

English version (Spanish version at the end) updated: 2022-07-30

Peristaltic pump

The Arduino code is based on <https://www.thingiverse.com/thing:2619479/files> and modified to meet my needs, especially related to calibration. We do not need to fill the vials with an exact volume, so I designed a “by eye” calibration method.

Two critical points to consider:

- Use PETG to print rotor pieces. It resists quite well up to 85°C. On the other hand, PLA, the most frequently used material for 3D printing softens at 65°C. The rest of the pieces were printed with PLA and work fine, at least for the moment.
- Food temperature: We cook the fly food at 95°C for 30 minutes. That high temperature softens PLA, and probably PETG. We now cook the food at 95 and let it cool (and maintain it at) 75°C to deliver into the vials. It is very important to keep food at a constant temperature to deliver a reproducible volume. Food viscosity increases when temperature decreases and if the pump has been calibrated at a high temperature to deliver a given volume in a given number of rotations, then the volume delivered by the same number of rotations will be smaller when the food cools down.

The design of the rotor is a modification of <https://www.thingiverse.com/thing:3148717>. I followed the designer recommendation to print with 100% infill. Remember to use PETG, at least for the rotor.

Two different calibration values can be saved in the programs named “moscas” (flies) and “gusanos” (worms). If necessary, those names can be changed in lines 161 and 162 of the code.

The pressure of the cover over the hose is adjusted with two long screws. If the cover is too loose, the rotor will not make enough pressure onto the hose and the food will not be pumped. If it is too tight, it will block the motor. The exact position should be found by trial and error. If you find that my design does not work in your hands, you can try with the original one <https://www.thingiverse.com/thing:3148717>.

A few comments on the electronic circuit. There are three things that are optional and can be removed. The first one is the possibility to activate the pump with a pedal. We found that it was not necessary after adding the option to repeat pumping cycles to the code. The second one are the three LEDs that indicate the mode of operation that is selected. The same information is shown in the screen. The last one is the fan to cool the motor and the motor driver. It does not seem necessary since none of them gets too hot during operation.

Bill of Materials (with links to suppliers in Argentina)

- 1- Stepper motor, 1.5A high Torque 4.2kg 17HS4401S x1 <https://articulo.mercadolibre.com.ar/MLA-899670687-nema-17-motor-paso-a-paso-18-alto-t-15a-usongshine- JM>
- 2- Bearings 605zz x9 <https://articulo.mercadolibre.com.ar/MLA-915854069-rodamiento-ruleman-605-zz-x20-unidades- JM>
- 3- driver A4899 x1 <https://www.todomicro.com.ar/motores-y-drivers-motores/288-modulo-a4988-pololu-driver-motor-paso-a-paso.html>
- 4- arduino promini 5V (it will also work with proMini 3.3V or arduino UNO) x1
- 5- Power source 12V 2A x1
- 6- step down regulator MP1584 x1
- 7- 16x2 LCD display with I2C adapter x1
- 8- Rotary encoder Ky-040 x1
- 9- LEDs 3mm (optional) x3

- 10- 100uF capacitor x1
- 11- resistors 220 Ohm, 10kOhm
- 12- Fan, 6cm (optional) x1
- 13- Plastic box: it can be printed, I used a 20x30 cm for electrical installations
- 14- silicone hose, external diameter 10mm, inner diameter 6mm (walls 2mm).
- 15- 15- nuts, bolts, screws, etc

It is critical to limit the current through the motor driver!

This is VERY important. If this is not correctly done the motor can be damaged. For a detailed explanation, visit this webpage: <https://www.pololu.com/product/1182> I used a cheap driver that does not have the same resistor values as the original one. This has to be taken into account when setting the current limit. I provide the details of the procedure for my motor and driver, you should do your own calculations:

Motor 17HS4401S

Max current 1.5A

For a A4988 driver: $I_{max} = V_{ref}/8 * \text{limiting resistor}$.

I_{max} is 1.5

Resistor: R100 0.1Ohm or 100 mOhm

$V_{ref} = 1.2V$

The driver is configured to work in a quarter steps mode with a logic level of 5V. Turn the potentiometer of the driver until you get the V_{ref} . In my case, I had to put the potentiometer at the max level to get a V_{ref} of 1.19V.

