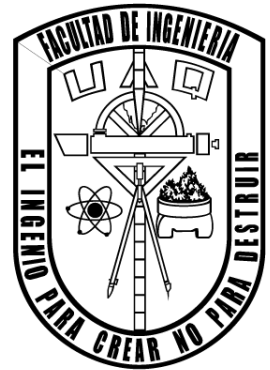
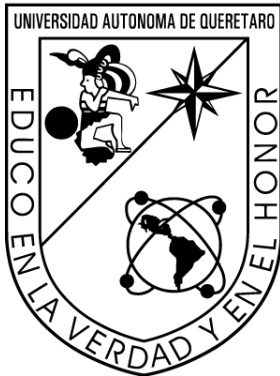


UNIVERSIDAD AUTÓNOMA DE QUERÉTARO



FACULTAD DE INGENIERÍA

Taller I

Profesor: Ing. Gilberto Bustamante Balderas.

Práctica No. 1.

Enfocador de Telescopio.

Alumnos: Colín Macías Luis Alberto,

Ortíz Méndez José Santiago,

Pizaña Juárez Luz Andrea,

Sinécio Rivera Juliette.

Fecha: 25 de junio de 2022

Índice

1. Antecedentes.	2
2. Marco teórico.	2
3. Objetivos.	2
4. Metodología.	2
5. Resultados.	6

1. Antecedentes.

Hace no más de medio año conseguí una cámara con la intención de acoplarla a mi telescopio para realizar fotografías. Al acoplar una cámara a un telescopio (sobre todo si cuenta con montura ecuatorial) se tiene que equilibrar el peso extra que da la cámara. Una de las maneras más sencillas y baratas de equilibrar la cámara es colocarla de forma vertical con respecto al eje de ascensión recta.

Eso presenta dos inconvenientes: El primero es una altura extra que dificulta el poder observar con el visor de la cámara (en caso de ser reflex) y el segundo es la disminución del campo visual con el nuevo ocular que pasa de aproximadamente 60° a únicamente 20°. Ambos problemas se pueden solucionar de una manera muy sencilla y disponible en la mayoría de las cámaras, el sistema Live View de Canon y análogos para otras marcas nos permite ver lo que verían nuestros ojos desde una pantalla que podemos colocar en una posición más cómoda.

Con esos problemas resueltos nos surge un problema nuevo. Cuando vemos a través de un ocular nos es muy sencillo ocupar las manos para ajustar la perilla del enfoque y darnos una imagen nítida, pero qué pasa si no estamos viendo con un ocular y en cambio ocupamos una pantalla que puede estar lejos del alcance de la perilla. Nuestro proyecto viene para tratar de disminuir esta problemática.

2. Marco teórico.

Las placas micro-controladoras, muchas piezas electrónicas y en general las computadoras son capaces de utilizar el método de comunicación síncrono **Serial**, este método envía la información un byte a la vez. Al conectarse por un número limitado de cables solamente permite tener un dispositivo maestro por cada dispositivo esclavo.

Una de las placas micro-controladoras que puede utilizar la comunicación serial es la Raspberry Pi Pico, que además de comunicarse de esa manera nos permite ejecutar programas Micropython que es una versión disminuida del famoso lenguaje de programación Python.

3. Objetivos.

El problema a resolver con este proyecto surge de la necesidad de enfocar con precisión un telescopio de manera semi-remota, intentando evitar en la medida de lo posible las interrupciones visuales como lo puede ser una vibración.

4. Metodología.

Explicación:

Para solucionar el problema debemos introducir una pieza capaz de emular el movimiento que nosotros realizamos con la mano al girar la perilla del enfoque. Como el movimiento que buscamos es un giro es que utilizaremos un motor a pasos, se pensó en alternativas como un servo motor y un motor de inducción pero ninguno cumplía con la necesidad que nos presenta el enfocador, pues debe de permitir dar más de una vuelta completa y además debe de ser preciso. Además de esas dos necesidades a cumplir debemos obtener un torque capaz de cargar el peso de la cámara, eso descarta el modelo 28byj-48 que es más barato, pero solo puede otorgarnos 0.34Kg/cm y nos deja con el modelo 17hs4401 que es capaz de otorgarnos 4Kg/cm.

De nada nos sirve tener un movimiento que emule el nuestro si no somos capaces de transmitirlo a la pieza que necesitamos mover. Para transmitir el movimiento ocupamos un cople de motor comúnmente ocupado en modelismo. Para hacer que nuestro motor se mueva necesitamos mandarle las órdenes de movimiento y la corriente que necesita para funcionar para eso ocupamos el driver A4988 y la alimentación viene desde la micro-controladora, posteriormente se cambiará la

fuelle para evitar fallos y otorgar una mayor potencia al motor.

