# Read first:

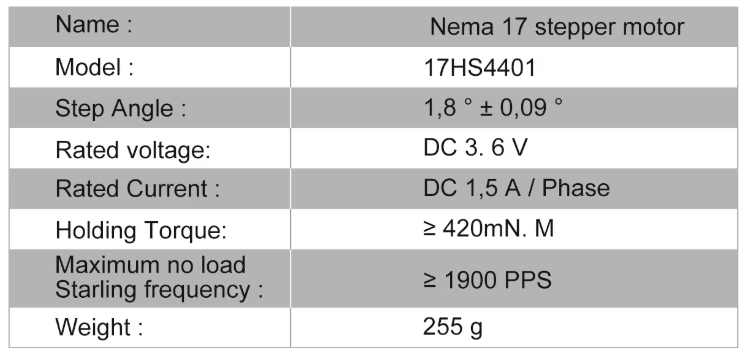
* Ojo a conexión motor – controlador
* Regular potenciómetro controlador
* OJO CON LOS CABLES, REGULAR EL GROSOR PARA GARANTIZAR EL AMPERAJE NECESARIO

# Motor

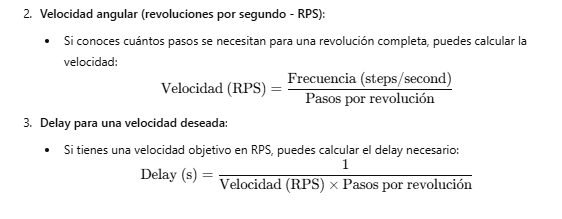
Nema 17 1.5A (17HS4401) 42N

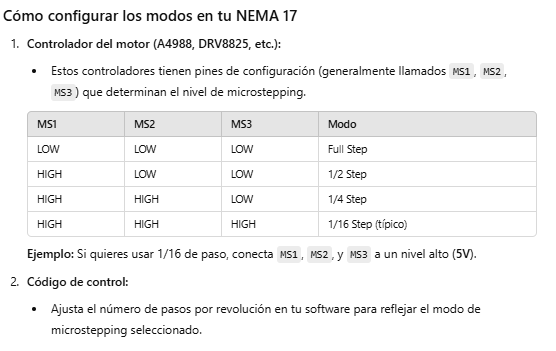
* \*\*\*Conexión: A+: Negro, A-: Azul, B+: Verde, B-: Rojo\*\*\*

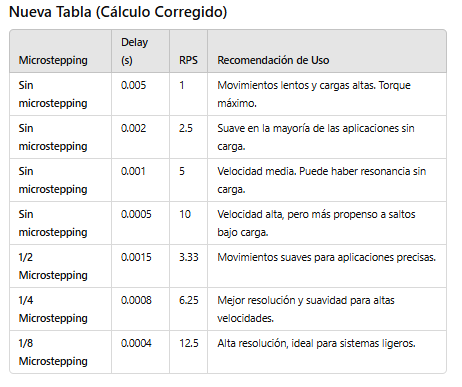
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Motor | Controlador |  | Motor | Controlador |
| A+ Negro | 1A |  | A- Verde | 1B |
| A- Verde | 1B |  | A+ Negro | 1A |
| B+ Rojo | 2A |  | B+ Rojo | 2A |
| B- Azul | 2B |  | B- Azul | 2B |



