

TITULACIÓN: GRADO EN INGENIERÍA ELECTRÓNICA  
INDUSTRIAL Y AUTOMÁTICA

## **Trabajo de Fin de Grado**

**“DISEÑO DE UN SISTEMA DE CONTROL DE BAJO  
COSTE DE UNA UNIDAD DE RENDIJAS  
CONFIGURABLES DE SUSTITUCIÓN PARA EL  
ESPECTRÓGRAFO MULTIOBJETO INFRARROJO  
EMIR”**

AUTOR:

DÁMASO GONZÁLEZ BARRERA

TUTOR:

JESUS MIGUEL TORRES JORGE

<b>1. LISTA DE ACRÓNIMOS Y ABREVIATURAS</b>	<b>1</b>
<b>2. INTRODUCCIÓN</b>	<b>1</b>
2.1 Estado del arte	4
2.2 Objetivos	5
2.3 Metodología	5
<b>3. REQUERIMIENTOS</b>	<b>7</b>
3.1. Requerimientos funcionales	7
3.2. Requerimientos de rendimiento	8
3.3. Requerimientos físicos	8
3.4. Requerimientos de control	9
<b>4. DESARROLLO DEL HARDWARE</b>	<b>9</b>
4.1. Diseño electrónico	9
4.2.1. Motores	9
4.2.2. Sensores	11
4.2.2.1. Sensores de home	12
4.2.2.2. Sensores de contacto entre barras	14
4.2.2.3. Sensores contacto con el travesaño	15
4.2.3. Arduino	16
4.2.4. Driver A4988	18
4.2.5 Fuente de alimentación	22
4.2.6 Centralización de la electrónica	23
4.2 Conexiones	25
4.2.1. Conectores	25
4.2.2. Cableado	25
<b>5. DESARROLLO DEL SOFTWARE</b>	<b>26</b>
5.1. Lenguaje	26
5.2. Entorno de desarrollo	27
5.3. Código de programación	29
5.3.1. Funciones	29
5.4 Carga del programa	36
<b>6. PRUEBAS REALIZADAS</b>	<b>38</b>
<b>7. PROBLEMAS Y SOLUCIONES</b>	<b>38</b>
<b>8. CÁLCULOS</b>	<b>42</b>
8.1. Pasos por vuelta	42
8.2. Velocidad de giro	42

8.3. Velocidad de avance horizontal	43
8.4. Tiempo en llegar a una posición concreta	43
8.5. Posición de una barra	44
<b>9. PRESUPUESTO MATERIAL</b>	<b>44</b>
<b>10. ABSTRACT AND FUTURE WORK</b>	<b>45</b>
10.1 Abstract	45
10.2 Future works	46
<b>11. REFERENCIAS</b>	<b>47</b>
<b>12. ANEXO 1: Esquema de centralización de la electrónica</b>	<b>48</b>
<b>13. ANEXO 2: Esquema de conexiones CSUR-Caja</b>	<b>49</b>
<b>14. ANEXO 3: Disposición de conexiones en los conectores</b>	<b>49</b>
<b>15. ANEXO 4: Hoja de datos sensores de home MY-COM H75/80</b>	<b>50</b>
<b>16. ANEXO 5: Hoja de datos de la fuente de alimentación RSP-320-24</b>	<b>51</b>
<b>17. ANEXO 6: Hoja de datos del driver A4988</b>	<b>53</b>
<b>18. ANEXO 7: Diagrama de flujo del programa del sistema de control</b>	<b>56</b>
<b>19. ANEXO 8: Conexiones eléctricas</b>	<b>57</b>
<b>20. ANEXO 9: Relación de pines Arduino-Drivers y Arduino-Sensores</b>	<b>59</b>

## 1. LISTA DE ACRÓNIMOS Y ABREVIATURAS

---

<b>CSU</b>	Configurable Slit Unit
<b>CSUR</b>	CSU de Repuesto
<b>EMIR</b>	Espectrógrafo Multiobjeto InfraRojo
<b>ECS</b>	EMIR Control System
<b>GTC</b>	Gran Telescopio Canarias
<b>IAC</b>	Instituto de Astrofísica de Canarias
<b>IR</b>	Infrarrojo
<b>MOS</b>	Multi-Object Spectroscopy
<b>ORM</b>	Observatorio del Roque de los Muchachos (La Palma)

