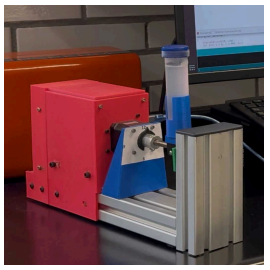
May 27, 2025

PROTOCOL FOR THE USE OF THE CASHo DEVICE



PLOS One

DOI
https://dx.doi.org/10.17504/protocols.io.4r3l29ew3v1y/v1

Nancy Juárez[1], Cindy Rivas Arzaluz[2], Oscar Gutierrez Perez[3], María . Ayala[4], Claudia Treviño[2], Andrés Aragón Martínez[1]

[1]Laboratorio de Gametos y Desarrollo Tecnológico, Facultad de Estudios Superiores Iztacala, UNAM;
[2]Consorcio de Fisiología del Espermatozoide, Instituto de Biotecnología, UNAM;
[3]Centro de Enseñanza, Investigación y Extensión en Producción Porcina (CEIEPP), FMVZ, UNAM;
[4]Unidad de Biología de la Reproducción, Laboratorio de Pubertad, Facultad de Estudios Superiores Zaragoza, UNAM

 Cindy Rivas

OPEN ACCESS

DOI: https://dx.doi.org/10.17504/protocols.io.4r3l29ew3v1y/v1

Protocol Citation: Nancy Juárez, Cindy Rivas Arzaluz, Oscar Gutierrez Perez, María . Ayala, Claudia Treviño, Andrés Aragón Martínez 2025. PROTOCOL FOR THE USE OF THE CASHo DEVICE. **protocols.io** https://dx.doi.org/10.17504/protocols.io.4r3l29ew3v1y/v1

Manuscript citation:
Welcome to the open-hardware sperm quality lab, featuring the new homogenizer: testing serotonin's effects on sperm motility. Submitted to PLOS ONE, June 04, 2025.

Protocol status: Working
We use this protocol and it's working

Created: January 09, 2025

Last Modified: May 27, 2025

Protocol Integer ID: 118019

Keywords: Homogenization of semen samples, Open-Source hardware, Objective homogenization, sperm motility, sperm samples, casho device proper semen homogenization, use of the customizable automated sperm homogenizer, customizable automated sperm homogenizer, objective sperm homogenization, semen homogenization, proper semen homogenization, semen sample, spermatozoa, correct number of spermatozoa, semen dose for insemination, casho device, semen dose, insemination, laboratory of gamete, accurate homogenization, cashoe, available for general lab use, crucial for animal production, general lab use, laboratory, sample

Disclaimer

Abstract

Proper semen homogenization is crucial for animal production and experimental work. Accurate homogenization ensures the correct number of spermatozoa per semen dose for insemination or representative samples for research. Typically, semen homogenization is performed manually with a "gentle" approach or using commercial devices not specifically designed for this purpose. While open-source devices for rotating, mixing, and shaking are available for general lab use, they are not optimized for objective sperm homogenization. The goal of this protocol is to provide the steps to follow for the use of the Customizable Automated Sperm Homogenizer (CASHo), a device designed and built in the Laboratory of Gametes and Technological Development to homogenize semen samples. At the end of protocol, you will find an example for customizing the firmware.

Materials

1. Computer
2. CASHo device
3. Sperm sample
4. Tube 50 mL

Before start

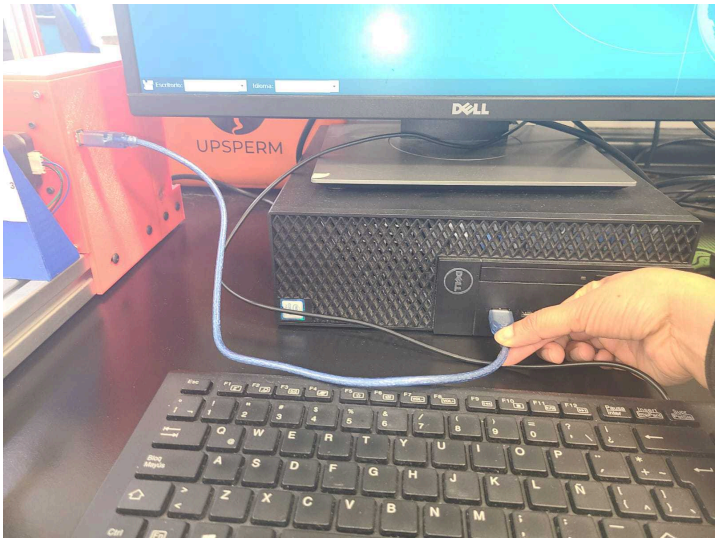
Requirements

1. Computer with the Arduino IDE
2. Upload the code of firmware to the Arduino UNO. That code is in Homogenizer_v6.ino file in https://osf.io/esfwg/. Upload the firmware is made one time unless the user made changes to the code in Homogenizer_v6.ino.

