



**universidad
de león**

Escuela de Ingeniería de Industrial e Informática

Asignatura de Arquitecturas Específicas y Empotradas

Memoria de proyecto de bar automatizado

Realizado por los alumnos:

Andrés García Álvarez

Pablo González de

Eduardo Juárez Robles.

Santiago Peláez Peláez.

Índice

Índice de ilustraciones	3
1. Resumen de la idea del proyecto y cómo surge.	4
2. Objetivos iniciales del proyecto.	4
3. Organización del grupo y las tareas.	4
4. Cronograma del proyecto.	6
5. Descripción de los materiales utilizados.	7
6. Descripción de las herramientas de Software utilizadas.	11
7.1 Construcción de la maqueta:	11
7.2 Diseño de circuitos:	11
Etapa 1:	12
Etapa 2:	12
7. Desarrollo y montaje.	13
8.1 Investigación previa.	13
8.2 Montaje de la maqueta.	13
8.3 Trabajando con los motores (shield L293D)	13
8.4 Módulo bluetooth.	14
8.5 Control de motores (RAMPS).	14
7.6 Implementación de las bombas impulsoras de agua.	15
8. Resultados obtenidos.	15
9. Presupuesto.	17
a. Componentes de nuestro proyecto final.	17
b. Componentes usados anteriormente	20
10. Posibles mejoras o evoluciones.	21
11. Bibliografía	22
Anexo I: Planos	23
Anexo II: Piezas impresas.	25
Anexo III: Esquemas eléctricos	29

Índice de ilustraciones

<i>Ilustración 1: Cronograma.....</i>	<i>6</i>
<i>Ilustración 2: Diagrama de Gant.....</i>	<i>7</i>
<i>Ilustración 3: Arduino mega 2560.....</i>	<i>7</i>
<i>Ilustración 4: Ramps 1.4.....</i>	<i>8</i>
<i>Ilustración 5: Controlador de motor paso a paso</i>	<i>8</i>
<i>Ilustración 6: Fuente de alimentación.....</i>	<i>8</i>
<i>Ilustración 7: Dispensador de bebida.....</i>	<i>9</i>
<i>Ilustración 8: Motor paso a paso nema 17.....</i>	<i>9</i>
<i>Ilustración 9: Cable alargador de 4 polos.....</i>	<i>10</i>
<i>Ilustración 10: Bomba de líquidos.....</i>	<i>10</i>
<i>Ilustración 11: Polea GT2 20 dientes 6mm</i>	<i>11</i>
<i>Ilustración 12: Correa de polea GT2 6mm</i>	<i>11</i>

1. Resumen de la idea del proyecto y cómo surge.

El proyecto consiste, en la creación de un prototipo de bar automatizado, es decir, que sirva diferentes mezclas de bebidas de forma automática.

Para explicarlo forma básica el proyecto consiste en unos raíles sobre los cuales, un carrito de forma automática se desplaza, mientras puja con un baso, de manera simultanea el carrito tiene un dispositivo que interactúa con unos dispensadores, los cuales van rellenando el vaso con la mezcla deseada por el cliente, una vez el vaso contenga la mezcla deseada volverá al punto de partida para que el usuario pueda recogerlo.

El proyecto consta de tres partes bien diferenciadas:

1. El carrito con el sistema de raíles.
2. Dispensadores de bebidas.
3. Bombas dispensadoras de bebidas.

La idea de nuestro proyecto surge en una de las clases de practicas, después de estar investigando, y buscando un proyecto interesante el cual pudiese proporcionarnos una buena experiencia durante las clases de esta asignatura, después de ver algunos ejemplos hemos decidido basarnos en un prototipo que nos ha gustado y que se denomina bartender.

2. Objetivos iniciales del proyecto.

Tras la elección de nuestro dispensador de copas podemos decir que con él buscamos la automatización de una tarea cotidiana, que fuese visualmente atractiva, que estuviese al nivel de unos alumnos de cuarto de carrera y que nos permitiese aplicar parte de las habilidades obtenidas hasta la fecha en estos cuatro años de formación.

Por otra parte, desde el punto de vista del proyecto pretendíamos poder realizar un proyecto completamente funcional y que ampliase los conocimientos de Arduino obtenidos en las primeras sesiones de la asignatura y que su realización fuese posible en el tiempo estimado de tres meses.

3. Organización del grupo y las tareas.

Las tareas del proyecto han recaído de igual forma entre los diversos integrantes del grupo, a lo largo de la realización del proyecto nos hemos reunido para poder avanzar en nuestro proyecto, con gran frecuencia, acudiendo a todas la practicas, y en tiempos libres que hemos intentado aprovechar al máximo el tiempo posible, cabe destacar que hemos trabajado fuera de la universidad, llegando a acudir a la casa de uno de nuestros compañeros para poder avanzar, desde el diseño que queríamos tener, como en los diversos momentos en los que había que realizar una parte de construcción de nuestro proyecto.

Según íbamos avanzando en el desarrollo del proyecto, las fases que llevábamos acabo iban evolucionando con nosotros, junto con reparto de tareas que ha sido, o ha pretendido ser lo mas equitativa posible, siendo la asignación de tareas una parte secundaria de la organización ya que todos nos ayudábamos. De la misma manera las fases si han estado bien definidas durante todo el proyecto.

Podemos definir las siguientes fases del proyecto:

Planificación: la fase mas temprana de todas, en esta fase podemos encontrar, las tareas en las cuales elegiremos el proyecto y plantearemos como será llevado a cabo. Las tareas que podemos distinguir en esta fase son:

elegir el proyecto: puede ser la tarea mas obvia, pero tuvo que llevarse a cabo varias reuniones hasta que los miembros del grupo estuvieron satisfechos con el proyecto.

Planos y materiales: en esta tarea nosotros junto con la ayuda del profesor realizamos los esbozos principales del proyecto y basándonos en estos haríamos una lista de materiales.

Pedir materiales: usando la lista de materiales obtenida, se pedirían, para poder empezar a usar el proyecto

Desarrollo: la fase de desarrollo, es sin duda la fase mas extensa, es en la fase en la que el proyecto adquiere su forma y avanza considerablemente, también es en la fase en la que nos surgieron mas problemas. Dentro de esta fase podemos encontrar las siguientes tareas:

Mejora de los planos: detallar mas los planos antiguos para empezar el proyecto

Construcción de la estructura: el objetivo de esta tarea es montar una estructura sobre la cual se va a hacer el resto del proyecto, esta tarea esta compuesta por diferentes partes, las cuales serian, la selección de materiales, calculo de medidas, fabricación de las piezas y ensamblaje de la estructura.