## 2. INTRODUCCIÓN

---

Un espectrógrafo es instrumento de medición destinado al estudio de los espectros de los cuerpos físicos, el mismo registra fotográficamente, o por medio de detectores fotosensibles, los diferentes componentes del espectro. El espectrógrafo se encuentra formado por los siguientes componentes:

- Telescopio: Sirve para amplificar la luz de los objetos que se van a observar, pero no tiene ni ocular ni cámara.
- Rendija: La imagen producida por el telescopio se enfoca sobre una rendija la cual define el tamaño y la porción de la imagen de la cual se necesita obtener el espectro. La imagen de la rendija será enfocada sobre la superficie del detector, de tal forma que las dimensiones de la rendija definen la resolución del instrumento. Una rendija ancha produce la entrada de una mayor cantidad de luz, genera una imagen más grande y por



lo tanto menor resolución. Una rendija delgada produce una entrada menor de luz y genera una imagen de menor tamaño pero con mayor resolución.

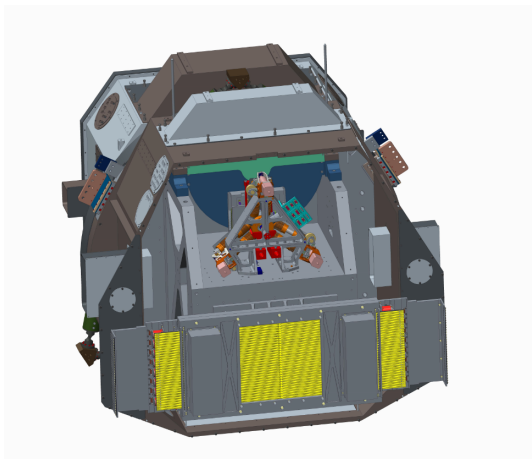
- Colimador: Sirve para producir un haz de luz paralelo cuya dimensión coincida con las dimensiones del prisma.
- Prisma: Sirve para dispersar la luz. Las dimensiones, la forma y el tipo de vidrio, determinan la medida de la dispersión de la luz.
- Lente y detector: En el caso de que la observación del espectro sea visual, el lente es un ocular y el detector es el ojo. Si el espectro se registra en una película o en un CCD, la longitud focal del lente determina la amplificación del espectro, determinando la cantidad de detalles visibles y su luminosidad.

Con un espectrógrafo es posible conocer las condiciones de temperatura, campos eléctricos y magnéticos, composición química y velocidad relativa de un objeto que se encuentra a millones de kilómetros.

EMIR (Espectrógrafo Multiobjeto InfraRojo) es un espectrógrafo multiobjeto de gran campo, en el infrarrojo, propuesto para el foco Nasmyth del GTC (Gran Telescopio de Canarias). Está equipado, entre otros, con tres subsistemas de alta tecnología de última generación, algunos especialmente diseñados para este proyecto: un sistema robótico reconfigurable de rendijas (CSU); elementos dispersores formados mediante la combinación de redes de difracción de alta calidad, fabricadas mediante procedimientos fotorresistivos, y prismas convencionales de gran tamaño, y el detector HAWAII-2 de Rockwell, diseñado para el infrarrojo cercano con un formato de 2048x2048 píxeles, y dotado de un novedoso sistema de control. EMIR Permitirá a los observadores obtener entre decenas y centenas de espectros de resolución intermedia simultáneamente, en las bandas de IR cercano, Z, J, H, K [6].

