0.1 hasta  55-67-78 RPM
    if(rpm>0):
        #print"ERROR LOGICO,INGRESO EN RPM, de 0.1 hasta  55-67-78 RPM "
        VelDyn=int(round(rpm*(10.23))
        if(articulacion=="Pinza"):
            libc.dxl_write_word(9,32,VelDyn)
            libc.dxl_write_word(8,32,VelDyn)
        elif(articulacion=="Munieca"):
            libc.dxl_write_word(7,32,VelDyn)
        elif(articulacion=="Art_4"):
            libc.dxl_write_word(6,32,VelDyn)
        elif(articulacion=="Art_3"):
            libc.dxl_write_word(5,32,VelDyn)
        elif(articulacion=="Art_2"):
            libc.dxl_write_word(4,32,VelDyn)
        elif(articulacion=="Art_1"):
            libc.dxl_write_word(3,32,VelDyn)
            libc.dxl_write_word(2,32,VelDyn)
        elif(articulacion=="Base"):
            libc.dxl_write_word(1,32,VelDyn)
        #print'Velocidad de '+str(articulacion)+' a '+str(rpm)+' RPM'
```

GRABAR MOVIMIENTOS

GUARDA UN NUMERO DE POSICIONES DEL BRAZO Y LAS EJECUTA EN SECUENCIA

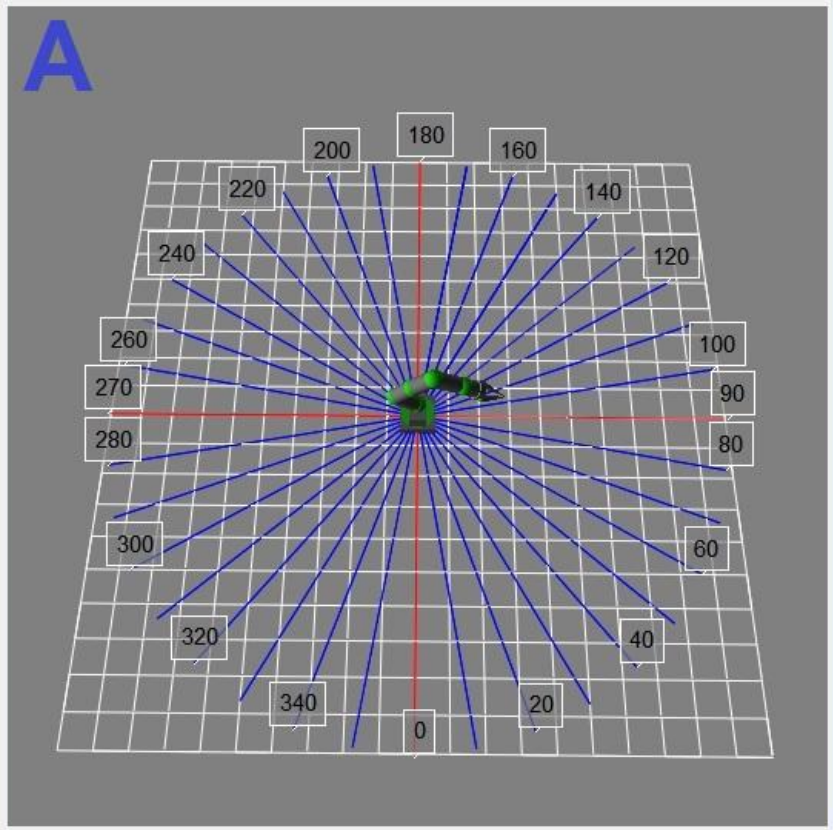
```
def grabar_mov(movimientos):
    leds(1,1,1,1,1,1,1,1,1)
    a=[]
    for i in range(movimientos):
        a.append([0]*5)
    #print (a)
    for i in range(movimientos):
        off()
        raw_input('Ingrese movimiento:'+str(i+1))
        on()
        for j in range(5):
            leds(0,0,0,0,0,0,0,0,0)
            delay(100)
            leds(1,1,1,1,1,1,1,1,1)
            delay(100)
        m=[leer_servo("Base"),leer_servo("Art_1"),}
                    leer_servo("Art_2"),leer_servo("Art_3"),
                    leer_servo("Art_4"),leer_servo("Munieca"),leer_servo("Pinza")]
        a[i]=m;
    raw_input("Matriz"+str(a))
    leds(0,0,0,0,0,0,0,0,0)
    return(a)
```

INTERFAZ GRÁFICA

Interfaz Grafica

File Opciones

A



Simulacion Coordenadas Grados

Caracteristicas de los Servomotores

	ID	Temperatura(ºC)	Voltaje(V)	Carga(%)
Servo	1	35	12.0	0
Servo	2	40	11.7	3
Servo	3	38	11.6	2
Servo	4	54	11.2	7
Servo	5	62	11.1	9
Servo	6	37	11.0	4
Servo	7	35	10.7	0
Servo	8	36	10.5	1
Servo	9	35	10.6	1

P 32 I 1 D 0

Servo # 10 Sintonizar

E

Posiciones

B

Base 90

Articulacion 1 224,65

Articulacion 2 99,14

Articulacion 3 173,94

Articulacion 4 99,49

Muñeca 270

Pinza 100

Servos Lectura Escritura

Actualizar

Torque

C

Todos Base Articulacion 1 Articulacion 2 Articulacion 3 Articulacion 4 Muñeca Pinza

Velocidades

D

Base 5 %

Articulacion 1 5 %

Articulacion 2 5 %

Articulacion 3 5 %

Articulacion 4 5 %

Muñeca 5 %

Pinza 5 %

Todas 5 %

Funciones

F

COM Home Cinemática

Paro Img Líneas Img Puntos

Coordenadas

H

X Actual 0.0 Y Actual 24.54 Z Actual 16.06

V V

X Next 0.0 Y Next 24.54 Z Next 16.14

X+ Y+ Z+

X- Y- Z-

Iteraciones 2.5

Posiciones Guardadas

G

1.-[90.26, 223.86, 98.17, 172.18, 99.05, 90.26, 100.0]
2.-[72.25, 223.42, 97.12, 171.47, 98.61, 89.82, 54.0]
3.-[75.76, 217.53, 78.93, 167.96, 120.32, 91.32, 68.4]
4.-[75.85, 215.6, 78.75, 167.61, 120.85, 90.97, 86.0]
5.-[75.85, 234.4, 94.66, 168.57, 110.57, 90.62, 86.0]
6.-[125.95, 206.46, 94.31, 168.57, 134.82, 90.35, 86.0]
7.-[125.95, 205.4, 93.34, 167.7, 134.38, 90.26, 67.0]
8.-[112.94, 221.84, 93.6, 167.87, 97.21, 90.44, 65.8]
9.-[111.8, 213.66, 94.39, 168.57, 92.72, 90.53, 60.8]
10.-[111.88, 213.13, 92.81, 167.08, 92.37, 90.35, 86.0]
11.-[110.48, 213.57, 91.23, 165.5, 139.13, 90.62, 86.0]
12.-[127.79, 211.46, 80.68, 166.64, 158.29, 90.26, 86.0]

T1 100 T2 100

Abrir Guardar >>> <<<

Insertar Secuencia 1

Grabar Ejecutar 1 Ejecutar Secuencia 1

Borrar Todo Borrar 1 Reemplazar Secuencia 1

Variables de imagen

I

Dimensiones X 10 Y 10

Desfase X -10 Y 15

Altura 0

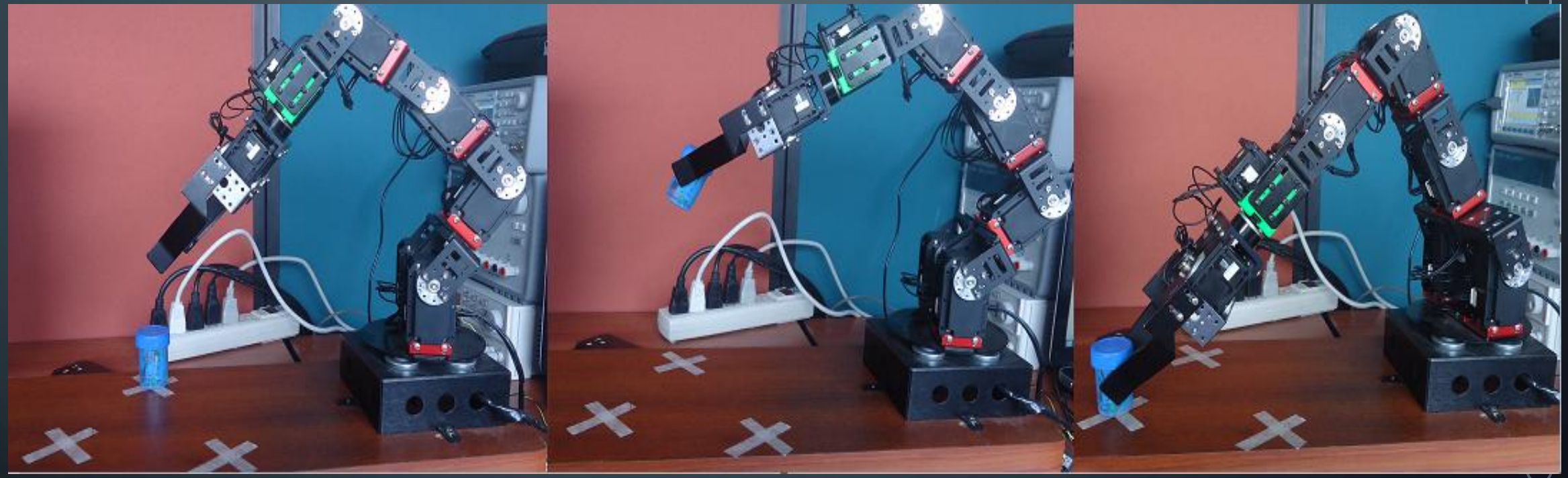
ANÁLISIS DE LA INTERFAZ GRÁFICA

- Facilidad de manejo
- Trabajo en tiempo real
- Interfaz usuario – máquina interactiva e intuitiva
- Software con arquitectura abierta

PALETIZADO

- Consiste en agrupar sobre una superficie una cierta cantidad de productos, con la finalidad de conformar una unidad de manejo que pueda ser transportada y almacenada con el mínimo esfuerzo y en una sola operación.