Montaje de los motores: colocación de los motores que van a ser usados por el proyecto, en las piezas adecuadas, para que el proyecto funcione adecuadamente.

Implementación de la electrónica y componentes electrónicos: consisten en colocar cada componente electrónico donde debe para que el proyecto funcione de manera optima sin que de fallos.

Solución de problemas: esta fue una tarea imprevista que no se pudo superar, debido a problemas con algunos componentes del proyecto, en este caso el mayor problema fue con los drivers de los motores, y después de haber intentado solucionar los problemas de diferentes maneras, como puede ser aumentando la refrigeración, o sustituyéndolos por otros, no se pudo llegar a nada.

Cambio a ramps: debido a los problemas que nos estaba dando el shield como ultimo recurso se decidió cambiar a otro mas adecuado.

Periodo de adaptación con el nuevo shield: solucionando diferentes tipos problemas del shield adaptándonos y al nuevo tipo de código, en resumen, una fase de aprendizaje forzosa.

Implementación de las bombas: se añaden los componentes que faltaban al proyecto.

fase de pruebas: se realizan las pruebas necesarias para saber si el proyecto funciona adecuadamente o si este produce algún fallo, de ser así será corregido también en esta fase.

Maquetación: en esta fase se pretendía darle al proyecto un acabado decente para ser expuesto de cara al público.

4. Cronograma del proyecto


















		Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1			Elección de proyecto	4 días	mié 10/10/18	lun 15/10/18	
2			Diseño de planos	4 días	mar 16/10/18	vie 19/10/18	1
3			Selección de materiales	1 día	lun 22/10/18	lun 22/10/18	2
4			Reparto de tareas	1 día	mar 23/10/18	mar 23/10/18	3
5			Construcción de la maqueta	6 días	mié 24/10/18	mié 31/10/18	4
6			Montaje de motores	1 día	jue 01/11/18	jue 01/11/18	5
7			Configuración de la electrónica	11 días	vie 02/11/18	vie 16/11/18	6
8			Implementación Bluetooth	2 días	lun 19/11/18	mar 20/11/18	7
9			Mejora disipación	11 días	lun 19/11/18	lun 03/12/18	7
10			Solución de problemas	37 días	lun 19/11/18	mar 08/01/19	7
11			Cambio RAMPS	3 días	mié 09/01/19	vie 11/01/19	10
12			Solución de problemas RAMPS	5 días	lun 14/01/19	vie 18/01/19	11
13			Implementación bombas	4 días	lun 14/01/19	jue 17/01/19	10
14			Realización de la memoria	5 días	lun 14/01/19	vie 18/01/19	10
15			Entrega	0 días	vie 18/01/19	vie 18/01/19	
16			Presentación	0 días	lun 21/01/19	lun 21/01/19	

Ilustración 1: Cronograma

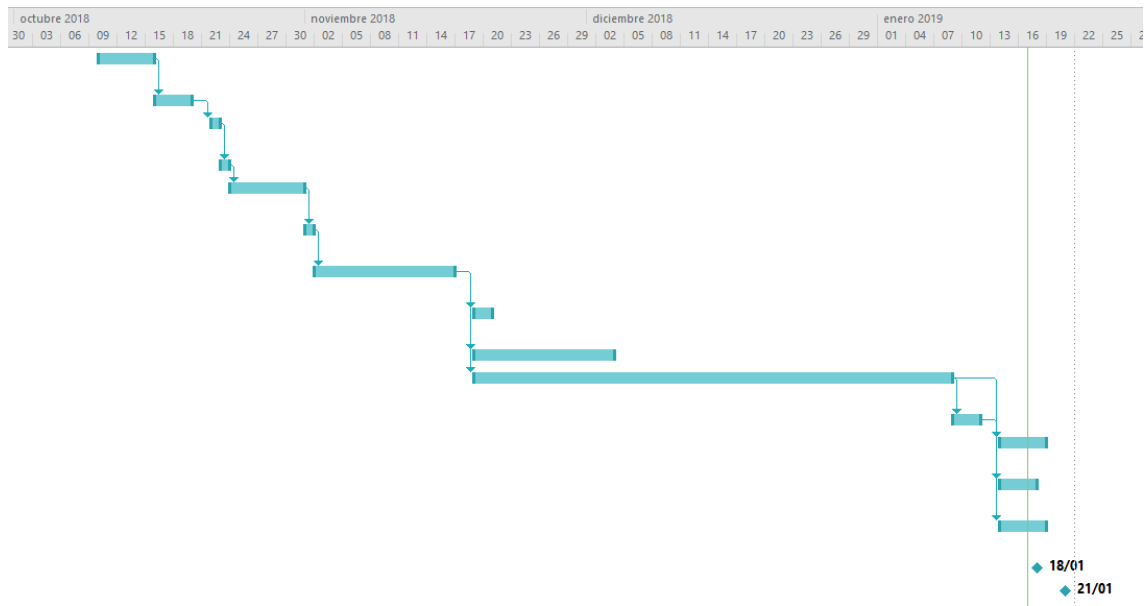


Ilustración 2: Diagrama de Gantt

5. Descripción de los materiales utilizados.

Los materiales utilizados, intentare mostrar todos aquellos que nuestro proyecto final utiliza.