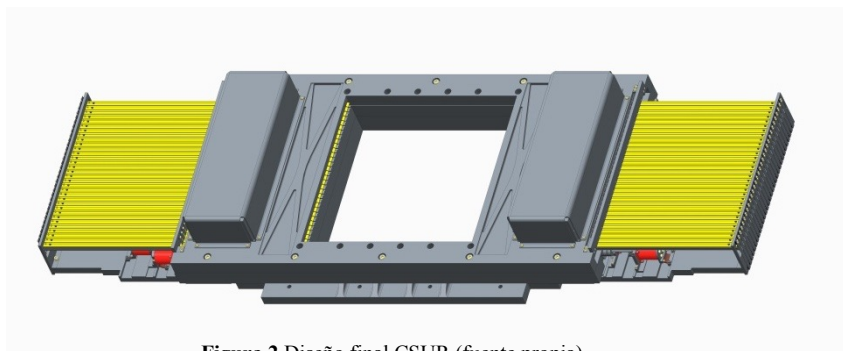
EMIR permite dos modos de observación: espectroscopía multiobjeto y modo imagen. Pero será la espectroscopía multiobjeto la funcionalidad de mayor prioridad y para ello es imprescindible una “Unidad de rendijas configurables” -en inglés, “Configurable Slit Unit”- (CSU).

La CSU diseñada consiste básicamente en una abertura por la que ha de pasar el haz de luz a lo largo de su recorrido hasta el detector, en la cual se coloca una batería de barras enfrentadas que, según la posición en la que se encuentren, bloquearán o desbloquearán el campo nominal de visión, permitiendo que el haz de luz llegue o no al detector.



**Figura 1** Diseño final CSUR en banco óptico de EMIR (fuente propia)

El diseño de la CSU se basa en el diseño proporcionado por el IAC que emplea motores paso a paso compatibles con ambientes criogénicos para mover, a través de husillos, las barras de la CSU de repuesto (“CSUR”), que llevan talladas las correspondientes cremalleras.



**Figura 2** Diseño final CSUR (fuente propia)

El diseño de la CSUR se basa en la CSU “titular” formada por 104 barras y con un sistema de control basado en PLCs industriales. Las modificaciones a dicho diseño tienen por objeto reducir la complejidad del mismo, para hacer viable su fabricación en vista de las limitaciones de presupuesto, sin que ello repercuta en la funcionalidad del mecanismo ni en su capacidad para cumplir las especificaciones.

La CSUR constará de 20 rendijas; y por tanto habrá 40 barras (el total de ambos lados) y sus correspondientes 40 motores, interruptores de fin de carrera y sensores de contacto entre barras.

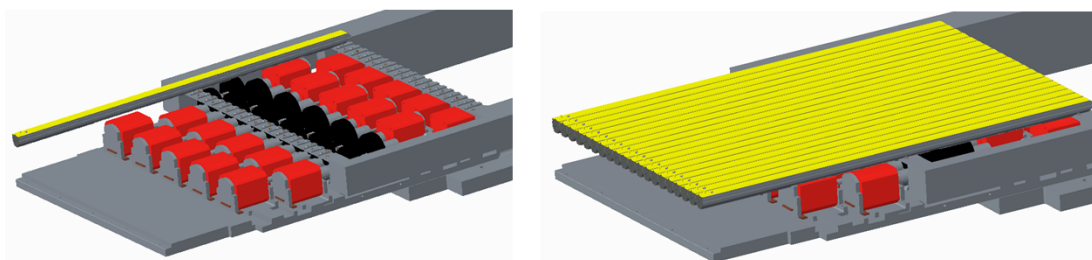


Figura 3 Diseño final CSUR nueva distribución con 20 barras (fuente propia)

La CSUR dispone de un límite físico en el movimiento longitudinal de las barras formado por un travesaño transversal a éstas y otro límite físico de home donde las barras resetean su posición y a partir de estas se toman su referencia. El travesaño dispondrá de un sensor de contacto y el home dispondrá de finales de carrera en caso de que alguna barra llegue a cierto límite. El espacio entre estos dos límites físicos es la distancia máxima de desplazamiento de las barras.

Se realizará un prototipo de la CSUR con únicamente 6 barras. El prototipo de 6 barras es esencialmente la unidad final de la CSUR, pero llevará incorporadas sólo 6 barras, y todos los elementos mecánicos correspondientes a las mismas: rodamientos, sistema de motor husillo y sus ejes soportes, interruptores de fin de carrera y de emergencia, sistemas de freno guía.