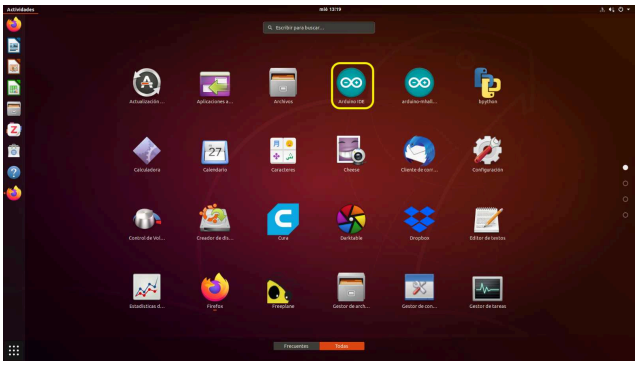
Steps to operate the CASHo device

1 Turn on the computer

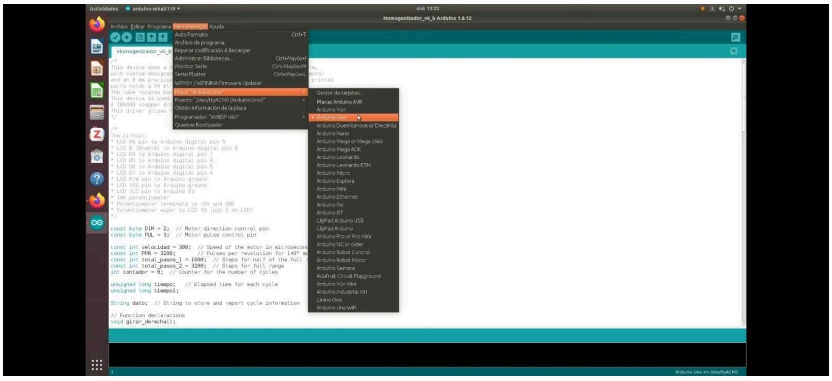
2 Connect the USB cable to the appropriate port on the computer

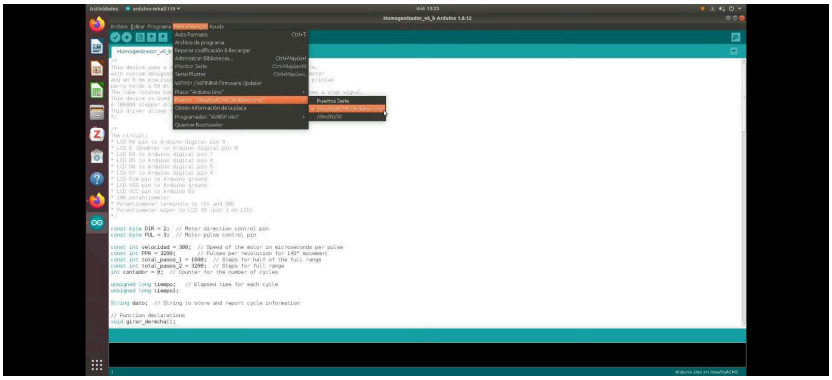


3 Open the Arduino IDE on your computer and open the Homogenizer_v6.ino file



3.1 Verify communication between the Arduino IDE and the Arduino UNO board. Go to the "Tools" menu and make sure the Arduino UNO board is selected and that the correct port is assigned (e.g. /dev/ttyACM0)