Ilustración 3: Arduino mega 2560

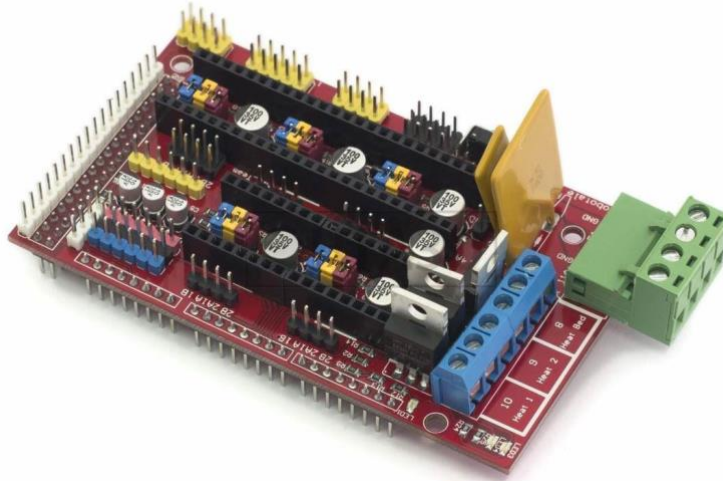


Ilustración 4: Ramps 1.4

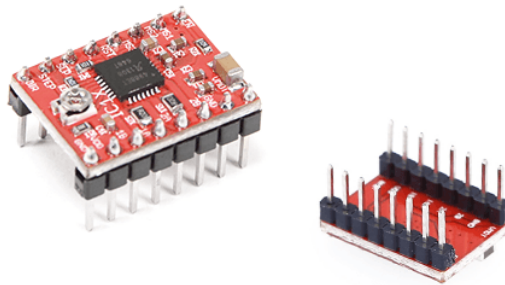


Ilustración 5: Controlador de motor paso a paso



Ilustración 6: Fuente de alimentación



Ilustración 7: Dispensador de bebida



Ilustración 8: Motor paso a paso nema 17

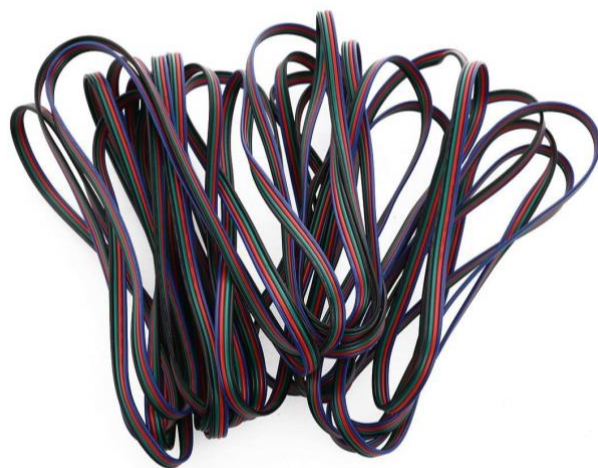


Ilustración 9: Cable alargador de 4 polos



Ilustración 10: Bomba de líquidos



Ilustración 11: Polea GT2 20 dientes 6mm



Ilustración 12: Correa de polea GT2 6mm

6. Descripción de las herramientas de Software utilizadas.

Para llevar a cabo la realización del proyecto, utilizamos el software de Arduino como framework de desarrollo.

Vamos a proceder a distinguir entre la fase en las que utilizábamos los drivers de motor L293D y cuando usábamos A4988.

7.1 Construcción de la maqueta:

Para la impresión de las piezas diseñadas en 3D utilizamos un software gratuito llamado Repetier. Esta herramienta permite colocar y dimensionar comunicar con la impresora.

Repetier se vale de Slic3r para laminar en capas los modelos tridimensionales y definir los movimientos que tienen que hacer los motores para poder depositar el material en el lugar adecuado. Este software es el que realmente genera los movimientos que tiene que realizar la impresora para cumplir su función.

7.2 Diseño de circuitos:

Para hacernos una idea aproximada de los componentes que íbamos a necesitar para manejar nuestros actuadores recurrimos a la herramienta circuito.io.

Etapa 1:

Cuando utilizamos los drivers L293D, programábamos para Arduino utilizando las llamadas típicas, leíamos información por puerto serie, accionábamos salidas digitales con la función `digitalwrite`, etc. El desarrollo de este tipo de software es muy sencillo y en apenas un par de días hicimos todo nuestro programa a partir de el código fuente obtenido de nuestra fuente de inspiración. Los motores se controlan con la librería `AF_Stepper.h` la cual permite interactuar con los mismos por medio de pasos (pasos positivos hace que se mueva el motor en una dirección y pasos negativos hacen que se mueva en la dirección opuesta).

Etapa 2:

Esta segunda etapa abarca desde el momento en que decidimos migrar nuestro sistema a RAMPS 1.4. En este caso también se utiliza el framework de Arduino. El software para el control de los motores que tomamos como base es Marlin. Este software es específico para impresoras 3D y de ahí vinieron todos los problemas.

Para hacer una función que enviase los movimientos de los motores tuvimos que copiar el buffer del puerto serie el string que contenía el gcode.

La adaptación para las bombas fue mucho más complicada ya que la función `digitalwrite` este software la ignora. Por ello tuvimos que tomar la decisión de usar unas salidas desde las cuales suministra 12V. La salida D09 que es la única que utilizamos finalmente, para arrancar la bomba solo tenemos que fijar la velocidad en 255 para que nos suministre los 12V y fijar la velocidad a 0 para que par

7. Desarrollo y montaje.

8.1 Investigación previa.

Comenzamos por la investigación de material de software suficiente para tomar como base para la realización del proyecto. Esta primera aproximación al código nos sirvió para saber como se manejaban los motores con el driver L293D.

8.2 Montaje de la maqueta.

Con los planos obtenidos de nuestra fuente y las posteriores modificaciones realizadas para adaptarlo a nuestras necesidades procedimos al montaje de la maqueta.

Los materiales principales utilizamos fueron madera como base estructural y piezas impresas en 3D para dar forma a anclajes y piezas móviles varias.

Las herramientas necesarias para el tratamiento de la madera fueron: sierra circular, taladro, atornillador, pistola de pegamento termofusible, etc.

Las piezas impresas en 3D hubo que reescalarlas y aplicarles un tratamiento con vaselina para reducir la fricción entre ellas y los movimientos fuesen más fluidos.

Colocamos los tres motores y el pulsador de fin de carrera en sus respectivas posiciones y procedimos a cablearlo todo para poder comenzar a trabajar con la electrónica y la preparación del software.

8.3 Trabajando con los motores (shield L293D)

Al comenzar a trabajar con los motores paso a paso no sabíamos como teníamos que colocar las bobinas y tuvimos que comprobar la continuidad en cada uno de los pines, una vez aprendido esto ya resultó más fácil trabajar con los motores. Cuando conseguimos hacerlos funcionar con la librería AF_Stepper.h, nos dimos cuenta que tras pasar cierto tiempo los motores comenzaban a perder pasos, esto era por el sobrecalentamiento de los drivers L293D que manejaban los motores. Comenzamos a buscar soluciones y lo primero que hicimos fue colocar unos pequeños disipadores, esto no resultó efectivo. El responsable de la asignatura nos sugirió buscar unos disipadores más grandes procedentes de una fuente de alimentación y tras intentar el montaje fallido tuvimos una equivocación e invertimos la polaridad, esto provocó una destrucción completa de la electrónica que posteriormente tuvimos que sustituir.