## 2.1 Estado del arte

En un espectrógrafo, el ocular se sustituye por una cámara. Se pueden calcular sus longitudes de onda a partir de sus posiciones en la película fotográfica. Existen dos tipos de espectrógrafo: Espectrógrafo con prisma y Espectrógrafo con rejilla de difracción, ambos tienen el mismo principio de funcionamiento la única diferencia es en el elemento encargado de crear el espectro, en un caso es el prisma y en el otro la rejilla de difracción.

La CSU “titular” actual cuenta con un conjunto de 104 barras, y por tanto 52 rendijas, y es controlada mediante PLCs industriales. Por lo que, para que sea económicamente viable, se ha reducido el número de barras de la CSUR, contando ahora con 40 barras y 20 rendijas manteniendo todas las funcionalidades del mecanismo sin que ello repercuta en la en su capacidad para cumplir las especificaciones.

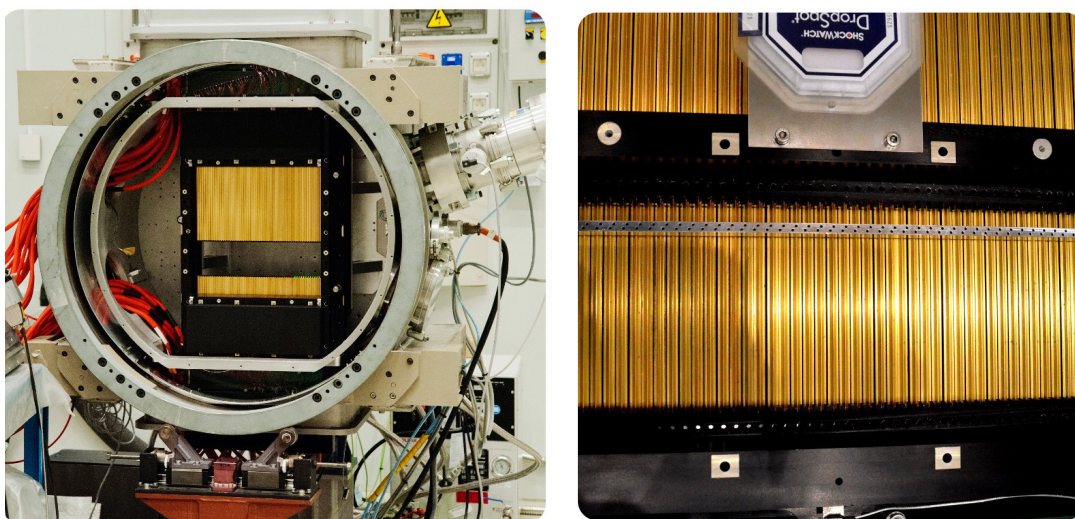


Figura 4 CSU titular instalada en el EMIR (fuente [www.iac.es](http://www.iac.es))

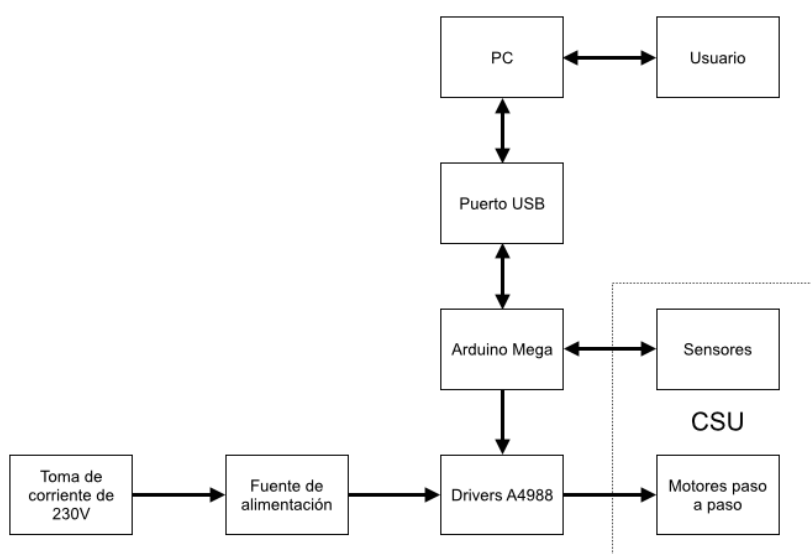
## 2.2 Objetivos

Este trabajo tiene por objeto principal realizar el diseño del sistema de control de la CSUR que sustituiría a la CSU “titular” de EMIR en los periodos de mantenimiento de la misma buscando la realización de un sistema de bajo coste que cumpla con todos los requerimientos y permita la realización de pruebas en el sistema mecánico del prototipo de 6 barras para su mejora.