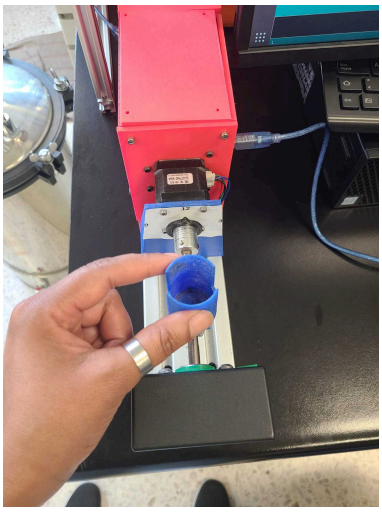


4 Ensure that the tube holder is in the vertical (default) position.

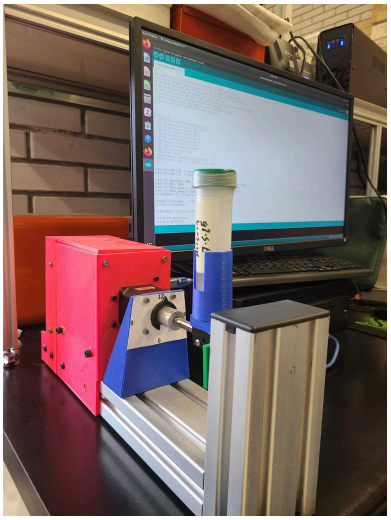
Note

The current version of tube holder is for 50 mL tubes. A tube holder for other capacity, can be attached to the rod adapter (See the manuscript *"Welcome to the open-hardware sperm quality lab, featuring the new homogenizer: testing serotonin's effects on sperm motility"*).

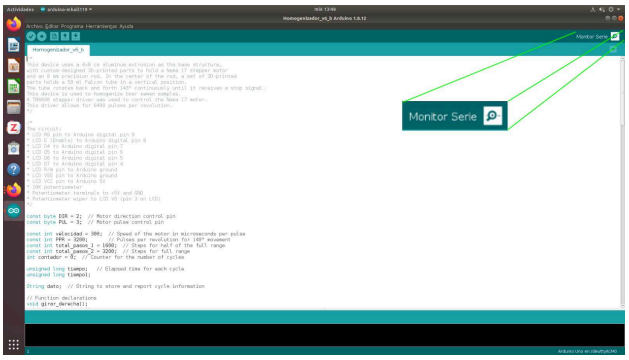
5 Connect the CASHo power supply to power the stepper motor controller. Hold the tube holder with your fingers and ensure that the piece does not move.

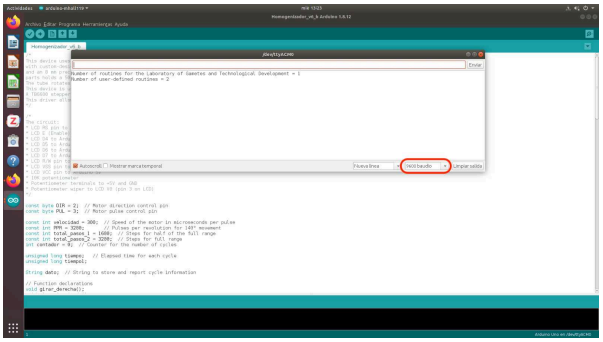


6 Insert a tube containing diluted semen into the tube holder.

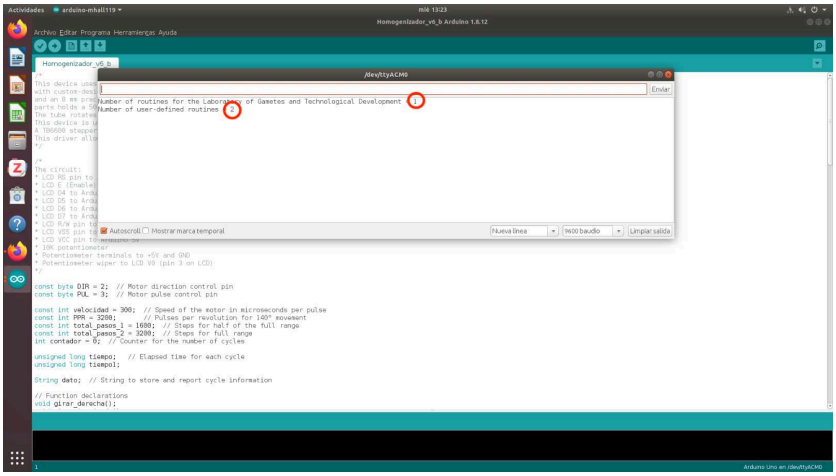


7 Open the Serial Monitor in the Arduino app (upper right corner in the main window) and confirm that communication is set to 9600 baud (bottom right corner in the Serial Monitor, see the red oval).