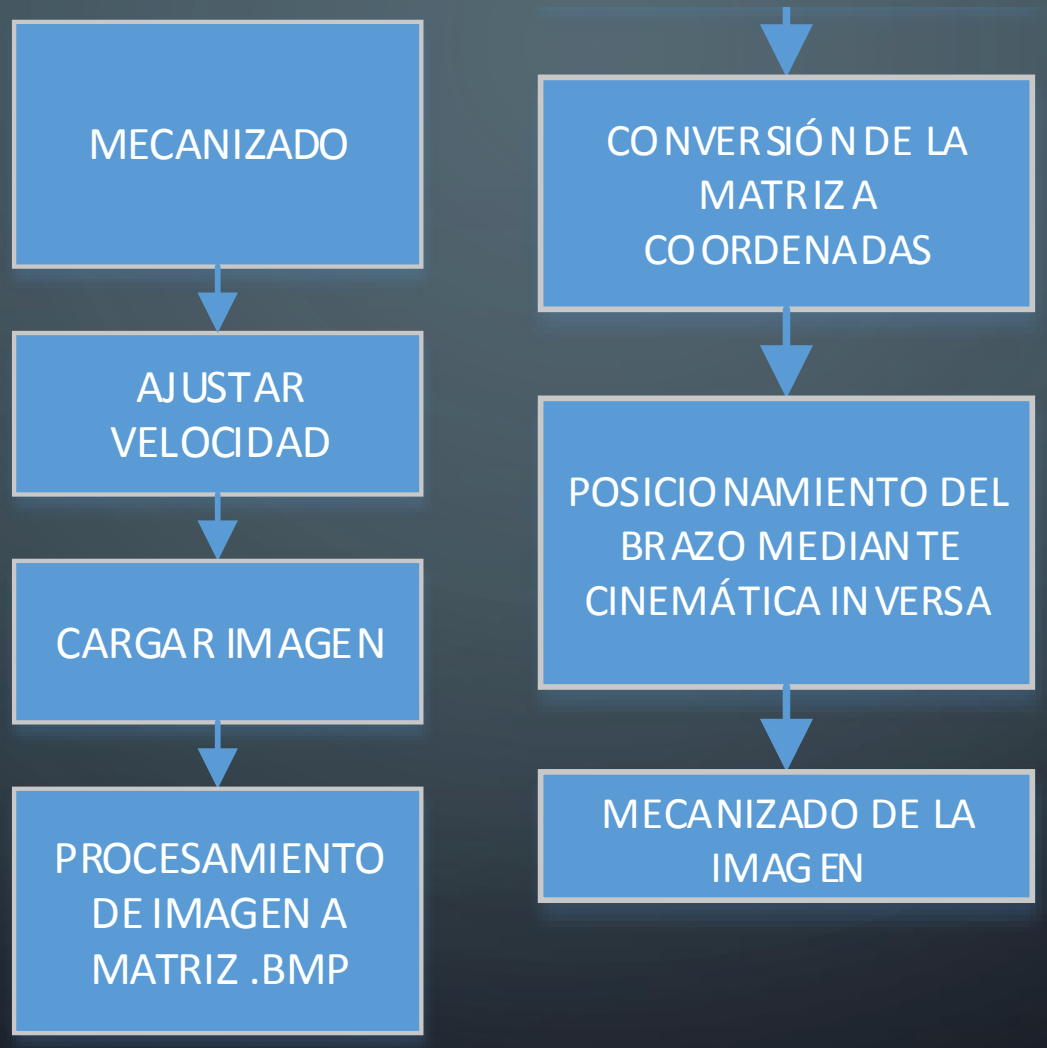

PALETIZADO
INGRESO DE COORDENADAS
POSICIONAMIENTO DEL BRAZO
TOMA DEL OBJETO
INGRESO DE LAS COORDENADAS DE LLEGADA
POSICIONAMIENTO DEL BRAZO Y LIBERACIÓN DE OBJETO



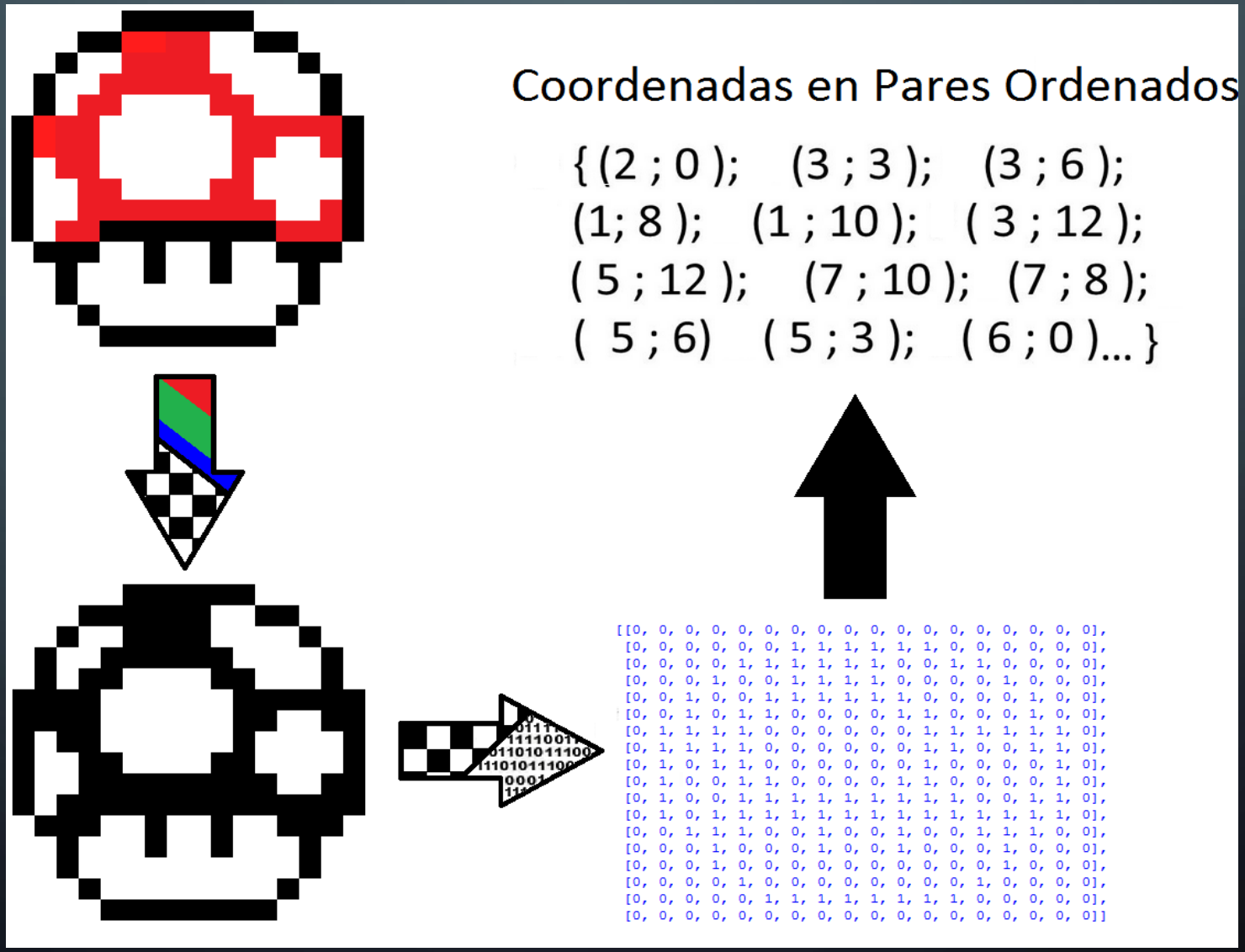
FUNCIÓN DE PALETIZADO

MECANIZADO

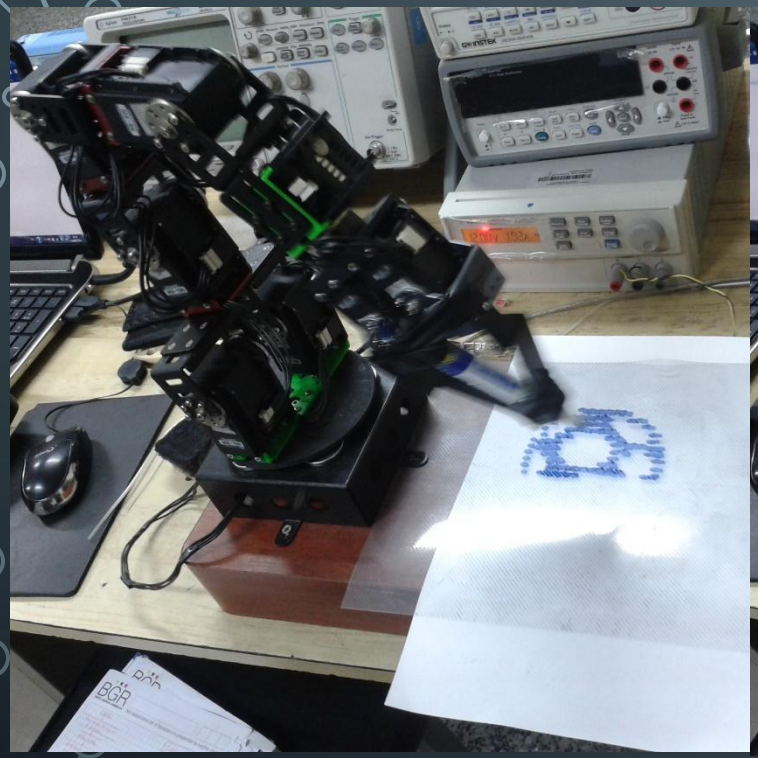
- Es un proceso de fabricación que comprende un conjunto de operaciones de conformación de piezas mediante la eliminación de material, ya sea por arranque de viruta o por abrasión.