## 2.3 Metodología

El presente Trabajo Fin de Grado se basa fundamentalmente en el diseño de la electrónica e informática para realizar un sistema de control que cumpla con los requerimientos especificados por el IAC. Dichos requerimientos estarán redactados en el Capítulo de Requerimientos.

La realización de este trabajo se ha llevado a cabo en varias fases. En primer lugar, se ha realizado un estudio previo sobre la mejor forma de realizar un óptimo control sobre el prototipo. Luego se han diseñado y probado, realizado las pertinentes conexiones, las partes electrónicas del sistema de control a la par que se ha diseñado el programa informático. En esta fase se han ido realizando mejoras a partir del diseño de partida debido a complicaciones que no se habían tenido en cuenta en un primer momento para hacer el diseño mas intuitivo y fácil de controlar por el usuario.



**Figura 5** Esquema general del funcionamiento del sistema de control

Una vez terminado el sistema de control, fue probado en la CSU, lo que permitió evaluar de un modo más realista el resultado obtenido y observar errores a corregir.

El sistema de control desarrollado está orientado al control del prototipo de 6 barras, pero su funcionamiento será exactamente el mismo que el de la unidad final de la CSUR. En la *Figura 4* se muestran las distintas partes que interactúan en el sistema de control.

A lo largo de este documento se detalla la función de cada una de las partes, desde el punto de vista de hardware y software. Únicamente se mencionarán partes esenciales desde el punto de vista mecánico y se centrará en todo el proceso de diseño de la electrónica de control.

Por otro lado, este sistema de control se ha desarrollado para la empresa INSTRUTEC S.L.U., la cuál ha sido la encargada del diseño de todas las partes mecánicas del prototipo de 6 barras de la CSUR, para la realización de pruebas y mediciones.

### 3. REQUERIMIENTOS

---

A continuación se resumen las especificaciones de la CSU, tal y como se detallan en la documentación interna del IAC, para su posterior referencia en subsiguientes secciones. Se han incluido sólo aquellos requerimientos que son significativos para el diseño del sistema de control de la unidad.

La información recogida en este apartado es un extracto del documento “*Configurable Slit Unit Contractual Specifications*” [4].

#### 3.1. Requerimientos funcionales

- **Concepto general:** La CSU permitirá hacer espectroscopía, con rendija larga o multiobjeto, mediante el uso de rendijas de gran longitud o configuraciones multi-rendija respectivamente, en el campo nominal de visión. La unidad deberá ser capaz de proporcionar las distintas configuraciones del campo nominal que sean especificadas por el usuario. El concepto de cómo posicionar las rendijas se basará en un mecanismo de rendijas configurables controlado desde el ECS (EMIR Control System), en el que múltiples barras se mueven ya sea para bloquear (para configurar una máscara de múltiples rendijas para la espectroscopía multiobjeto o de rendijas largas para la espectroscopía estándar) o desbloquear (para el modo imagen) el campo nominal de visión.
- **Capacidad de imagen directa:** La CSU deberá ser capaz de eliminar todas las rendijas del campo de visión.
- **Bloqueo del mecanismo bajo desconexión:** El mecanismo debe mantener la configuración de geometría requerida cuando todos los componentes electromecánicos estén desactivados.

## 3.2. Requerimientos de rendimiento

- **Tiempo de reconfiguración de las rendijas:** El tiempo máximo de reconfiguración de la CSU deberá ser inferior a 5 minutos.

## 3.3. Requerimientos físicos

- **Superficie útil para imágenes:** El área sin obstrucciones para la formación de imágenes cuando se retiran las barras será de  $340 \times 340 \text{ mm}^2$ . Esta área deberá estar libre de componentes del mecanismo.
- **Superficie útil para espectroscopía:** La superficie útil para la espectroscopía será de al menos  $340 \times 205 \text{ mm}^2$ , la dimensión más grande está alineada con el eje XCU.

