BRAZO ROBOTICO



5-B Mecatrónica

Controladores lógicos programables

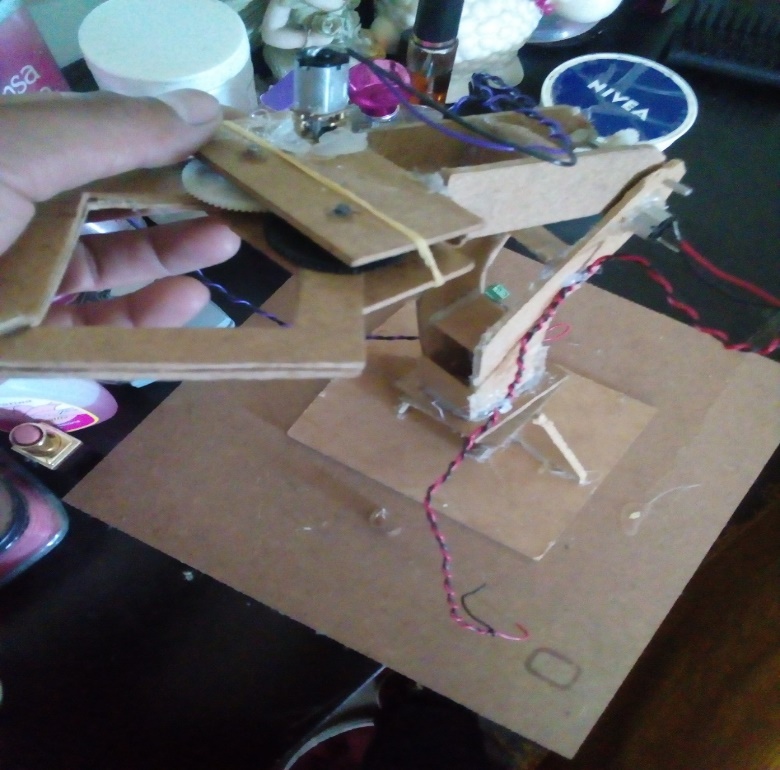
* PROYECTO FINAL
* NEGRETE HERNANDEZ JOHN PAUL
* MARTIN BARAJAS MORALES
* BENJAMIN NECISO
* LEONARDO FABIAN CONTRERAS JUAREZ

INTRODUCCION

Lo que nosotros buscamos para este proyecto final es un brazo que este hecho de madera y que con unos engranes reciclados pueda dar movimiento para poder levantar una lata de aluminio, para poder realizar esto es necesario tener unos motores reductores de 6 volts para el movimiento del brazo poder mejorarlo agregando el uso de la RaspBerry pi3 para que tenga un movimiento programado y por qué programado fácil para así poder en un futuro usar este brazo robótico en una línea de producción en la cual se utilizan movimientos repetitivos y exactos.

MATERIALES

1. 3 tablas de 14 cm
2. 3 motores reductores
3. Engranes reciclados
4. Tornillos con tuercas
5. Cables
6. Baquelita y circuito armado en simulador
7. Raspberry pi3
8. Módulo de relevadores

lo primero que hicimos fue recortar la madera alas piezas exactas del brazo para poder conectar los engranes y que coincidieran un poco después con un taladro chico hicimos las perforaciones en la madera para poder meter los tornillos.

Lo segundo fue conectar los tres motores reductores para que funcionaran con una batería de 9 volts y con un engrane en el centro del motor

LO QUE QUEREMOS MEJORAR

Lo que queremos hacer es cambiar el material que es de madera por uno de plástico así está un poco mejor para poder meterle mas motores y una mejor fácil de conectar con los engranes es necesario mejorar la conexión con los motores reductores también queremos mejorar las bases de la maqueta algo un poco mas grande o mas cuadrado mas pesado para que en ves de que levante una lata levante algo mucho mas pesado y pues que mejore también la conexión en la baquelita

Algo similar a este brazo



**Objetivo general**

Lo que buscamos nosotros es tener un mayor control en nuestro brazo robótico y mejorar el movimiento del mismo para poder así implementarlo en una línea de producción es por eso que el brazo tendrá una programación dedicada a levantar o colocar piezas pequeñas.

**Lo que se busca mejorar con nuestra automatización**

1. Queremos un mejor manejo
2. Tener mayor fuerza en el agarre
3. Tener una mejor movilidad en los motores
4. Mejorar la calidad de lo que realicemos con el robot (material de construcción).

**Materiales para programar**

1. Raspberry pi3
2. Módulo de relevadores
3. Computadora con el programa y librerías de Python
4. Cables
5. Dos motores reductores
6. Leds para darle más presentación al robot

**Automatización**

La importancia de la raspberry en nuestro proyecto es para poder darle una automatización y hacer que realice un patrón o cualquier tipo de movimiento y poder mejorarlo y controlarlo con una mejor precisión es necesario que la raspberry este conectado a nuestros motores reductores para tener el manejo requerido uno de los motores que seria el de la pinza sería el que tendría mejor agarre ya que gracias a eso se determinara el peso que queramos

**Marco teórico**

Investigamos que se puede programara con una raspberry con pocas entradas para que sea más económico también es una forma fácil de manejarlos usando una conexión de VNC a la laptop