Velocidad de rotación: 200 pasos por revolución



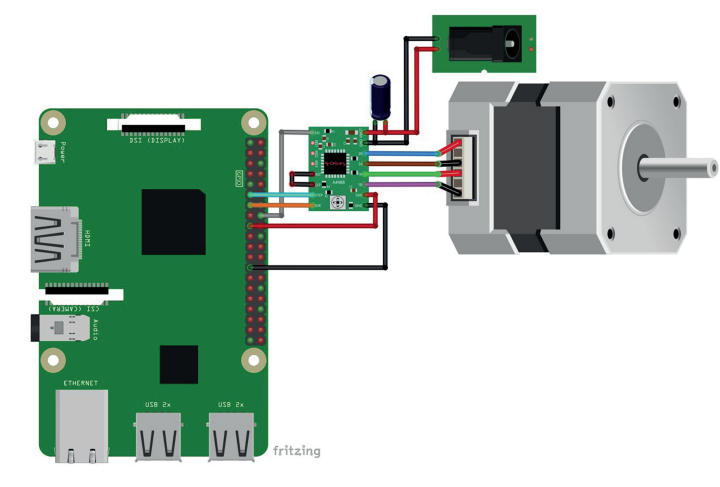


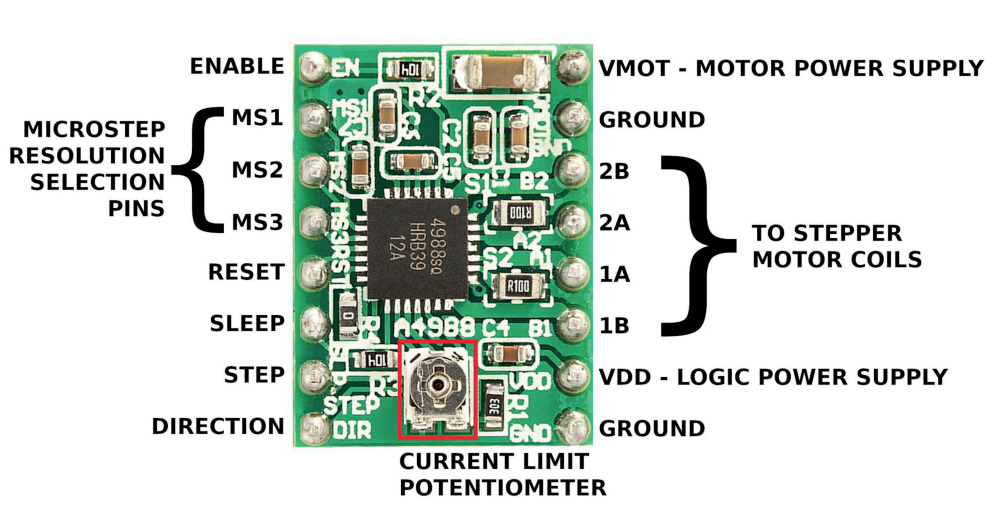


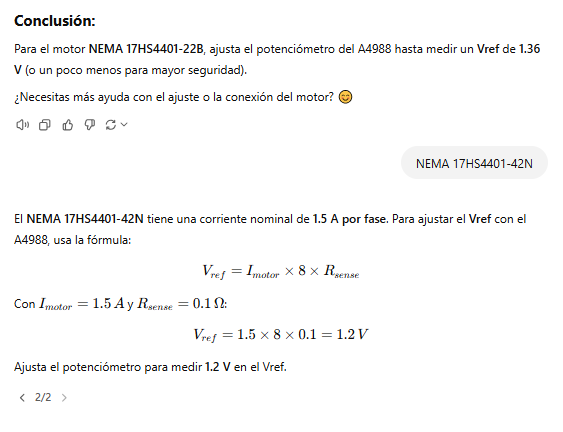
|  |  |  |  |
| --- | --- | --- | --- |
| Microstepping | Delay (s) | RPS | Recomendación de Uso |
| Sin microstepping | 0.005 | 1 | Movimientos lentos y cargas altas. Torque máximo. |
| 1/2 Microstepping | 0.0015 | 3.33 | Movimientos suaves para aplicaciones precisas. |
| Sin microstepping | 0.002 | 2.5 | Suave en la mayoría de las aplicaciones sin carga. |
| Sin microstepping | 0.001 | 5 | Velocidad media. Puede haber resonancia sin carga. |
| 1/4 Microstepping | 0.0008 | 6.25 | Mejor resolución y suavidad para altas velocidades. |
| Sin microstepping | 0.0005 | 10 | Velocidad alta, pero más propenso a saltos bajo carga. |
| 1/8 Microstepping | 0.0004 | 12.5 | Alta resolución, ideal para sistemas ligeros. |