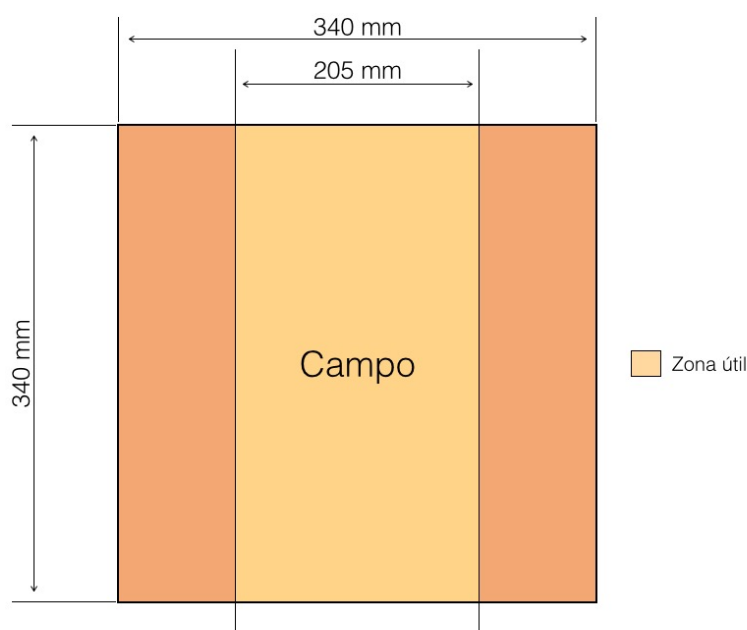


Figura 6 Dimensiones del campo y superficie útil para espectroscopía

- **Longitud de rendija larga:** La longitud de la rendija será de 340 mm.
- **Ancho de rendija larga:** El ancho de la rendija deberá ser seleccionable por el usuario en un intervalo de al menos entre 340 y  $1000 \mu\text{m}$ .
- **Longitud de las rendijas en la configuración multi-rendija:** La longitud de cada rendija será tal que el área descrita esté cubierta por al menos 52 rendijas. La longitud de la rendija seleccionada será la misma para todas las rendijas del conjunto, con la posible excepción de las que están en cada extremo de la CSU en la dirección XCSU.

- **Ancho de las rendijas en la configuración multi-rendija:** El ancho de la rendija deberá ser seleccionable por el usuario en un intervalo de al menos entre 340 y 1000 $\mu$ m.

### 3.4. Requerimientos de control

- **Comandos de movimientos.** El sistema de control deberá contar con los siguientes comandos o instrucciones:
  - Mover un conjunto de barras hacia la posición de home
  - Parar movimiento
  - Mover una barra o conjunto de barras hacia una posición determinada
  - Configurar una rendija o conjunto de rendijas en una posición y con un ancho determinados
- **Estados.** El sistema de control deberá indicar los siguientes estados:
  - Operando
  - Comando hecho o terminado
  - Fallo
  - Información relevante

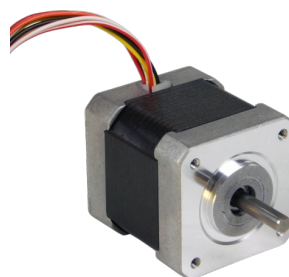
## 4. DESARROLLO DEL HARDWARE

---

### 4.1. Diseño electrónico

#### 4.2.1. Motores

Los motores seleccionados son unos motores paso a paso APPLIED MOTION HT17-275 de corriente continua (*Figura 7*) cuyas características se redactarán en la *Tabla 1*.