Los siguientes componentes en teoría eran idénticos a los iniciales, pero se comportaban de una manera más errática todavía, haciendo las mismas acciones en intervalos de tiempo suficientemente espaciados como para que los componentes trabajasen siempre a la misma temperatura, en ocasiones funcionaban correctamente y en ocasiones no. Debido a este comportamiento tan extraño, nos pusimos a investigar y descubrimos que los drivers L293D daban una intensidad de corriente de 600mA constantes y 1.2A en picos. Tras este hallazgo decidimos proceder a hacer pequeñas pausas durante la realización de una tarea por de los motores para así

desestresar los componentes. Tras el paso cierto tiempo los componentes, tanto el shield como el Arduino, dejaron de funcionar por completo.

Como estas acciones no resultaron, procedimos a colocar una disipación “extrema” (el disipador de una CPU de PC) que al principio funcionaba, pero tras largos movimientos continuados en el tiempo volvía a funcionar de manera errática. Probamos también a cambiar la como estaban funcionando los motores para ver si de este modo consumían menos,

8.4 Módulo bluetooth.

La implementación del módulo de bluetooth HC-05 fue bastante sencilla mientras trabajábamos con el shield L293D. Este módulo consta de 6 pines de los cuales nosotros utilizamos 4: VCC (3.3V), GND, RXD y TXD. Este dispositivo lo utilizamos de manera que las instrucciones en lugar de enviarlas por el puerto serie que está conectado al ordenador utilizamos una aplicación que utiliza la capacidad bluetooth para comunicar con el Arduino por los puertos RXD y TXD. Por este motivo no fue necesario desarrollar nuestro propio software de Android.

8.5 Control de motores (RAMPS).

Puesto que los motores paso a paso no acababan de responder adecuadamente, tomamos la decisión de migrar nuestro sistema a una plataforma con unas especificaciones más adecuadas a nuestro proyecto, esto es Arduino Mega con shield RAMPS 1.4 y drivers de motor A4988. Estos drivers ofrecen una corriente de salida de 2A y está comprobado que funcionan adecuadamente con este tipo de motores. El software que incorporamos a la placa en esta ocasión ha sido Marlin, que es un software específico para impresoras 3D.

Aquí el reto se encuentra en la adaptación del código para que ejecute desde dentro los gcode en lugar de recibirlos por puerto serie. Otro problema es descubrir que pines auxiliares se utilizan debido a que el código tiene definidos pines para gran número de placas, lo cual dificulta enormemente la lectura del mismo.

Conseguido salvar el escollo de la adaptación del código comprobamos que los motores ahora se movían de manera mucho más silenciosa, sutil y precisa. De todos modos, hubo que adaptar el potenciómetro regulador de potencia del chip A4988 para conseguir el resultado óptimo.

A pesar de que RAMPS y los drivers A4988 ya eran adecuados para nuestras especificaciones, en esta ocasión también dejaron de funcionar adecuadamente los drivers, en concreto pasaron a realizar movimientos en una sola dirección. Es decir, cuando le mandábamos moverse de por ejemplo la posición 0 a la 200 y de nuevo a la 0, el dispositivo realizaba los movimientos de 0 a -200 y a continuación a -400. Esto es imposible que lo realice un driver en perfecto estado ya que hay que calibrar con fines de carrera los movimientos y donde está el fin de carrera se toma la coordenada 0 y a partir de ahí solo tenemos referencias mayores que 0.

7.6 Implementación de las bombas impulsoras de agua.

La idea inicial para llevar a cabo la implementación del código referente a las bombas era conectándolas por medio de un relé a la fuente de alimentación y activarlas mediante la instrucción `digitalwrite(pin, HIGH|LOW)`. El problema es que Marlin no obedece las instrucciones normales de Arduino, las ignora, por ello hemos tenido que buscar alternativas dentro de las posibilidades que ofrece este software de control de impresoras 3D.

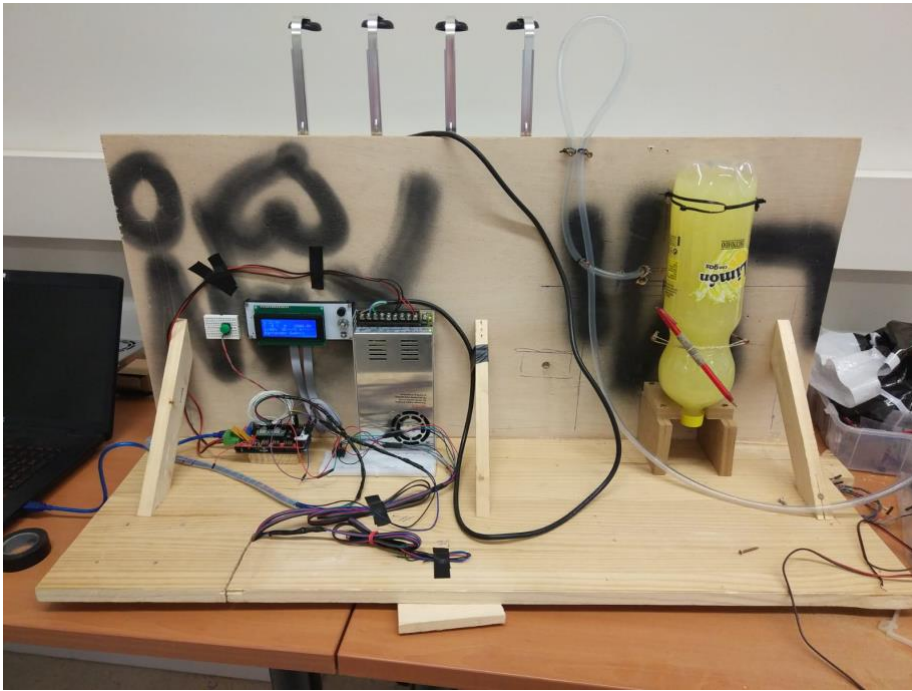
Las salidas D10, D09, D08 están implementadas de tal manera que pasan directamente voltaje en el momento que reciben una instrucción determinada.

La más fácil de implementar ha sido la salida D09 que en principio está destinada a un ventilador, lo que nosotros hacemos para activar la bomba es fijar ese ventilador a máximas revoluciones para que este nos de 12V y que la bomba trabaje en condiciones óptimas.

Las otras dos salidas D10 y D08 hacen funcionar unas resistencias cuando les marcamos un objetivo de temperatura por lo que es mucho más difícil “engañar” al software.

8. Resultados obtenidos

El resultado final está compuesto por una estructura de madera que sujeta a los demás componentes. Estos son, por la parte trasera: Un Arduino mega 2560, un shield RAMPS 1.4, una fuente de alimentación, una pantalla de información y todo el conjunto de cables necesarios convenientemente colocados y agrupados por componentes.