# Controlador (A4988)

**condensador de 100 µF**

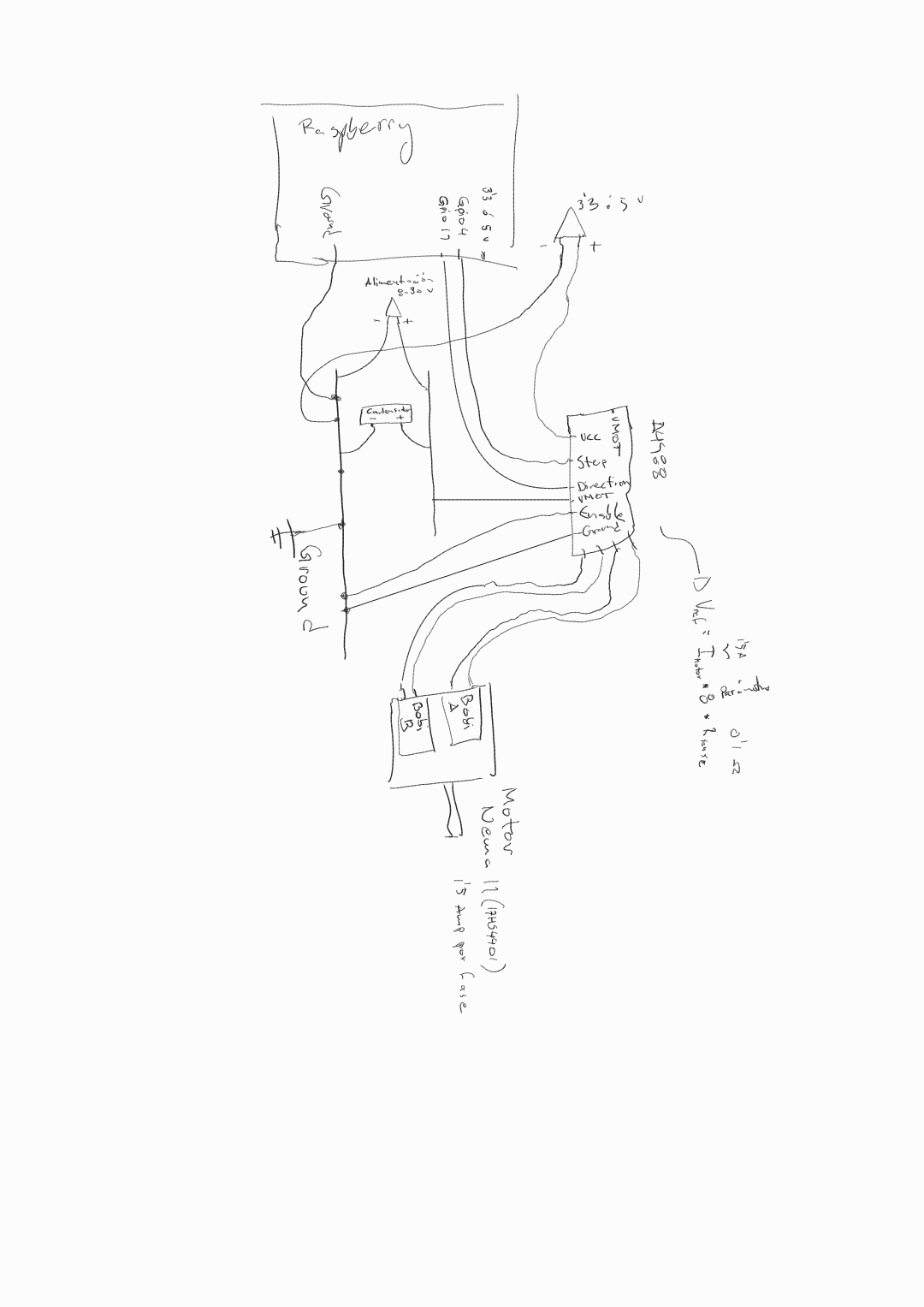




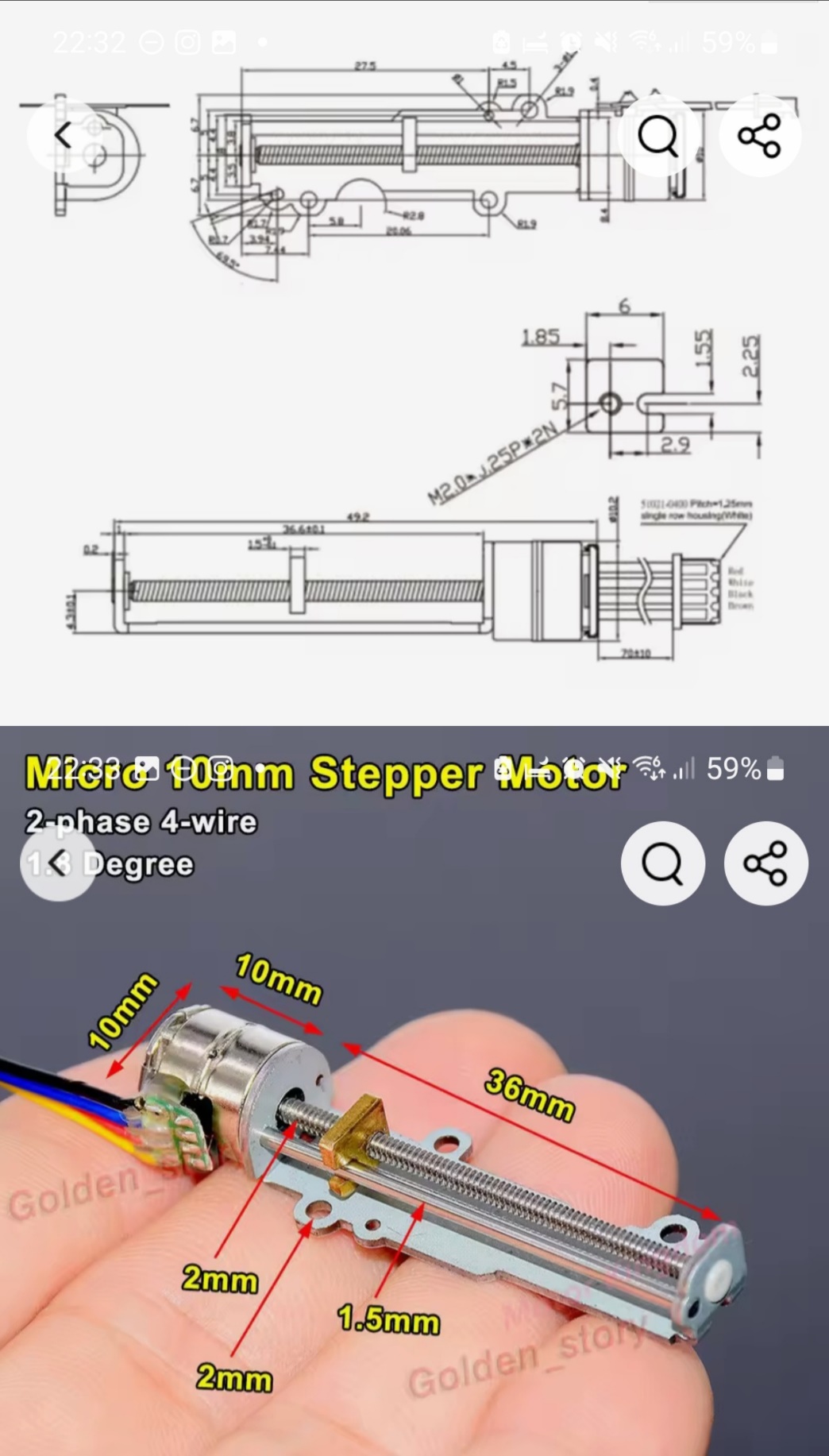


# Electrónica:

Condensador de



# Posicionador



A confirmar:

20 pasos por revolución

Diámetro del motor paso a paso: 10mm

Longitud del eje de salida: 36mm Carrera 34mm

Diámetro del eje del tornillo 2mm

Diámetro de la varilla: 1,5mm

Ángulo De paso: 18 grados/Paso

300mA por fase

5 V

Motor Híbrido

Longitud del cable: 80mm

Paso de rosca: 0,23mm

Rojo A +, Blanco A-, negro B +, Marrón B-

# Programacio:

Código base para probar motor:  
  
import RPi.GPIO as GPIO

import time

# Configuración de los pines GPIO

STEP\_PIN = 17 # Pin GPIO para STEP

DIR\_PIN = 27 # Pin GPIO para DIR

# Configuración inicial de GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(STEP\_PIN, GPIO.OUT)

GPIO.setup(DIR\_PIN, GPIO.OUT)

def move\_motor(steps, direction, step\_delay):

"""

Mueve el motor NEMA 17 un número de pasos en la dirección especificada.

:param steps: Número de pasos a mover.

:param direction: Dirección del giro (True = horario, False = antihorario).

:param step\_delay: Retardo entre pasos (en segundos).

"""

# Configurar la dirección

GPIO.output(DIR\_PIN, direction)

# Generar pulsos para el motor

for \_ in range(steps):

GPIO.output(STEP\_PIN, GPIO.HIGH)

time.sleep(step\_delay) # Tiempo en HIGH

GPIO.output(STEP\_PIN, GPIO.LOW)

time.sleep(step\_delay) # Tiempo en LOW

try:

print("Iniciando movimiento del motor...")

# Girar en sentido horario 200 pasos (una revolución completa para un motor de 1.8° por paso)

print("Giro horario...")

move\_motor(200, True, 0.001) # 200 pasos, 1ms de retardo por paso

time.sleep(1) # Pausa de 1 segundo

# Girar en sentido antihorario 200 pasos

print("Giro antihorario...")

move\_motor(200, False, 0.001) # 200 pasos, 1ms de retardo por paso

except KeyboardInterrupt:

print("Movimiento interrumpido por el usuario.")

finally:

# Limpieza de los pines GPIO

GPIO.cleanup()

print("GPIO limpio. Programa terminado.")