User instructions: The LCD display shows a menu with 11 options that can be navigated turning the knob clockwise or counterclockwise. Menu options are selected by pressing the knob once. Once inside the option, turn the knob to set values and press again to leave. If the knob is pressed continuously the motor will move at a constant speed until the knob is released. That will let the operator fill the hose with food. The pump has three operation modes: Dosifica (dosing), bombea (pump), calibra (calibration).

Menu 0: Start--> starts the selected mode (dosifica/bombea/calibra). Pressing a second time stops the pump once the current cycle is finished.

Menu 1: Tiempo (time)--> Sets the time required to fill one vial. When this value is set, the speed expressed as RPM is automatically calculated. If the resulting speed is > 200 RPM, the time is decreased to get a max speed of 200 RPM.

Menu 2: Velocidad (speed)--> Sets the speed, expressed in RPM, and calculates the time required to fill one vial.

Menu 3: Ciclos (cycles) --> Number of filling cycles that will be executed when pressing START (number of vials to fill). Max value is 200. This value can be increased in line 139 of the code.

Menu 4: Pausa (pause) --> Resting time between two consecutive cycles.

Menu 5: Modo (mode)--> It has three options. Dosifica (= dosing. The motor will move a given number of turns defined during the calibration and stop for the time established in menú 5. The cycle is repeated the number of times indicated in menú 4). Bombea (=pump. The motor turns at a constant speed) and Calibra (=calibration. The pump executes a calibration cycle and returns to the dosing mode)

Menu 6: PGM --> Used to store two different dispensing volumes (ie two different calibration values)

Menu 7: Sentido (turning direction)--> H (Horario: clockwise) and AH (AntiHorario: counter clockwise)

Menu 8: M--> is the number of steps that the motor will move in the program “Moscas” (200 steps is one turn. In fact, as the driver is set at quarter steps, 800 steps are required to move the motor one turn. The number set in this menu is internally multiplied by 4).

Menu 9: G--> Same as above, but for the program Gusanos. Both 9 and 10 can be manually set, and the speed will be automatically calculated to complete the cycle in the time indicated in menu 2 (time).

Menu 10: Saves the configuration in the memory of the Arduino board. The values of each parameter are loaded from memory at start. After calibration (or changing speed, cycles, puase time, etc) the new settings can be saved by selecting this option.

How to use the pump?

Place the silicone hose between the rotor and hose cover and tighten the cover with the screws until an optimal position is reached. An optimal position is that at which the bearings are tightly adjusted to hose and the motor is not blocked. (<https://youtu.be/28UxOIC15p4>)

Calibration

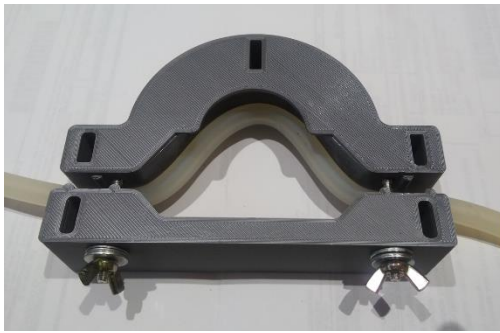
Select the program to calibrate in menú 7 (Moscas or gusanos. Or, for example, big and small vial) Select the option “calibrar” in menú 6. The LCD will show the message “Presionar hasta llenar el tubo” (press until the vial is filled). Place the end of the hose inside one empty vial and keep the knob pressed. The motor will begin to move slowly and pump the food. Release the knob when the desired level of food is reached. The program will save the number of turns and calculate how long it will take to fill one vial (menu 2) at the speed set in menu 3. The number of turns and steps will be diplayed for a few seconds and will automatically change to the dosing mode.