Aquí entra nuestra tarjeta micro-controladora Raspberry Pi Pico que nos permite realizar el puente entre nuestro ordenador y nuestro driver del motor. Para hacer que funcione ocupamos el código **main.py**.

Por último tenemos nuestra interfaz gráfica que corre dentro de un ordenador la cual nos permite enviar ordenes a la tarjeta microcontroladora la interfaz gráfica se encuentra en el archivo **enfocador.py**. Se puede ocupar cualquier ordenador siempre y cuando tenga instalado Windows (por el momento no hay compatibilidad con Linux), cuente con Python3 y tenga disponible un puerto USB para conectar la tarjeta.

Código.

Código del programa instalado en la Raspberry Pi Pico.

```
1 import machine
2 import time
3
4 reset = machine.Pin(13, machine.Pin.OUT)
5 step = machine.Pin(14, machine.Pin.OUT)
6 direction = machine.Pin(15, machine.Pin.OUT)
7 led = machine.Pin(25, machine.Pin.OUT)
8 sensor = machine.ADC(4)
9
10 def temperatura():
11     return 27-(sensor.read_u16()*3.3/65535-0.706)/0.001721
12
13 def pro_horario(velocidad, pasos):
14     reset.value(1)
15     for i in range(int(pasos)):
16         step.value(1)
17         step.value(0)
18         time.sleep(float(velocidad))
19     reset.value(0)
20
21 def con_horario(velocidad, pasos):
22     direction.value(1)
23     reset.value(1)
24     for i in range(int(pasos)):
25         step.value(1)
26         step.value(0)
27         time.sleep(float(velocidad))
28     reset.value(0)
29     direction.value(0)
30
```

v3/main.py

Parte del código de la interfaz gráfica.

```
1 from tkinter import *
2 from tkinter import ttk, messagebox
3 import os
4 import _thread
5 import time
6 from sender import Sender
7
8 ##### RAICES #####
9 raiz = Tk()
10 raiz.title('caralacm v3.1')
11 raiz.resizable(False, False)
12
13 if(os.name=='nt'):
14     raiz.iconbitmap(bitmap='escudo.ico')
15 elif(os.name=='posix'):
16     messagebox.showinfo(title='Advertencia', message='Sistema Operativo no soportado.')
17     raiz.destroy()
```

```

18     exit()

20 ##### MARCOS #####
marco1 = Frame(raiz) # CONTIENE IMAGEN FONDO
22 marco1.pack(fill='both', padx=5, pady=5)

24 marco2 = Frame(raiz) # CONTIENE BOTONES Y DESCRIPCIONES
marco2.pack(expand=True, padx=5, pady=5)

26 marco3 = Frame(raiz) # CONTIENE INDICADORES DE CONEXION
28 marco3.pack(expand=True, padx=5, pady=15)

30 ##### GLOBALES #####
32 estado_boton = False
estado_conexion = False

34 etiqueta_conexion = StringVar(value='Sin Conexión')

36 combo = ttk.Combobox(marco3, state='readonly', values=[f'COM{i}' for i in range(1, 100)],
    width=7)
38 combo.grid(row=0, column=3, sticky='ns')
combo.set('COM19')

40 def temperatura():
    while(True):
42         if not(estado_boton):
            enviador.send('temperatura()')
44             temperatura = float(enviador.receive())
            print(temperatura)
46             if(temperatura > 40):
                messagebox.showerror(title='Advertencia', message='Temperatura muy alta.')
48                 raiz.destroy()
                exit()
50         time.sleep(1)

```

v3/enfocador.py

En este código vemos la presencia de la clase Sender [1], esta es utilizada para comunicar la interfaz con la tarjeta.

Conexiones.