**Referencias**

1. <https://www.google.com.mx/search?rlz=1C1CHBF_esMX812MX813&biw>
2. <http://www.infoplc.net/files/documentacion/automatas/infoPLC_net_1_Intro_Automatas.pdf>
3. <http://www.aldakin.com/aplicaciones-plc-industria-moderna/>

**presupuesto**

1. Cables $100
2. Dos motores reductores $500
3. Raspberry pi3 $1,000
4. Módulo de relevadores $60

**CÓDIGO DE CONTROL DE MOVIMIENTO**

From tkinter import \*

Import time

Import RPi.GPIO as GPIO

GPIO.setmodo(GPIO.BOARD)

GPIO.setup(11, GPIO.OUT)

GPIO.setup(13, GPIO.OUT)

GPIO.setup(15, GPIO.OUT)

GPIO.setup(19, GPIO.OUT)

GPIO.setup(21, GPIO.OUT)

GPIO.setup(23, GPIO.OUT)

P= GPIO.PWM(11, 50)

P2= GPIO.PWM(13, 50)

P3= GPIO.PWM(15, 50)

P4= GPIO.PWM(19, 50)

P5= GPIO.PWM(21, 50)

P6= GPIO.PWM(23, 50)

p.start (2.5)

p2.start (2.5)

p3.start (2.5)

p4.start (2.5)

p5.start (2.5)

p6.start (2.5)

class App:

def\_loit\_(self,master):

frame=frame(master)

frame.pack()

Label(frame, text=”51”).grid(row=0,columna=0)

Label(frame, text=”52”).grid(row=1,columna=0)

Label(frame, text=”53”).grid(row=2,columna=0)

Label(frame, text=”54”).grid(row=3,columna=0)

Label(frame, text=”55”).grid(row=4,columna=0)

Label(frame, text=”56”).grid(row=5,columna=0)

Scalel=scale(frame,from\_=0,to=180,orient=HORIZONTAL,command=sel

f.update1

scalel.grid(row=0,columna=1)

Scalel2=scale(frame,from\_=0,to=180,orient=HORIZONTAL,command=self.update1

Scale2.grid(row=1,columna=1)

Scalel3=scale(frame,from\_=0,to=180,orient=HORIZONTAL,command=sel

f.update1

scale.grid(row=2,column=1)

Scalel4=scale(frame,from\_=0,to=180,orient=HORIZONTAL,command=sel

f.update1

scale4.grid(row=3,columna=1)

Scalel5=scale(frame,from\_=0,to=180,orient=HORIZONTAL,command=sel

f.update1

scale5.grid(row=4,columna=1)

Scalel6=scale(frame,from\_=0,to=180,orient=HORIZONTAL,command=sel

f.update1

scale6.grid(row=5,columna=1)

def updatel(self,anglel):

duty=float(anglel)/18.0+2.5

p.changeDutyCycle(duty2)

def update2(self,angle2):

duty2=float(anglel)/18.0+2.5

p2.changeDutyCycle(duty2)

def update3(self,angle3):

duty23=float(anglel)/18.0+2.5

p3.changeDutyCycle(duty2)

def update4(self,angle4):

duty4=float(anglel)/18.0+2.5

p4.changeDutyCycle(duty2)

def update5(self,angle5):

duty5=float(anglel)/18.0+2.5

p5.changeDutyCycle(duty2)

def update6(self,angle6):

duty6=float(anglel)/18.0+2.5

p6.changeDutyCycle(duty2)

ventana = Tk()

ventana.WTB\_title(“servo control”)

app=App(ventana)

ventana.geometry(“740x300+0+0”)

ventana.mainloop()

descripción del funcionamiento de la programación

principalmente comenzamos importando las librerías visuales (Tkinter) con las cuales generaremos las interfaces visuales para que el usuario pueda controlar el brazo de una manera más sencilla y por ultimo librerías necesarias para el reconocimiento de las funciones en la raspberry pi3 (RPi.GPIO).

después de haber proporcionado las librerías en las que se trabajaran comenzamos a desarrollar el código comenzando por especificar las variables que se utilizaran en este caso sería enlazar variables con los pines (GPIO) de la raspberry así como valores que tendrán de inicio al usarlos y porque valores de inicio fácil por que como estamos trabajando con motores tenemos que especificar a los grados que inicia el motor.

Después de este paso comenzaremos a crear la interfaz grafica en esta interfaz utilizaremos etiquetas para saber qué es lo que realiza cada botón, agregaremos botones y todo esto estará empaquetado en un frame (ventana) es una interfaz muy simple pero intuitiva para el uso del usuario.

Como último paso a cada uno de los botones agregados le pondremos funciones y en estas funciones usaremos las variables anteriormente definidas esto enlazara al botón virtual con el PIN de la raspberry.

En estas funciones especificamos el grado en el que empieza a trabajar el motor y tendrán negaciones esto para que el motor pueda girar en sentido contrario al ya programado.

**Conclusión**

Este es un proyecto un poco innovador ya que utilizaremos material reciclado engranes reciclados plástico reciclado los cables, los motores y la raspberry sería lo único que compraremos.

Le daremos una programación sencilla reflejando lo aprendido en las materias de programación estructurada y controladores lógicos programables, pero con un buen funcionamiento final