Pumping

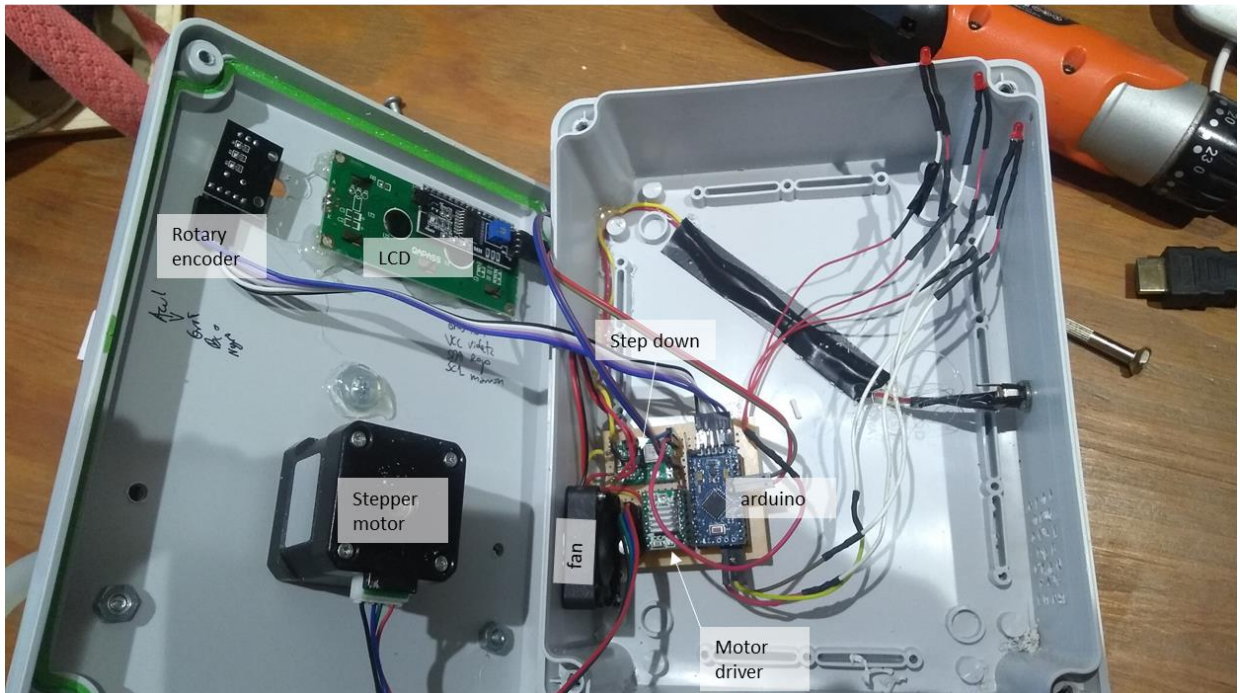
Go to menú 1 and press the knob. The motor will move and the Word RUN will be displayed in the screen. If the mode “bombea” is selected, the motor will rotate until the knob is pressed again. If it is in mode “Dosifica”, the motor will move for one cycle, make a pause and repeat the process as many times as indicated by menu4. Press the knob to stop dosing. Remember that the motor will stop once the current cycle has finished.



Top, bottom, and side view of the rotor.



Hose cover



Versión en castellano

Bomba peristáltica

El código del arduino está basado en <https://www.thingiverse.com/thing:2619479/files> y modificado según mis necesidades, en particular, la forma de calibración. Nosotros no necesitamos un volumen exacto y además la consistencia de la comida varía bastante en función de la temperatura, por lo que el ajuste para que dispense un cierto volumen puede cambiar bastante en función de la temperatura de la comida. Por eso implementé un calibrado “a ojo” que es más que suficiente para mis necesidades.

Hay dos puntos críticos a tener en cuenta: -material utilizado para imprimir las piezas: el PLA se ablanda a 65°C aproximadamente. Las primeras pruebas fueron impresas con ese material y no funcionaron bien, en particular, el rotor. Por eso rediseñé el rotor y lo imprimí en PETG, que resiste bien hasta 85°C. El resto de las piezas no fueron reimprimadas en ese material. Aparentemente, funcionan bien en PLA. -temperatura de la comida: cocinamos la comida en una olla a baño maría con termostato fijo a 95°C. A esa temperatura, el PLA se ablanda. El rotor es particularmente sensible a este efecto porque se deforma el orificio donde encaja el motor. Por eso, lo reimprimamos con PETG y además bajamos la temperatura a 70°C para distribuirla. Es decir, cocinamos a 95 y luego bajamos a 70°C. De esa manera, logramos distribuir 4 litros de comida sin ningún inconveniente. Por otra parte, es importante que se mantenga la temperatura de la comida ya que si se calibra a una temperatura para que impulse un cierto volumen de comida en cada ciclo y luego la comida se enfría, el volumen dispensado va a ser menor. La solución es recalibrar (es muy fácil de hacer) cuando sea necesario o mantener la comida a una temperatura fija.

El modelo inicial del rotor fue copiado de <https://www.thingiverse.com/thing:3148717> pero luego modificado y es el que se adjunta. Recomienda imprimirlo con infill 100%. Yo recomiendo usar PETG, al menos para el rotor.