Por la parte frontal tendríamos el carro impreso en 3D con los motores y la plataforma con el vaso, sobre los raíles que le permiten moverse entre botellas y bombas.



Mediante un discreto cable USB, se conecta el sistema con un ordenador portátil mediante el que se le pueden pasar los comandos con las instrucciones para servir las copas. Se espera que en un futuro próximo se le puedan enviar las instrucciones mediante una aplicación móvil que se conectar al sistema por bluetooth, ha día de hoy existe un conflicto entre como enviamos las ordenes de movimiento desde dentro de Marlin y como se envían las solicitudes de bebida a través del puerto serie.

El sistema resultante permite servir una mezcla seleccionada entre tres tipos de bebidas alcohólicas y dos tipos de mezclas, una con gas y otra sin gas.

Antes de empezar a funcionar con la máquina necesitaremos calibrar los ejes x e y para tener las coordenadas bien definidas. Posteriormente ya podremos preparar las bebidas.

Solución de problemas:

En caso de que los motores actúen de manera extraña o incluso que nos se muevan y si metan ruido debemos proceder a comprobar que todo el cableado de los motores está adecuadamente conectado.

En caso de que no actúe cuando mandamos instrucciones debemos comprobar primero que el cable USB esté bien conectado, en segundo lugar, usar la pantalla para mover los motores y así comprobar que los drivers están en buen estado y en tercer lugar proceder a hacer un reset.

9. Presupuesto.

El presupuesto de nuestro proyecto podríamos dividirlo en dos partes:

a. Componentes de nuestro proyecto final

Ítems	Descripción	Unidades	Precio unitario	Total
0	Arduino mega 256	1	8,77	8,77
1	Ramps 1.4	1	10,99	10,99
2	Controlador motor paso a paso	2	1,99	3,98
3	Fuente alimentación	1	16,99	16,99
4	Dispensador de bebidas	1	26,38	26,38
5	Motores paso a paso	3	13,99	41,97

6	Cable de 4 polos	5	0,99	4,95
7	Manguera de 8mm	4	0,3	1,2
8	Bombas de agua	2	9,99	19,98
9	Polea del motor	1	1,84	1,84
10	Correa dentada motor	2	1,35	2.7

<https://solectroshop.com/product-spa-897-Modulo-Mega2560-R3-USB-CH340.html>

https://www.amazon.es/ARCELI-Controlador-Impresora-Shield-Arduino/dp/B07BSRS9WS/ref=sr_1_2?ie=UTF8&qid=1547738332&sr=8-2&keywords=ramps

https://www.amazon.es/ELEGOO-Controlador-Ramps1-4-Disipador-Impresora/dp/B06ZZH81WT/ref=sr_1_17?ie=UTF8&qid=1547738415&sr=8-17&keywords=ramps+controlador

https://www.amazon.es/GHB-Transformador-Potencia-Voltaje-Alimentación/dp/B01421FJJO/ref=sr_1_2?ie=UTF8&qid=1547738591&sr=8-2&keywords=fuente+de+alimentacion+12v

https://www.banggood.com/es/Beverage-Liquor-Dispenser-4-Bottle-Bar-Alcohol-Holder-Cocktail-Drink-Shot-Bracket-Wall-Mounted-p-1364361.html?gmcCountry=ES¤cy=EUR&createTmp=1&utm_source=googleshopping&utm_medium=cpc_union&utm_content=2zou&utm_campaign=ssc-es-all&gclid=EAlaIqobChMI_frF_4_13wIVghbTCh29KAFHEAQYAyABEgJ9fvD_BwE&cur_warehouse=CN

https://www.amazon.es/Motor-Pasos-Fases-Cables-Impresora/dp/B06ZY9G8KG/ref=sr_1_1?ie=UTF8&qid=1547738902&sr=8-1&keywords=motores+paso+a+paso+12v

https://www.amazon.es/UEETEK4-Color-cable-extensión-alambre/dp/B019GUK5JU/ref=sr_1_1?ie=UTF8&qid=1547739009&sr=8-1&keywords=cable+4+hilos

https://www.amazon.es/dp/B01FTL6NUU/ref=cm_sw_r_cp_api_eJQ.Bb01G3J7T

https://www.amazon.es/canal-Relay-Módulo-Arduino-Raspberry/dp/B01N1K2WW8/ref=sr_1_17?ie=UTF8&qid=1547739704&sr=8-17&keywords=rele+12v+arduino

<https://solectroshop.com/product-spa-599-2x-Polea-GT2-20-Dientes-6mm-Aluminio-Pulley-Impresora-3D-Reprap.html>

<https://solectroshop.com/product-spa-788-Correa-GT2-2m-6mm-Impresora-3D.html?rec=101800103>

b. Componentes usados anteriormente

Una segunda parte que seria la formada por elementos que, aún no estando presentes en nuestro proyecto actual, si que han formado parte del desarrollo, sin obtener los resultados deseados:

Ítems	Descripción	Unidades	Precio unitario	Total
1	Arduino mega 256	1	8,77	8,77
2	Ramps 1.4	1	10,99	10,99
3	Fuente alimentación	1	16,99	16,99
4	Dispensador de bebidas	1	26,38	26,38
5	Motores paso a paso	3	13,99	41,97
6	Cable de 4 polos	5	0,99	4,95
7	Manguera de 8mm	4	0,3	1,2
8	Bombas de agua	2	9,99	19,98
9	Relés de 12v	2	6,37	12,74
10	Arduino uno	3	6,07	18,21
11	Shield L293R	3	2,35	7,05
12	Ventilador 12v	2	2,44	4,88

<https://solectroshop.com/product-spa-481-UNO-R3-ATmega328-ATmega16U2.html>

<https://solectroshop.com/product-spa-146-L293D-Tablero-de-Expansion-Mega-Controlador-Servo.html>

<https://solectroshop.com/product-spa-343-Ventilador-12V-2x-Cables-Funda-CPU-PC.html>

10. Posibles mejoras o evoluciones.

El enfoque que le hemos querido dar al proyecto siempre ha sido comercial, pretendiendo conseguir un producto de alta gama el cual se pueda usar en diferentes establecimientos como bares y restaurantes, en incluso en algunas casas.