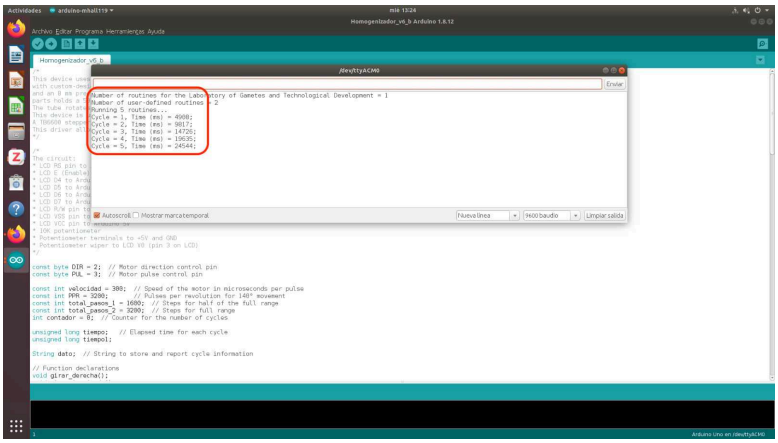




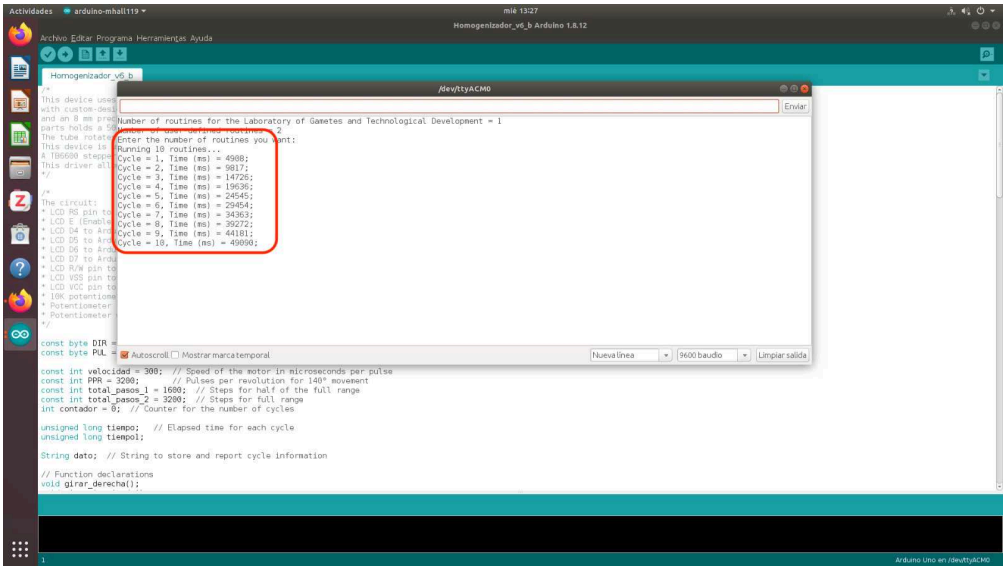
7.1 Type 1 or 2 to choose between options:



1 = Is the number of cycles used in the manuscript *"Welcome to the open-hardware sperm quality lab, featuring the new homogenizer: testing serotonin's effects on sperm motility"* (5 cycles).



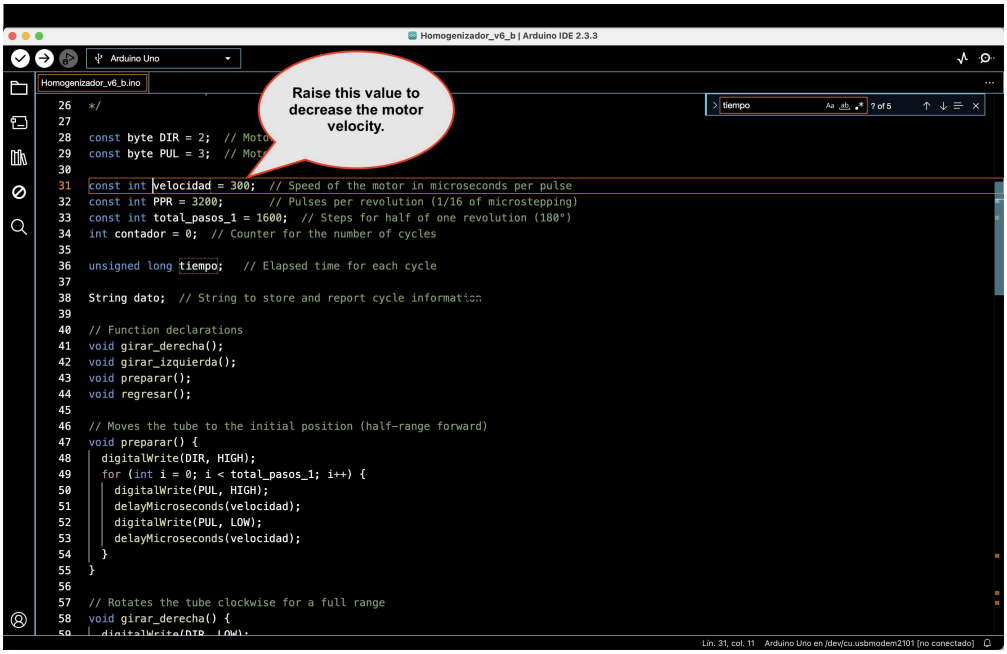
2 = This option allow choose a number of cycles defined by the user.
The motor shaft will begin to rotate, and the movement of the tube holder will be visible.
In the Serial Monitor output, you will see the number of cycles and the duration (in seconds) of each cycle.
In this case were 10 cycles.