Mi idea era hacer una bomba que pudiera usarse por dos laboratorios diferentes, el mío, que trabaja con moscas y otro de gusanos (por eso tiene dos programas denominados “moscas” y “gusanos”. Se puede cambiar el nombre en las líneas 161 y 162 del código). En este último caso, se necesita distribuir un medio de cultivo estéril y por lo tanto, necesitan esterilizar la manguera luego de cada uso. Para que fuera sencillo cambiar la manguera luego de cada uso diseñé las piezas que adjunto. Pero luego no funcionaron tal como esperaba, ya que para evitar que el motor se trabe es necesario que las piezas tengan algo de juego, que puedan moverse levemente. La solución que encontré fue hacerles un agujero con un taladro y atravesar un tornillo para mantener unidas las dos piezas entre sí y dejar que puedan moverse. Los orificios ya están incluidos en el modelo adjunto. Este diseño se parece bastante en su funcionamiento a aquel del que copié el rotor. La manguera puede aun cambiarse con relativa facilidad. Si sólo vas a ser usado para comida de moscas, quizás pueda ser más práctico imprimir directamente el modelo completo del que saqué el rotor, que también incluye una carcasa para alojar el motor y el controlador. El cambio de la manguera parece más trabajoso en ese modelo, pero eso no sería un inconveniente si no es necesario esterilizarla luego de cada uso ya que puede lavarse el interior pasando agua caliente. Acerca del circuito eléctrico. Hay tres cosas que son opcionales. Una es la activación con un pedal. La idea era llenar los tubos de a uno activando la bomba con un pedal, pero encontramos que la alternativa de generar un programa con un número de ciclos (=tubos) y una pausa entre ellos era mejor. La otra característica opcional son los leds que indican el modo de funcionamiento. La última es el ventilador para enfriar el driver del motor y el motor mismo. Las especificaciones del driver indican que puede llegar a ser necesario refrigerarlo. Como es barato, preferí agregarlo, pero me parece que funcionaría igual de bien sin él.

Lista de materials con links a proveedores en Argentina

- 1- motor paso a paso (stepper motor) 1.5A Alto Torque 4.2kg 17HS4401S x1 <https://articulo.mercadolibre.com.ar/MLA-899670687-nema-17-motor-paso-a-paso-18-alto-t-15a-usongshine- JM>
- 2- ruleman 605zz x9 <https://articulo.mercadolibre.com.ar/MLA-915854069-rodamiento-ruleman-605-zz-x20-unidades- JM>
- 3- driver A4899 x1 <https://www.todomicro.com.ar/motores-y-drivers-motores/288-modulo-a4988-pololu-driver-motor-paso-a-paso.html>
- 4- arduino promini 5V (se puede usar también de 3.3V o un arduino UNO) x1
- 5- Fuente 12V 2A x1
- 6- Regulador step down MP1584 x1
- 7- Display LCD 16x2 con adaptador I2C x1
- 8- Encoder Rotativo Ky-040 x1
- 9- LEDs 3mm (opcional) x3
- 10- condensador de 100uF x1
- 11- resistencias 220 Ohm, 10kOhm
- 12- ventilador de 6cm (opcional) x1
- 13- caja plástica: se puede imprimir. Usé una caja estanca de aprox 20x30 cm
- 14- la manguera es de silicona diámetro ext 10mm, diámetro int 6mm (paredes 2mm).
- 15- tornillos, tuercas...

AJUSTES de corriente máxima para el driver del motor

Este punto es muy importante. Hay que seguir las instrucciones que se encuentran aquí: <https://www.pololu.com/product/1182> El módulo que conseguí no es el original y no tiene las resistencias que se indican en ese instructivo. De todas maneras, el método de ajuste es el mismo. A continuación, pego mis notas respecto de esto:

Motor 17HS4401S según el que lo vendió la intensidad máxima es 1.5A

para ajustar el A4988 $I_{max} = V_{ref}/8 * \text{resistencia limitadora}$

I_{max} es 1.5 la resistencia es R100 0.1Ohm o 100 mOhm

$V_{ref} = 1.2V$

Uso el driver en modo quarter steps con un nivel lógico de 5V.

Ajustando el potenciómetro del módulo al máximo, el máximo valor V_{ref} alcanzado es de 1.19V, que es el V_{ref} con el que se alcanza el I_{max} de 1.5 A es decir, que quedó en el límite superior.