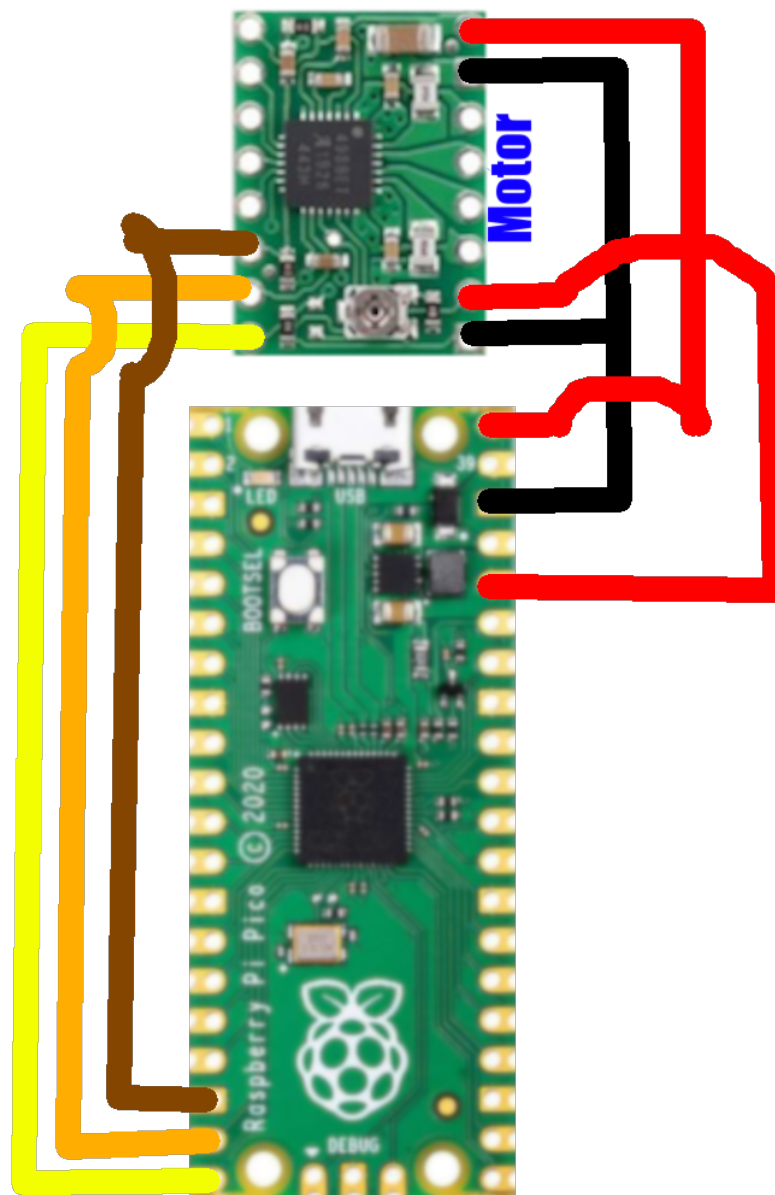
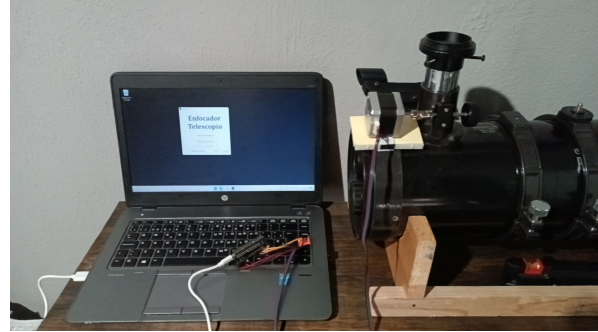
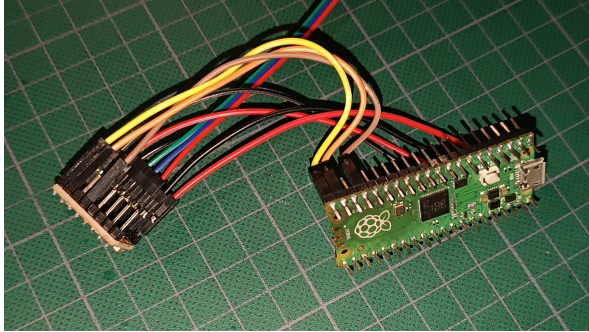


Figura 1: f. [3] [4] Esquema de la conexión.

Solo hay que mencionar que la entrada del motor debe ser en el mismo orden de los cables.

5. Resultados.

El proyecto nos permite realizar la función para la cual fue destinado, sin embargo no presenta la precisión tan exacta que se pensaba presentaría, para tener una mejor precisión será necesario cambiar la fuente del poder que va conectada al driver del motor y configurar el driver para que el movimiento sea más pequeño.



Referencias

- [1] Cocking, R. (2022, 25 junio). Controlling a Raspberry Pi Pico remotely using PySerial. RAREblog. Recuperado 22 de mayo de 2022, de <https://blog.rareschool.com/2021/01/controlling-raspberry-pi-pico-using.html>
- [2] Colín, L. (s. f.). GitHub - CARALACM/Enfocador-Telescopio: Sirve para enfocar un telescopio. GitHub. <https://github.com/CARALACM/Enfocador-Telescopio>
- [3] García, V. (s. f.). Esquema práctico general. [Esquema]. Electrónica Práctica Aplicada. <https://www.diarioelectronicohoy.com/blog/imagenes/2020/07/a4988.jpg>
- [4] Raspberry Pi Pico. (s. f.). [Fotografía]. Tekno Movo. <http://teknomovo.com.mx/wp-content/uploads/2021/05/rasp-pico.jpg>