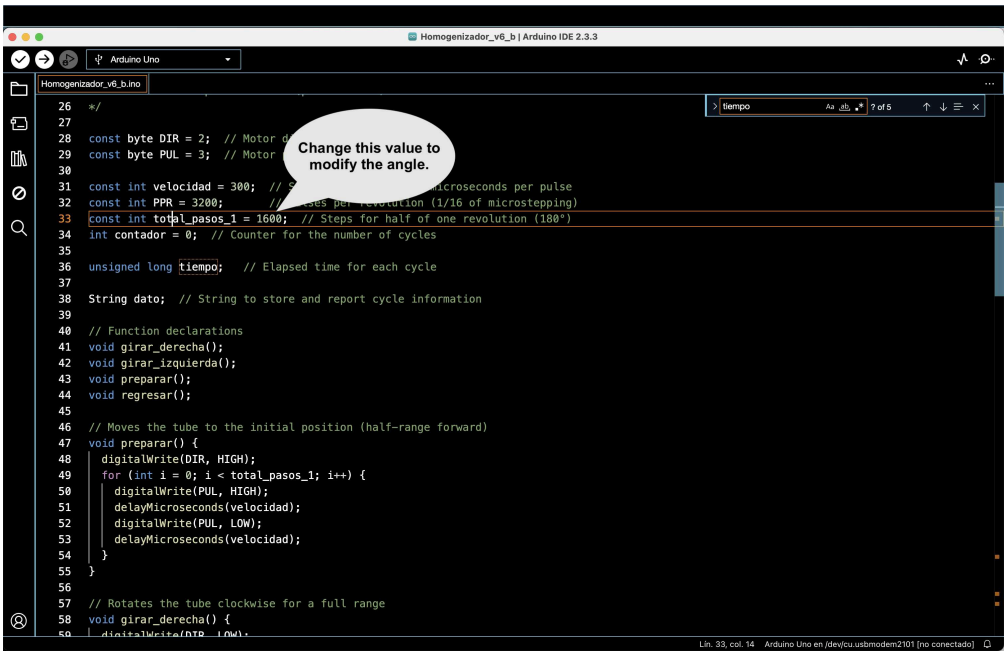
8 Once the routine is complete, close the Serial Monitor window, exit the Arduino IDE, and disconnect both the USB cable and the power supply.

Firmware customization (example)

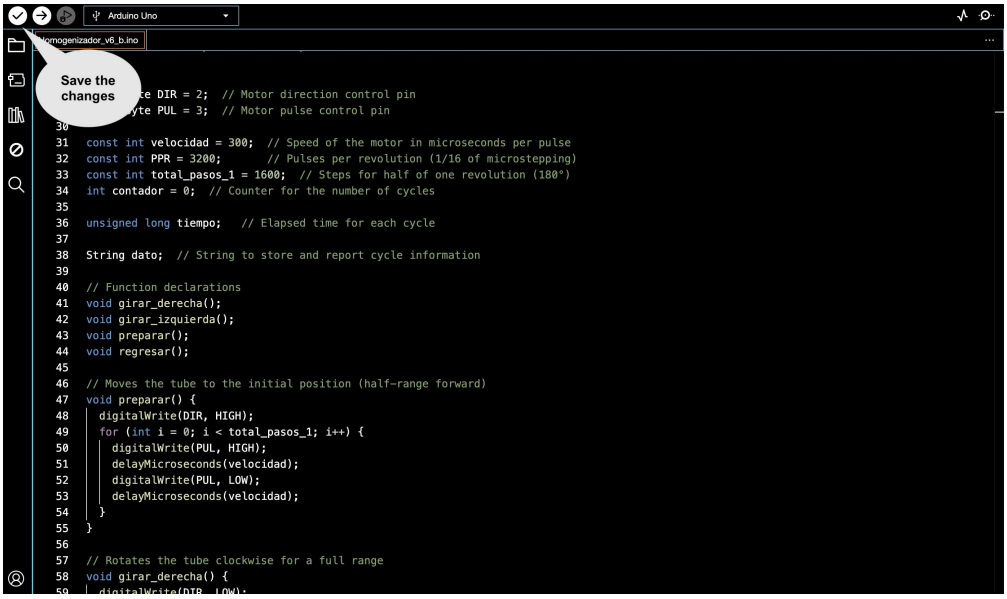
9 You can customize the device's firmware depending on your needs or the type of sample to be homogenized.
If you want to modify the velocity of the motor, you must change the value of "velocidad" variable in line 31, as indicated by the bubble text. Increase the value if you want decrease the velocity of the motor.