Las mejoras en las que mas interesados implementar para conseguir el producto final deseado, serian las siguientes ordenadas de mayor importancia a menor:

1. Conectividad bluetooth: esta mejora pretende establecer una conectividad entre el proyecto y un dispositivo, para facilitar el uso por parte del usuario.
2. Interfaz grafica: al igual que la mejora anterior esta pretende, facilitar al usuario a la hora de manejar el dispositivo ya que en el proyecto actual funciona mediante comandos.
3. Aumento de bebidas. Esta mejora es meramente funcional, ya que a mas bebidas mas combinaciones, al mayor numero de combinaciones el proyecto alcanza un mayor interés para el usuario.
4. Reducción de tamaño: hacer un tamaño mas compacto, es casi un requisito indispensable para manipular el dispositivo. En el tamaño actual manipular el dispositivo en el sentido de desplazarlo de un lugar a otro, se vuelve una tarea tediosa.
5. Mejorar el acabado: ya que nuestra intención siempre ha sido la creación de un prototipo funcional descuidando en gran parte la estética, en un producto profesional este dispositivo tendría un acabado mas estético y atrayente
6. Mejora de componentes: a medida que avancemos en el proyecto y los componentes se vayan quedando obsoletos, se irán sustituyendo por elementos mas punteros.
7. Dispensador de hielos