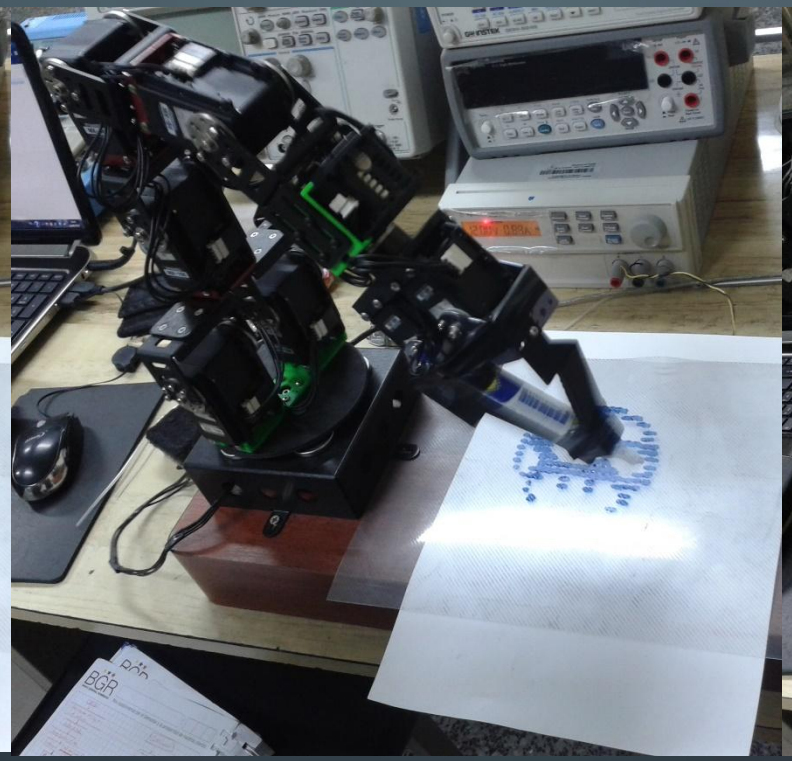

MECANIZADO
AJUSTAR VELOCIDAD
CARGAR IMAGEN
PROCESAMIENTO DE IMAGEN A MATRIZ .BMP
CONVERSIÓN DE LA MATRIZ A COORDENADAS
POSICIONAMIENTO DEL BRAZO MEDIANTE CINEMÁTICA INVERSA
MECANIZADO DE LA IMAGEN

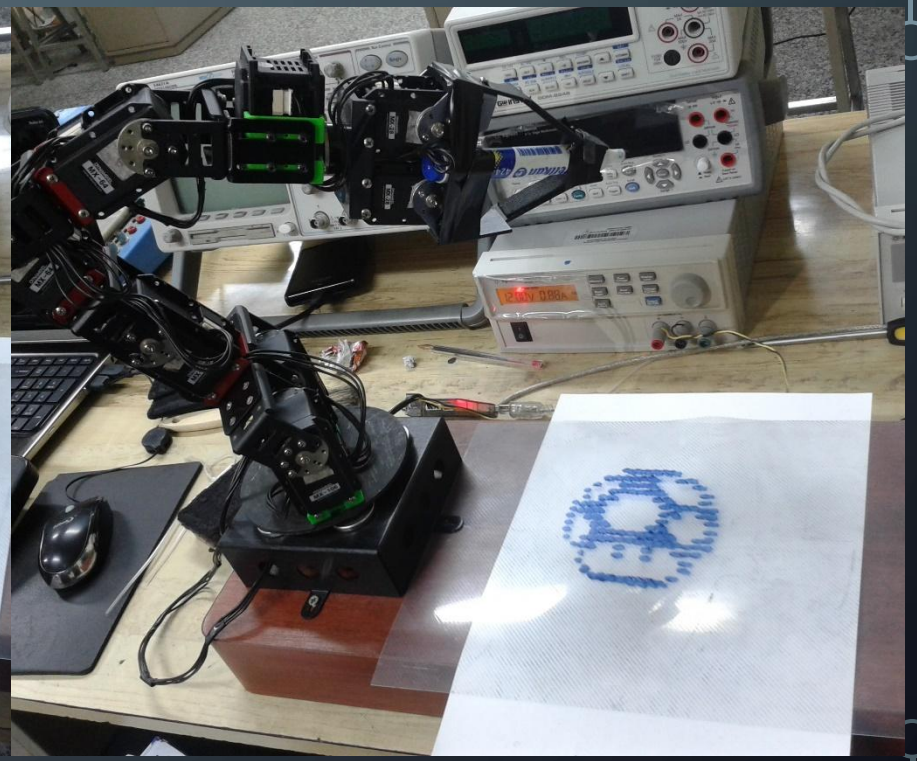
PROCESAMIENTO DE IMAGEN-MECANIZADO POR PIXELES



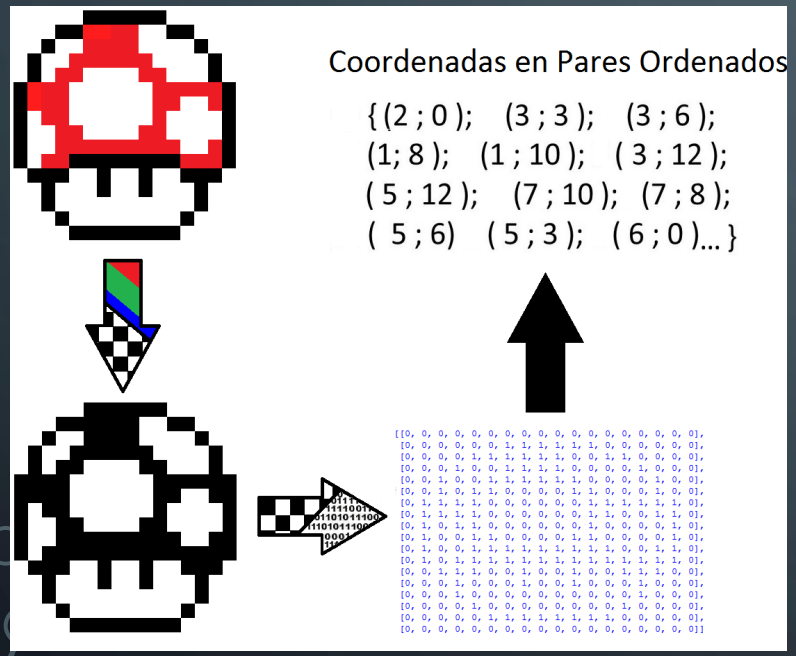
PROCESAMIENTO DE IMAGEN-MECANIZADO POR PIXELES





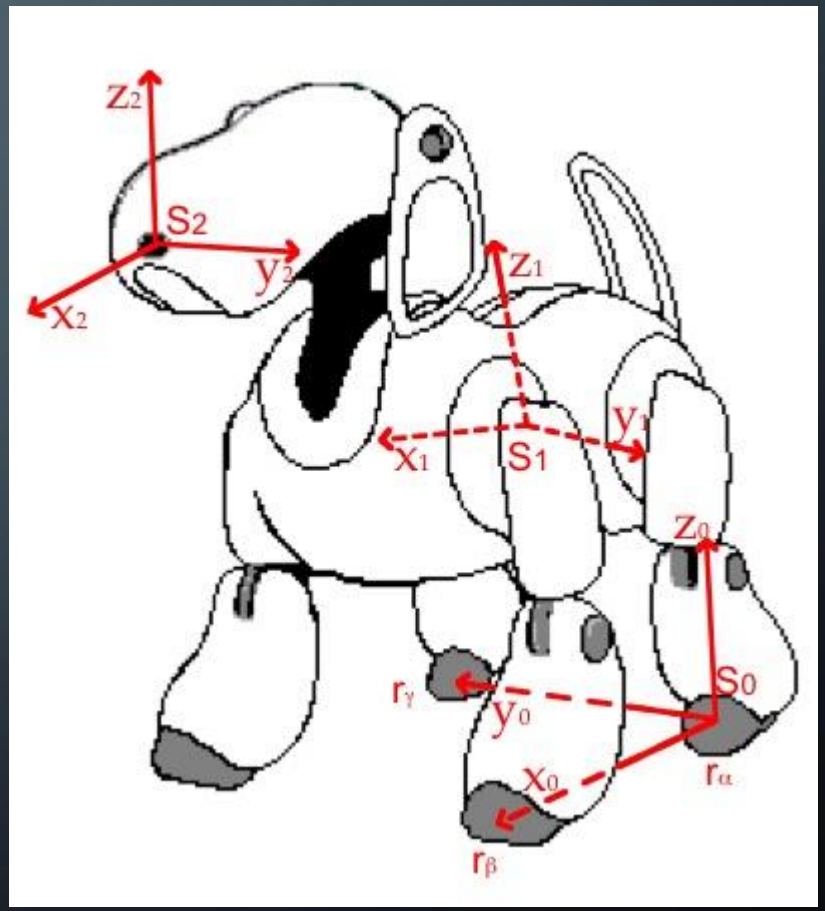


PROCESAMIENTO DE IMAGEN-MECANIZADO POR LÍNEAS

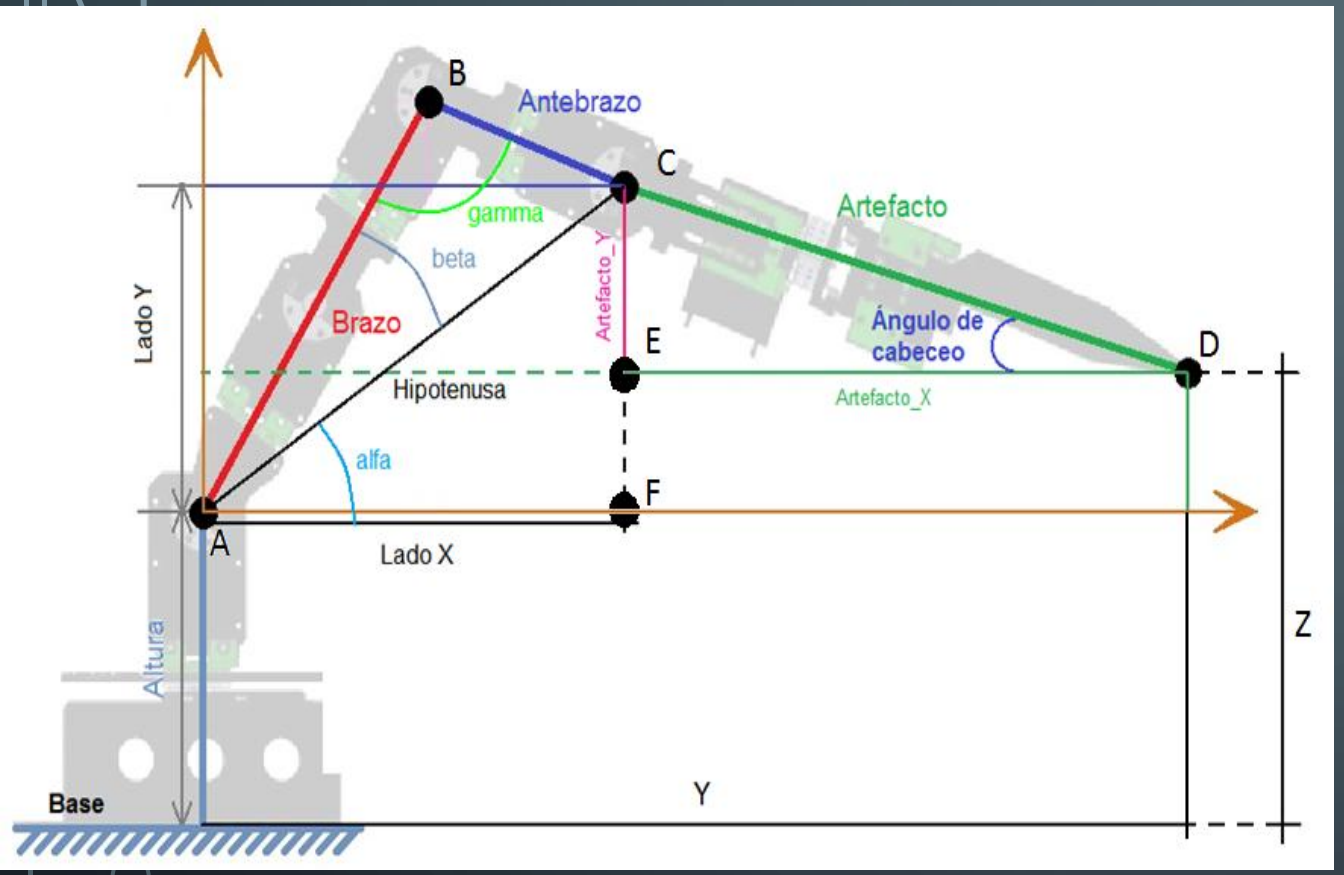


CINEMÁTICA INVERSA

- En la robótica es la técnica que permite determinar el movimiento de una cadena de articulaciones para lograr que un actuador final se ubique en una posición concreta.



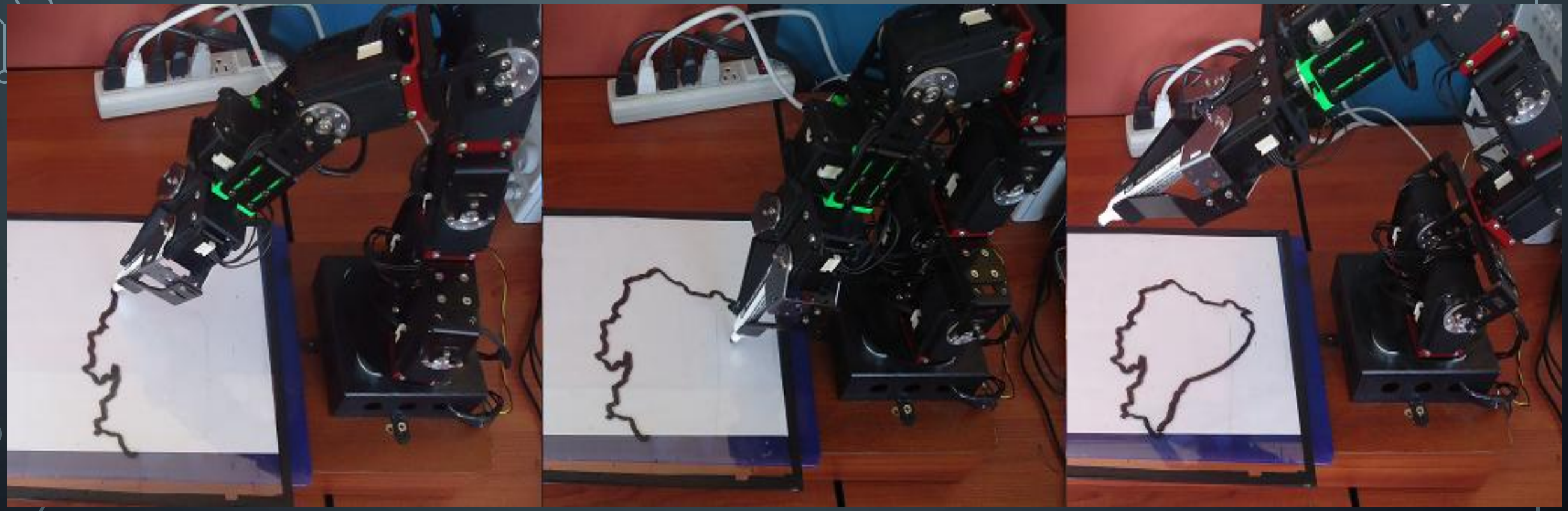
CÁLCULOS CINEMÁTICA INVERSA



( )

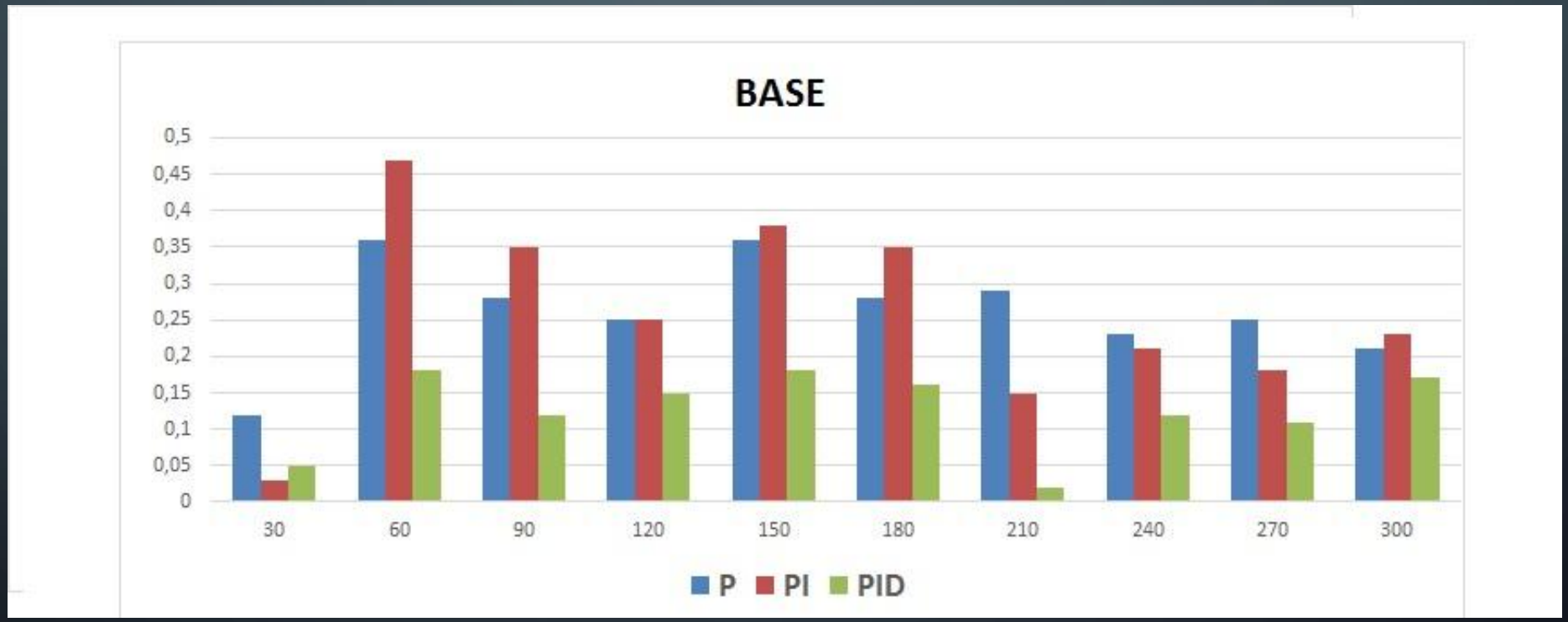
( )

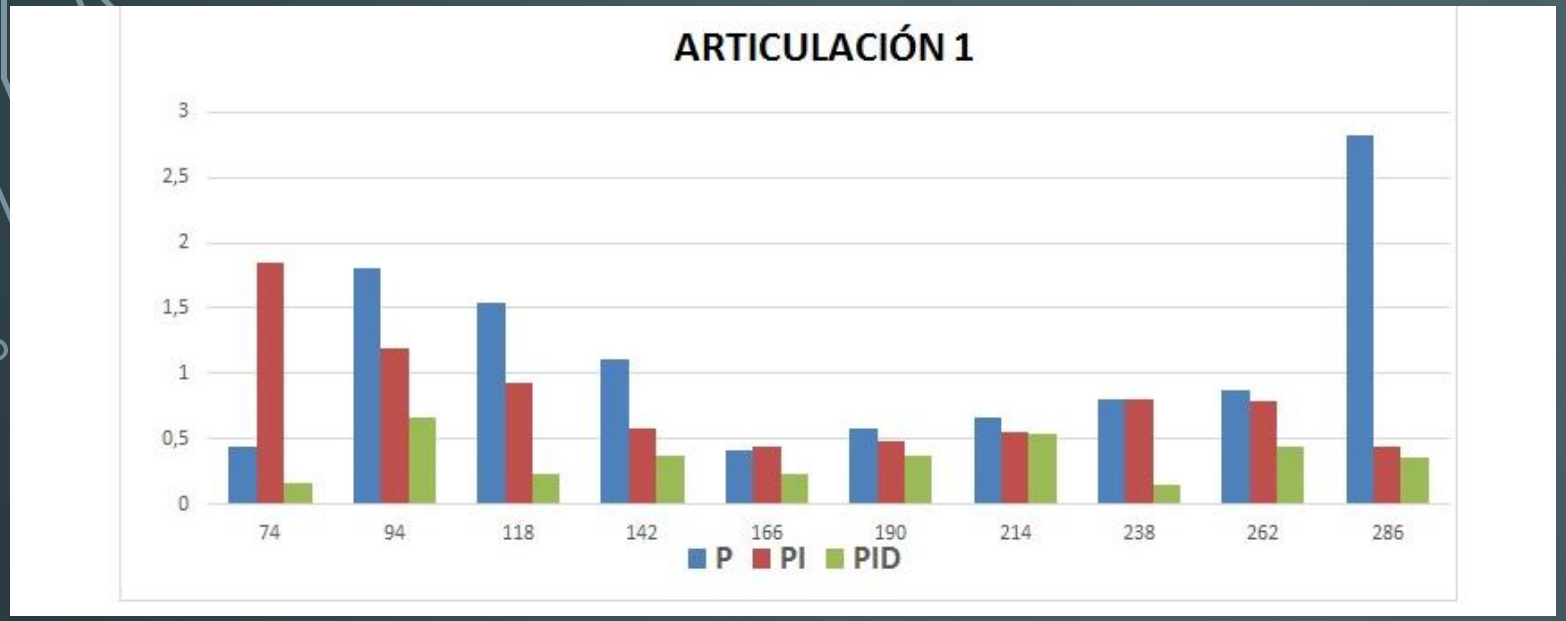
EJEMPLOS DE MECANIZADOS

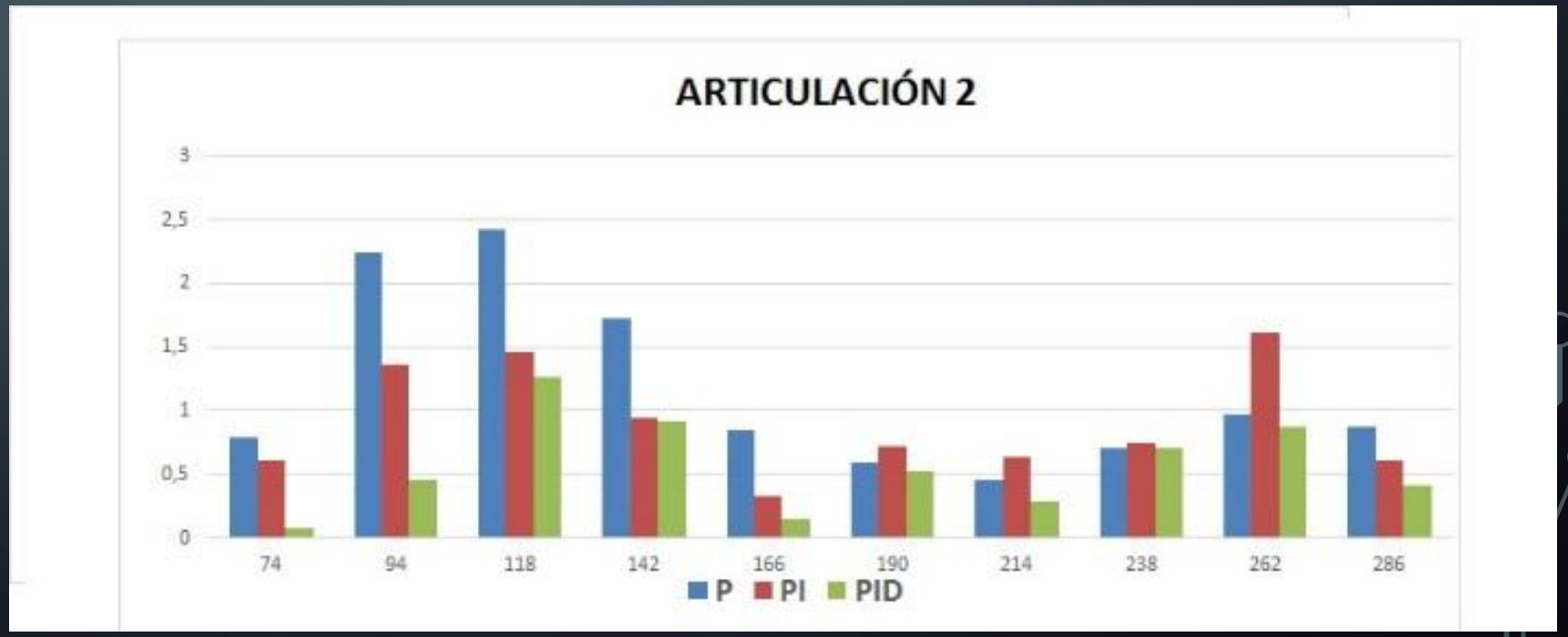


PRUEBAS DEL SISTEMA

- Gráficas del control P, PI, PID en las articulaciones del brazo robótico



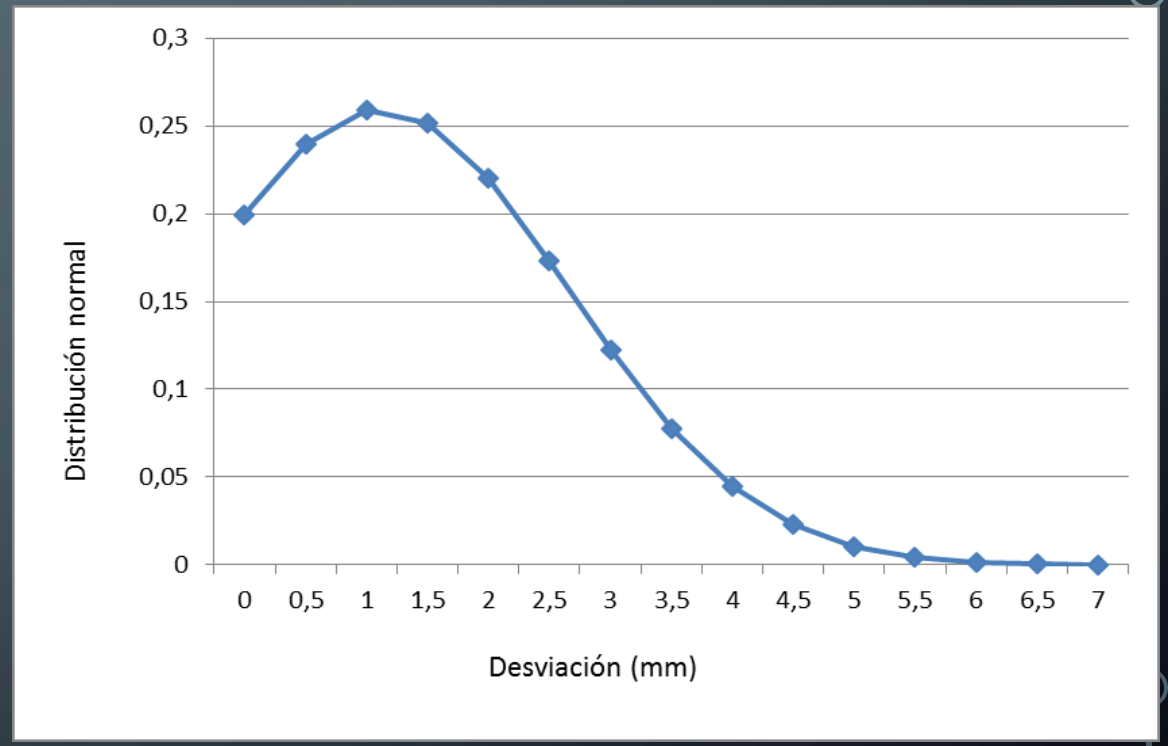

ARTICULACIÓN 1
3
2,5
2
1,5
1
0,5
0
74
94
118
142
166
190
214
238
262
286
P
PI
PID


ARTICULACIÓN 2
3
2,5
2
1,5
1
0,5
0
74
94
118
142
166
190
214
238
262
286
P
PI
PID

• Pruebas de desviación en el paletizado