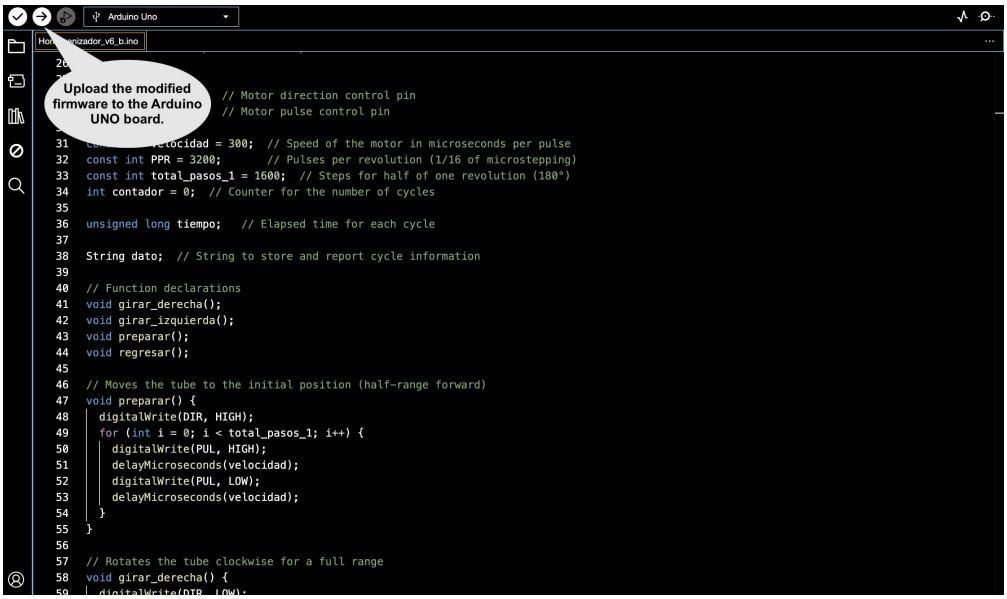
If you want to modify the angle of the motor, you must change the value of "total_pasos_1" variable in line 33, as indicated by the bubble text. 3200 steps per second give a revolution; thus, if you want change the value of angle you must modify this value. In our case 1600 steps were setting to obtain an angle of 180°.



To apply the changes, they must be saved.



Finally, upload the firmware to the Arduino UNO board to use the customized version.


Arduino Uno
Upload the modified firmware to the Arduino UNO board.
// Motor direction control pin
// Motor pulse control pin
velocidad = 300; // Speed of the motor in microseconds per pulse
const int PPR = 3200; // Pulses per revolution (1/16 of microstepping)
const int total_pasos_1 = 1600; // Steps for half of one revolution (180°)
int contador = 0; // Counter for the number of cycles
unsigned long tiempo; // Elapsed time for each cycle
String dato; // String to store and report cycle information
// Function declarations
void girar_derecha();
void girar_izquierda();
void preparar();
void regresar();
// Moves the tube to the initial position (half-range forward)
void preparar() {
digitalWrite(DIR, HIGH);
for (int i = 0; i < total_pasos_1; i++) {
digitalWrite(PUL, HIGH);
delayMicroseconds(velocidad);
digitalWrite(PUL, LOW);
delayMicroseconds(velocidad);
}
}
// Rotates the tube clockwise for a full range
void girar_derecha() {

go to step #7