11. Bibliografía

<http://diymachines.co.uk/?p=10>

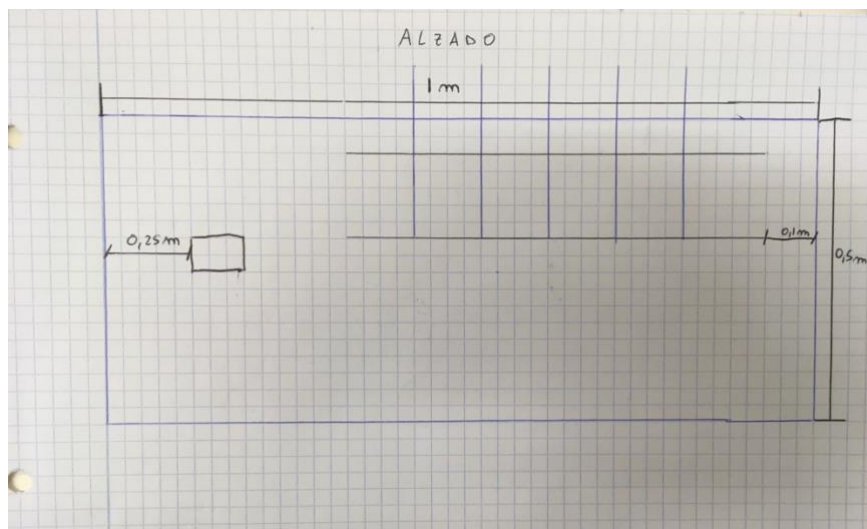
<http://diymachines.co.uk/?p=34>

<http://diymachines.co.uk/?p=57>

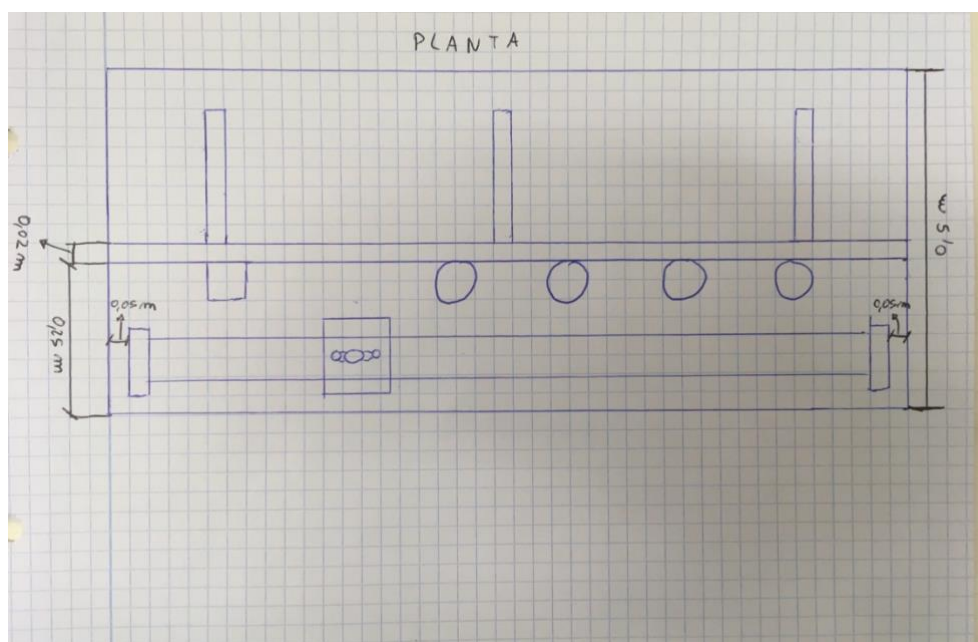
<http://diymachines.co.uk/?p=62>

Anexo I

Planos



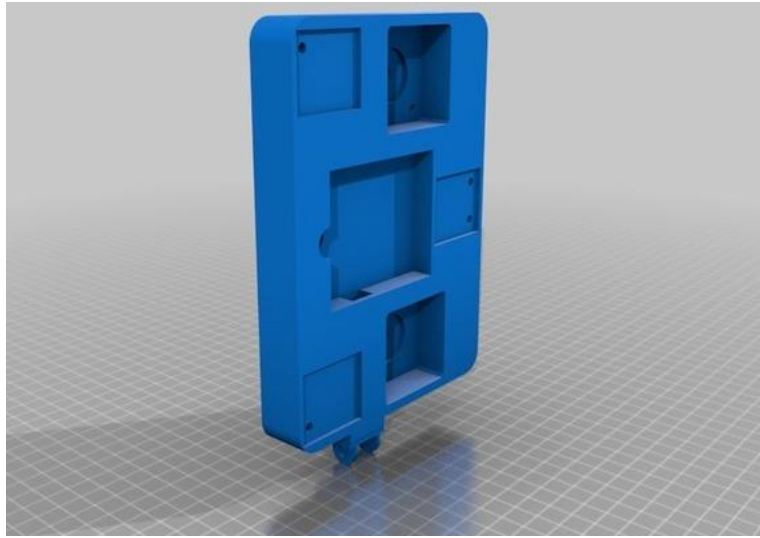
Plano 1 Vista Alzado



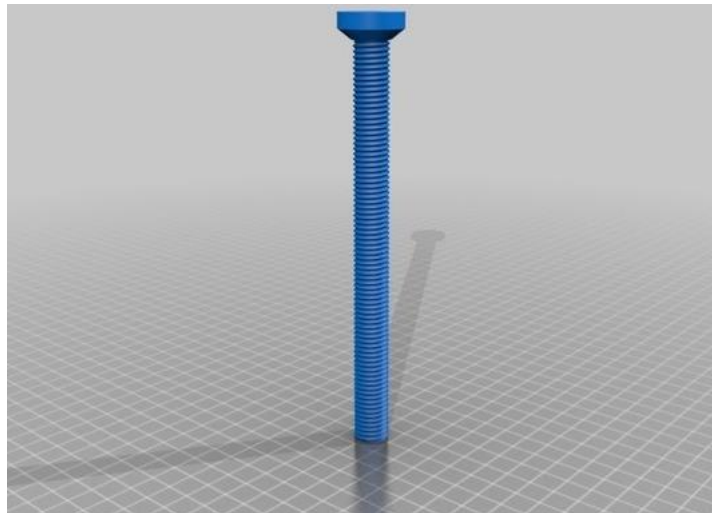
Plano 2: Vista Planta

Anexo II

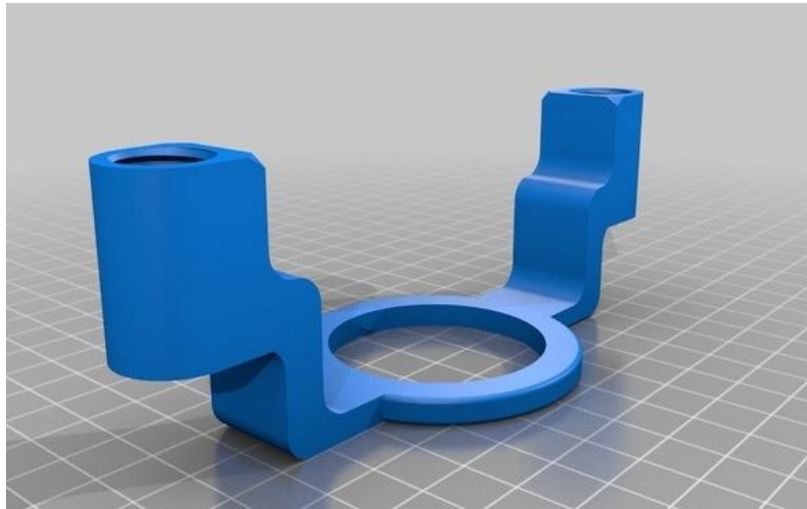
Piezas impresas



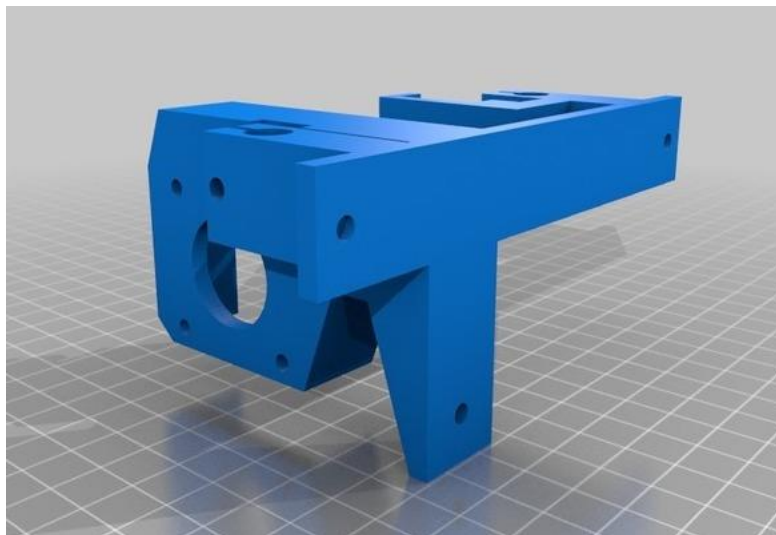
Pieza impresa 1: Base



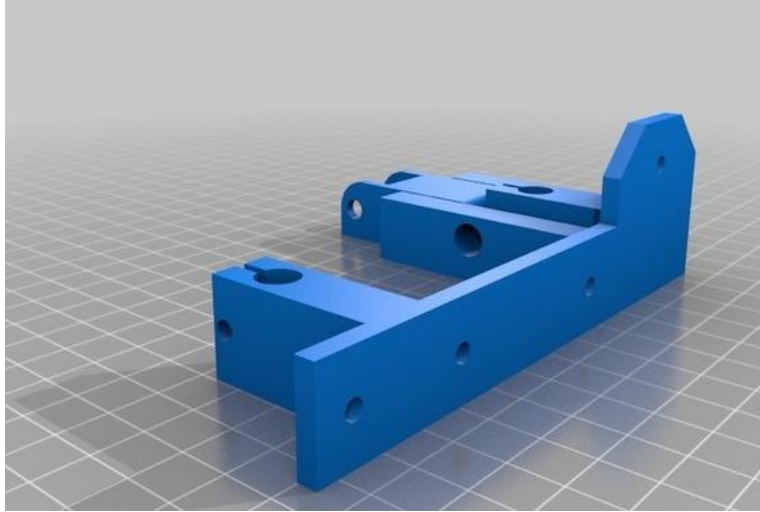
Pieza impresa 2: Tornillo de elevación



Pieza impresa 3: Pulsador de dispensador



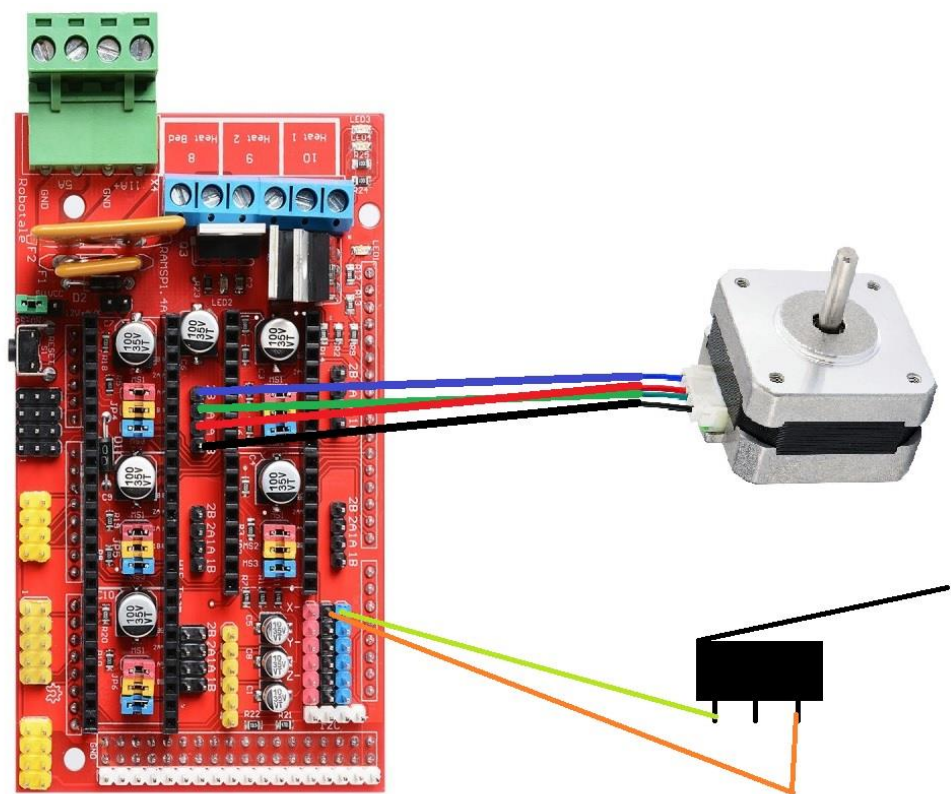
Pieza impresa 4: Soporte motor eje x y fin carrera