Art.	Vel. (%)	Dispersión mm	Observación
Base	2	0	N
	4	0	N
	6	0	N
	8	0	N
	10	0	N
	12	1	N
	14	1,5	N
	16	1,5	N
	18	2	N
	20	2	N
	22	2	N
	24	2	N
	26	2	N
	28	2	N
	30	2	P.O.B
	32	2	P.O.B
	34	2	P.O.B
	36	2	P.O.B
	38	2	P.O.B
	40	2	P.O.B
	42	2	P.O.B
	44	2	P.O.B
	46	2	P.O.B
	48	2	P.O.B
	50	2	P.O.B
	52	2,5	P.O.B
	54	2,5	P.O.B

Art.	Vel. (%)	Dispersión mm	Observación
Articulacion 1	0,4	0	N
	0,8	0	N
	1,2	0	N
	1,6	0	N
	2	0	N
	2,4	0	N
	2,8	1	N
	3,2	2	N
	3,6	3	N
	4	3	N
	4,4	2,5	N
	4,8	2	N
	5,2	3	N
	5,6	2	N
	6	2	P.O.B
	6,4	1	P.O.B
	6,8	2,5	P.O.B
	7,2	3	P.O.B
	7,6	2	M.O.B.
	8	2,5	M.O.B.
	8,4	3	M.O.B.
	8,8	3	M.O.B.
	9,2	2,5	M.O.B.
	9,6	3	M.O.B.
	10	2,5	M.O.B.
	10,4	2	M.O.B.
	10,8	1,5	M.O.B.



- Pruebas de desviación en el mecanizado

Prueba	Medido	Ideal		
	Desfase Y	Desfase Y	Error	Promedio
1	13,0	13,2	0,2	0,1
2	13,5	13,6	0,1	
3	14,0	14,1	0,1	
4	14,6	14,7	0,1	
5	15,0	15,3	0,3	
6	15,6	15,8	0,2	
7	16,0	16,2	0,2	
8	16,5	16,5	0,0	
9	17,0	17,2	0,2	
10	17,6	17,6	0,0	
11	18,2	18,5	0,3	
12	18,5	18,6	0,1	
13	19,0	19,0	0,0	
14	19,6	19,6	0,0	
15	20,2	20,3	0,1	
16	20,6	20,7	0,1	
17	21,1	21,1	0,0	
18	21,7	21,7	0,0	
19	22,3	22,4	0,1	
20	22,6	22,7	0,1	

Prueba	Ideal	Medido		
	Desfase X	Desfase X	Error	Promedio
1	-13,0	-13,0	0,0	0,138
2	-11,6	-11,7	0,1	
3	-10,3	-10,1	0,2	
4	-8,9	-9,2	0,3	
5	-7,5	-7,3	0,2	
6	-6,2	-6,1	0,1	
7	-4,8	-5,0	0,2	
8	-3,4	-3,5	0,1	