Funcionamiento

Al encenderla, aparece en el display un menú con 11 opciones. Girando el control en sentido horario o antihorario se cambia de menú. Presionando la misma perilla, se entra en el menú y pueden seleccionarse opciones. Presionando nuevamente, se sale del menú. En cualquier momento, si se mantiene presionado el selector comenzará a funcionar la bomba a una velocidad fija (no necesariamente la misma que la seleccionada en el menú 3) en forma continua hasta que se suelte. Eso permite llenar la manguera.

Menú 0: Start--> inicia el programa (dosifica/bombea/calibra). Presionando nuevamente, interrumpe la ejecución al finalizar el ciclo actual.

Menú 1: Tiempo --> permite establecer el tiempo de llenado deseado. Calcula automáticamente las RPM. Si RPM es mayor que 200, ajusta el tiempo para no sobrepasar 200.

Menú 2: Velocidad --> permite ajustar RPM, calcula automáticamente el tiempo de llenado.

Menú 3: Ciclos --> número de ciclos a repetir al presionar start. (es decir, número de tubos a llenar). Máximo 200. Se puede aumentar en la línea 139 del programa.

Menú 4: Pausa --> tiempo de espera entre dos ciclos. Permite posicionar la manguera sobre un nuevo tubo vacío

Menú 5: Modo --> Tiene tres opciones. Dosifica (ejecuta un ciclo durante el que gira el número de veces seleccionado durante la calibración y luego frena. Hace una pausa y vuelve a arrancar hasta completar el número de ciclos indicados en el menú 4). Bombea (el motor gira a la velocidad seleccionada en menú 3) y Calibra (ejecuta un ciclo de calibración y luego vuelve automáticamente a modo Dosifica)

Menú 6: PGM --> permite almacenar dos volúmenes diferentes para dispensar comida

Menú 7: Sentido --> sentido de giro

Menú 8: M--> es el número de pasos que da el motor en el programa moscas (200 pasos es una vuelta).

Menú 9: G--> es el número de pasos que da el motor en el programa gusanos (200 pasos es una vuelta).

Menú 10: Guarda configuración en memoria: Al iniciar el programa carga valores por defecto en la memoria. Luego de calibrar (o modificar la velocidad, ciclos, pausa) se debe ejecutar esta opción del menú para que se guarden en la memoria. En caso contrario, la vez siguiente que se encienda el equipo volverá a los valores por defecto.

Inicio

Ajustar la posición del cubrerotor para que los rulemanes ajusten perfectamente bien la manguera, pero que no bloqueen el motor. La manguera está bien ajustada cuando al soplar por un extremo no es posible hacer circular aire. (<https://youtu.be/28UxOIC15p4>)

Calibración

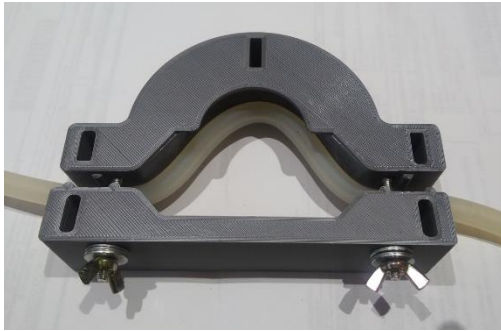
Seleccionar en el menú 7 un programa para calibrar (moscas o gusanos. O tubo grande/chico). En menú 6 seleccionar la opción calibrar. Aparece un mensaje que dice: “Presionar hasta llenar el tubo” Posicionar la manguera sobre un tubo vacío y mantener presionado el botón. El motor comenzará a girar lentamente. Soltar el botón cuando se alcance el nivel de comida deseado. El programa almacena el número de vueltas dado y calcula cuánto tiempo demorará en llenar el tubo (nuevo valor del menú 2) en función de la velocidad indicada en el menú 3. Luego muestra el número de pasos y el número de vueltas del motor. Al finalizar la rutina pasa al modo dosifica automáticamente.

Dosificación

Girar la perilla hasta llegar al menú 1 y presionar el botón. Aparece la leyenda RUN. Según el modo, bombea, dosifica o calibra. Si está en modo “Dosifica”, arranca el motor, cumple un ciclo y hace una pausa (de duración establecida en menú 5). Repite el proceso tantas veces como se haya indicado en el menú 4. La ejecución de la serie de ciclos se puede interrumpir presionando nuevamente la perilla. El efecto no es inmediato, primero tiene que terminar de ejecutar el ciclo actual y luego frena.



Vista superior, inferior y lateral del rotor



Cubierta de la manguera