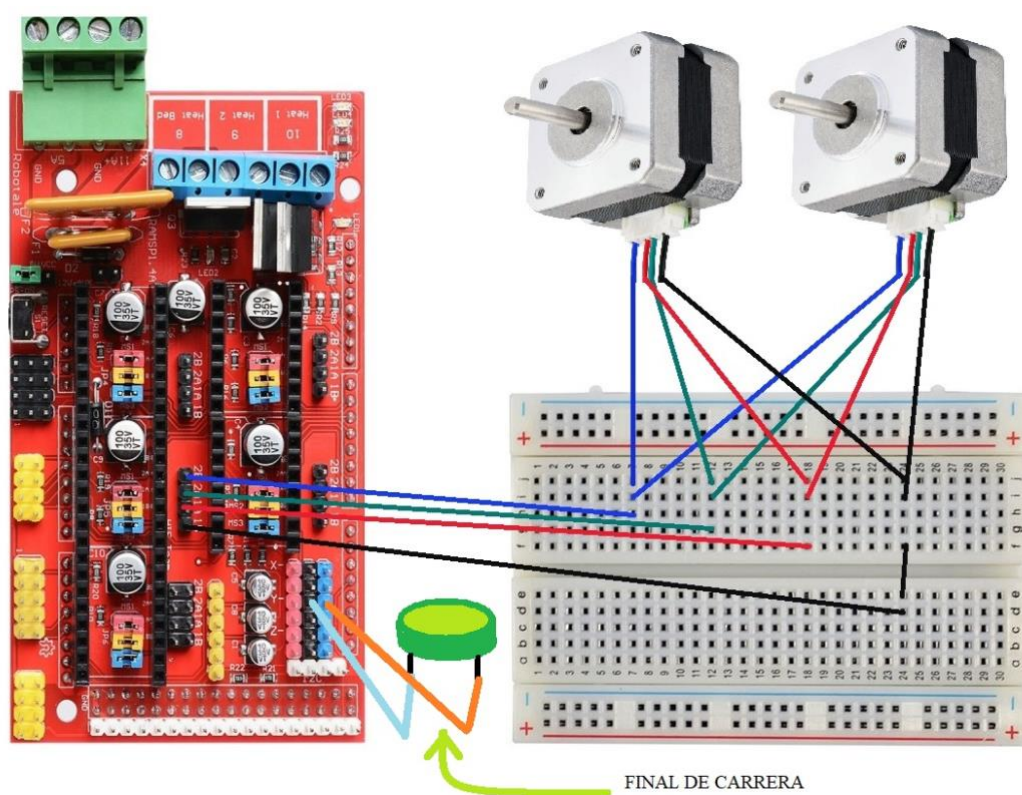
Pieza impresa 5: Soporte y tensor eje x

Anexo III

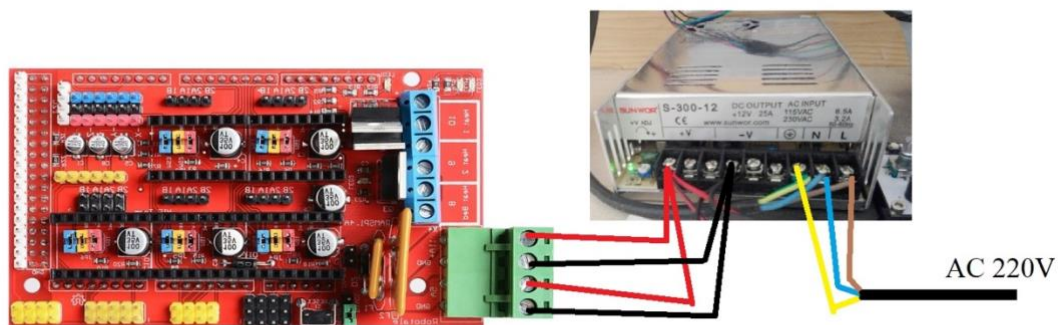
Esquemas eléctricos.



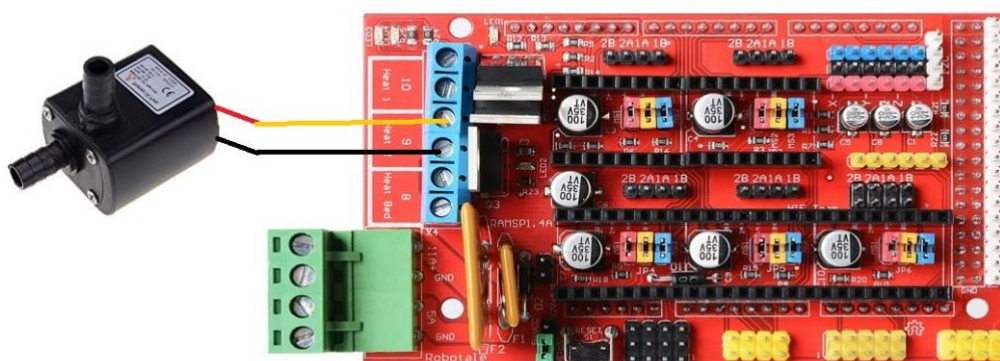
Esquema eléctrico 1: Motor eje X y pulsador fin de carrera



Esquema eléctrico 2: Motores eje Y y pulsador para calibrar



Esquema eléctrico 3: Conexión con la fuente de alimentación



Esquema eléctrico 4: Conexión de la bomba