9	-2,1	-2,0	0,1	
10	-0,7	-0,7	0,0	
11	0,7	0,9	0,2	
12	2,1	1,7	0,4	
13	3,4	3,6	0,2	
14	4,8	4,8	0,0	
15	6,2	6,3	0,1	
16	7,5	7,6	0,1	
17	8,9	9,0	0,1	
18	10,3	10,5	0,2	
19	11,6	11,7	0,1	
20	13,0	13,2	0,2	

CONCLUSIONES:

- El brazo robótico ensamblado responde correctamente al algoritmo de control en las funcionalidades de mecanizado y paletizado.
- Al tratarse de un sistema de control desarrollado en software libre permite que cualquier persona con conocimientos de programación pueda realizar cambios en las librerías para mejorarlas, estudiarlas o basarse en ellas para futuros proyectos.

- El protocolo de comunicación RS-485 permite comunicar los nueve dispositivos (servomotores) que conforman el brazo robótico en serie asegurando los valores de voltaje necesarios para que funcione normalmente el sistema robótico cuando estos trabajan al máximo.

- Con la programación realizada tanto en funciones como en interfaz gráfica, se requirió del estudio de otros campos como la cinemática inversa, matemática de vectores, linealización de rangos, procesamiento de imágenes, comprensión de la mecánica en robots, etc. Y así poder relacionar todo en un solo sistema.
- Se realizaron funciones de movimiento, ajuste de torque, velocidad, lectura y guardado de los mismos, entre otras; todas estas dependiendo de los parámetros o datos que se podían leer del servomotor.

- Para la función del paletizado se trabajó también en coordenadas polares, en donde al brazo robótico se le envía los datos de coordenadas en valores de radio, ángulo de giro y altura, se obtuvo los resultados esperados ya que tomaba objetos de dicha posición y los colocaba en otro lugar igualmente designado por el usuario.

- Para la función del mecanizado se tuvo que realizar el estudio e investigación de la cinemática inversa, al comprender este tema la idea fundamental era leer la imagen que se trataba de dibujar y transformarla a una matriz de coordenadas para que el brazo robótico recorra dichos puntos y trace la figura cargada al programa. Para esto lo que se realizó es procesamiento de imágenes en Python y una relación de escalado de la matriz obtenida de la imagen en unos y ceros a una matriz de coordenadas por donde el brazo robótico recorrería en el espacio.

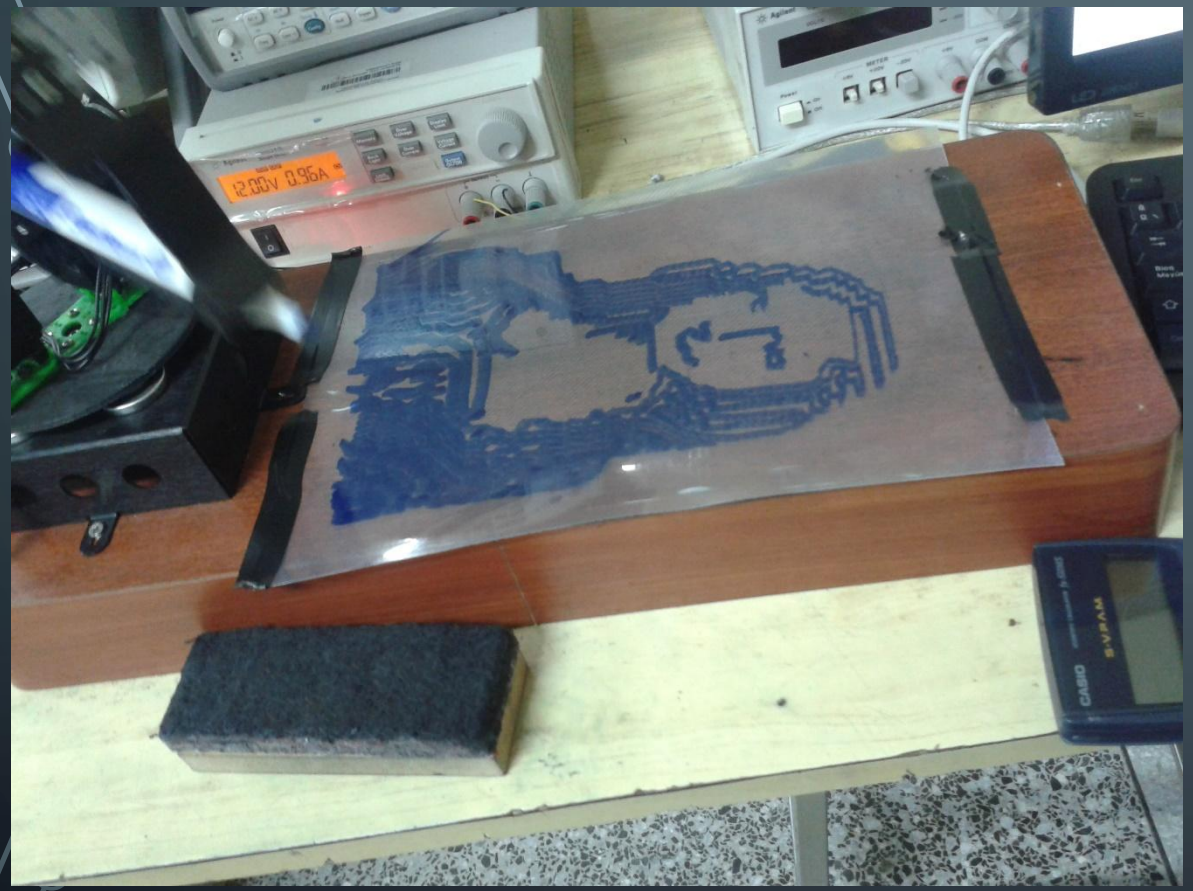
RECOMENDACIONES

- La estructura del brazo robótico no admite que este gire indefinidamente en su propio eje, debido a sus cables de comunicación. Por esto se debe establecer una zona de trabajo no mayor a los 359°, además se debe tomar en cuenta que el plano Z no se admiten valores negativos porque la base del brazo robótico se encuentra en un plano horizontal.

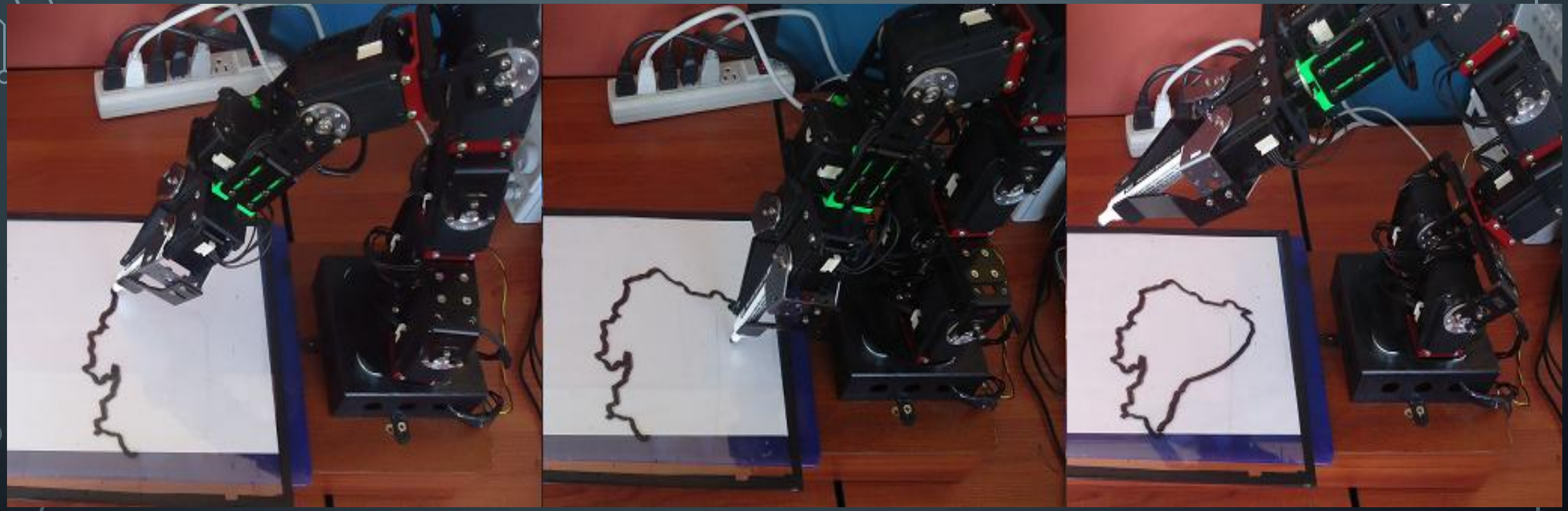
- La interfaz gráfica que se desarrolló posee un sistema de detecciones de errores, uno de los más comunes que ocurre mientras se está controlando el brazo robótico es el error de sobre carga, esto se debe a que los servomotores están sometidos a una gran fuerza, la cual el servomotor no puede soportar. En el monitoreo se puede ver el porcentaje de carga de cada servomotor, lo recomendable es que este porcentaje no exceda el 30%, debido a que puede activar la alarma de Overload, que posee cada servomotor, haciendo que este se apague.

- El programa de control desarrollado en el lenguaje de programación Python, no puede ejecutar varias tareas a la vez, por lo cual no puede monitorear una secuencia del brazo robótico en su interfaz gráfica. Para solucionarlo se debe implementar un sistema multi-hilo o multi-tarea en el programa, esto permitirá ejecutar una secuencia de instrucciones mientras se desarrollan otras.

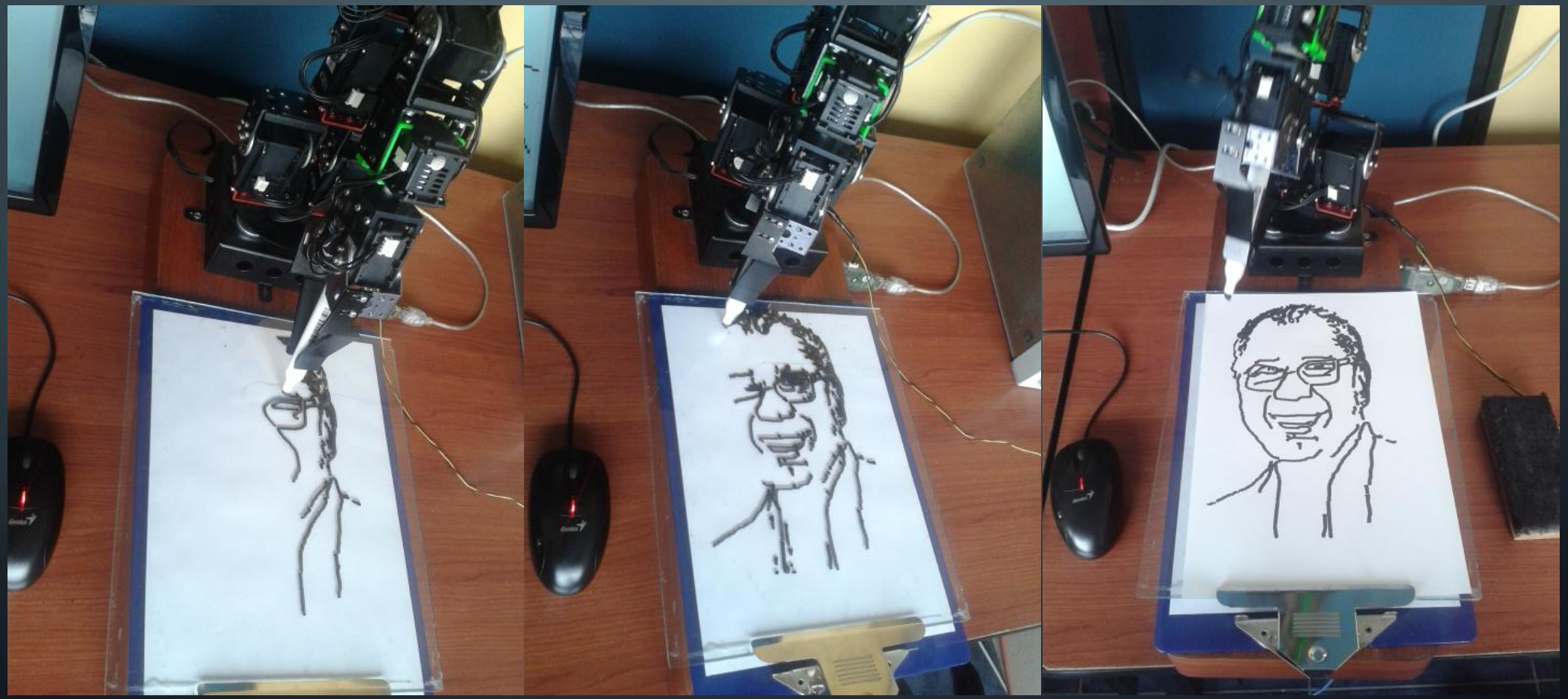
EJEMPLOS DE MECANIZADOS



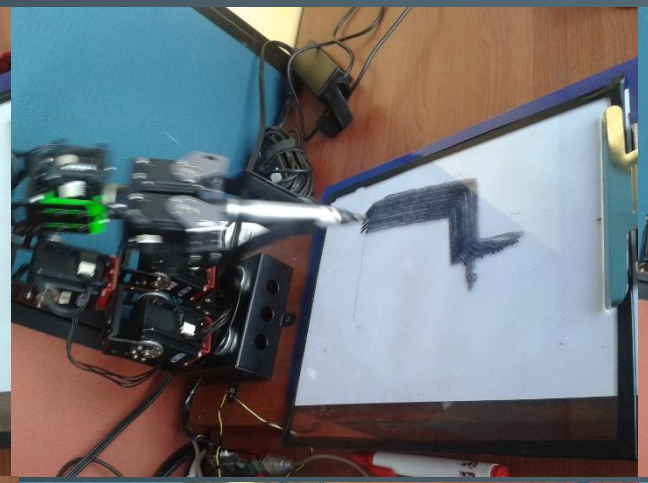
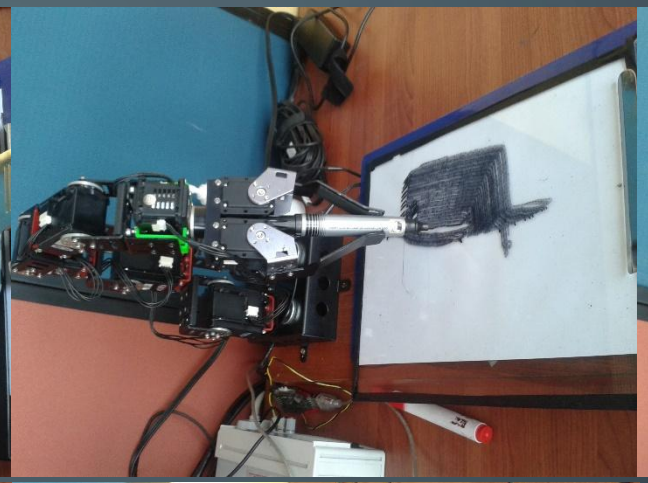
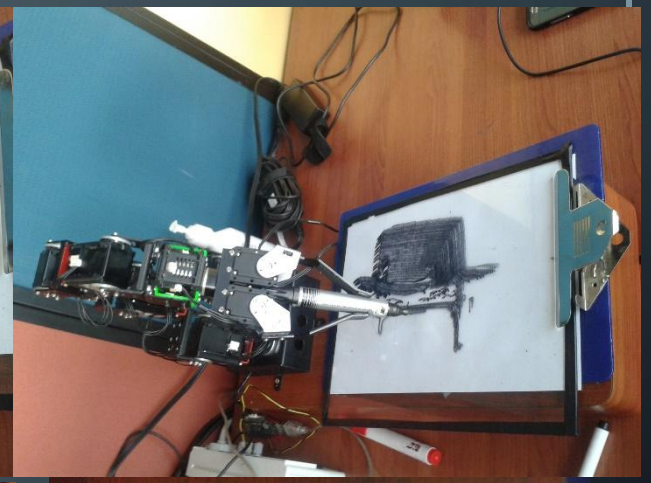
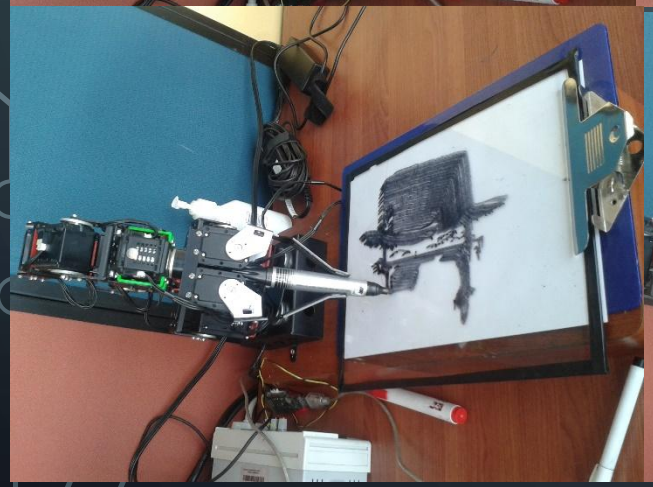
EJEMPLOS DE MECANIZADOS

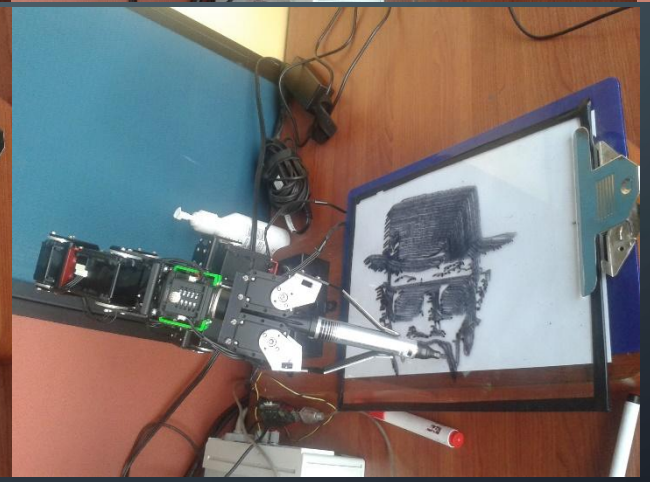
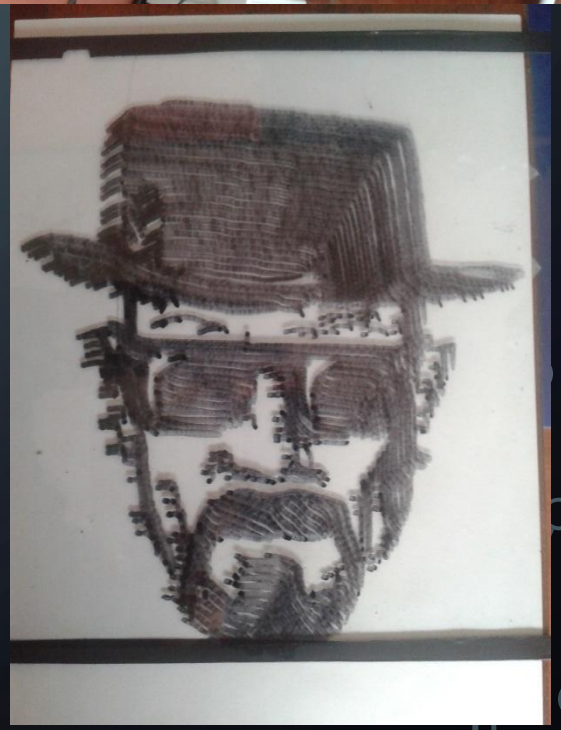


EJEMPLOS DE MECANIZADOS

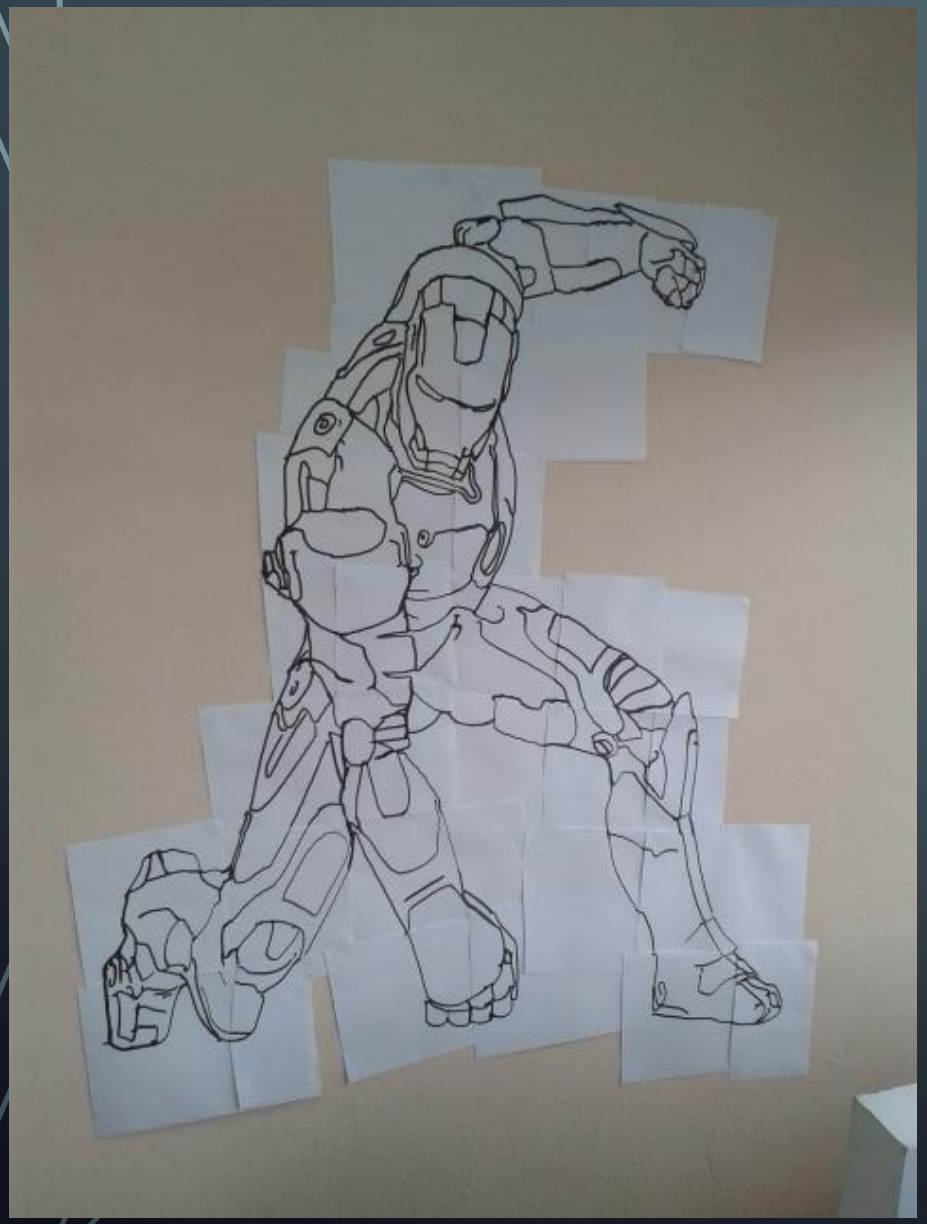


EJEMPLOS DE MECANIZADOS

EJEMPLOS DE MECANIZADOS



GRACIAS