**Figura 7** Motor Applied Motion HT17-275



<b>Características técnicas HT17-275 (Bipolar)</b>	
Par máximo mantenido	5,61 kg-cm
Ángulo de paso	1,8°
Corriente en conexión serie	0,85 A/fase
Resistencia en conexión serie	6,6 Ohms/fase
Inductancia en conexión serie	12,8 mH/fase
Corriente en conexión paralela	1,7 A/fase
Resistencia en conexión paralela	1,7 Ohms/fase
Inductancia en conexión paralela	3,2 mH/fase

**Tabla 1** Características técnicas motores HT17-275

Los datos extraídos del documento de Especificaciones de Applied Motion, indican que el motor HT17-275: “En movimiento, a 1 vuelta por segundo en modo 1/5 de micropaso, con 0.8 A, el motor puede entregar hasta 2100 grf·cm, independientemente de la temperatura, en un rango de 97K hasta 130K, y a temperaturas bajas incluso más, hasta 2900 grf·cm en movimientos cortos”. La velocidad de 1 vuelta por segundo con un paso de 2 mm en la cremallera de las barras (siguiendo el diseño del IAC) implica un posicionado total de 272.5 mm (avance máximo de las barras) en 137 segundos, lo cual cumple las Especificaciones generales de la CSU de reconfiguración en 300 segundos (ver apartado 3.2).

El motor HT17-275 es capaz de entregar el par necesario independientemente del tipo de conexión, por lo que se ha optado finalmente por una conexión en serie, que necesita una menor corriente pudiendo reducir de esta manera la corriente máxima de la fuente de alimentación (ver apartado 4.2.5), lo que permite el uso de una fuente de alimentación mas económica. Según el datasheet de los motores, la conexión serie admite una corriente máxima de 0,85 A por fase, por lo que se limitará a 0,8 A por seguridad y para evitar sobrecalentamientos.

Características técnicas HT17-275 (Bipolar serie)	
Corriente en conexión serie	0,85 A/fase
Resistencia en conexión serie	6,6 Ohms/fase
Inductancia en conexión serie	12,8 mH/fase

Tabla 2 Características técnicas motores HT17-275 en conexión bipolar serie

## 4.2.2. Sensores

El sistema de control, dispondrá de tres tipos de sensores: sensores de llegada a home (ver apartado 4.2.2.1), de contacto entre barras (ver apartado 4.2.2.2), y de contacto con el travesaño (ver apartado 4.2.2.3). Todos los sensores irán implementados con resistencias en configuración pull-up o pull-down, según proceda, para forzar que éstos proporcionen dos únicos estados: HIGH (un "1" lógico) y LOW (un "0" lógico). De tal manera que una configuración pull-down forzará la salida a cero (LOW) mientras el sensor no se encuentre activo, y justo al contrario ocurre con la configuración pull-up, que forzará la salida a 5V (HIGH) mientras el sensor no se encuentre activo.

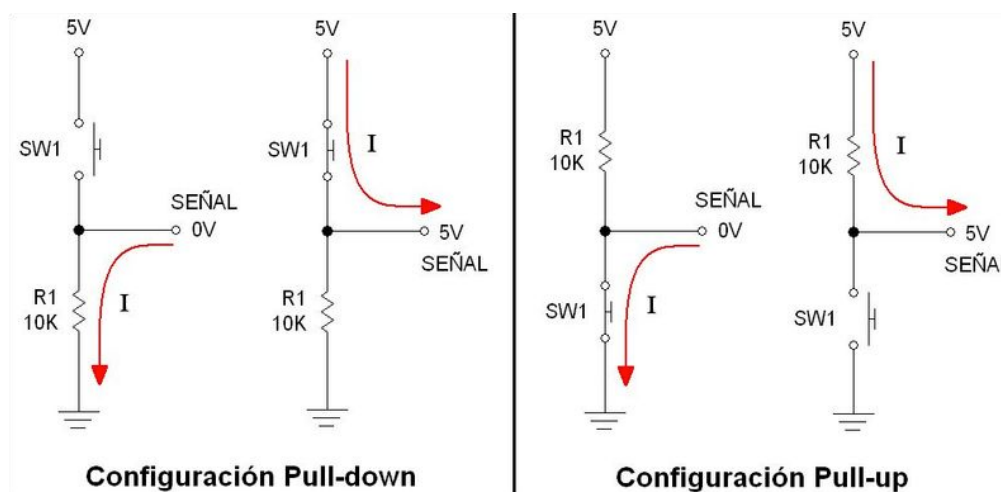


Figura 8 Esquemas de configuración pull-down y pull-up

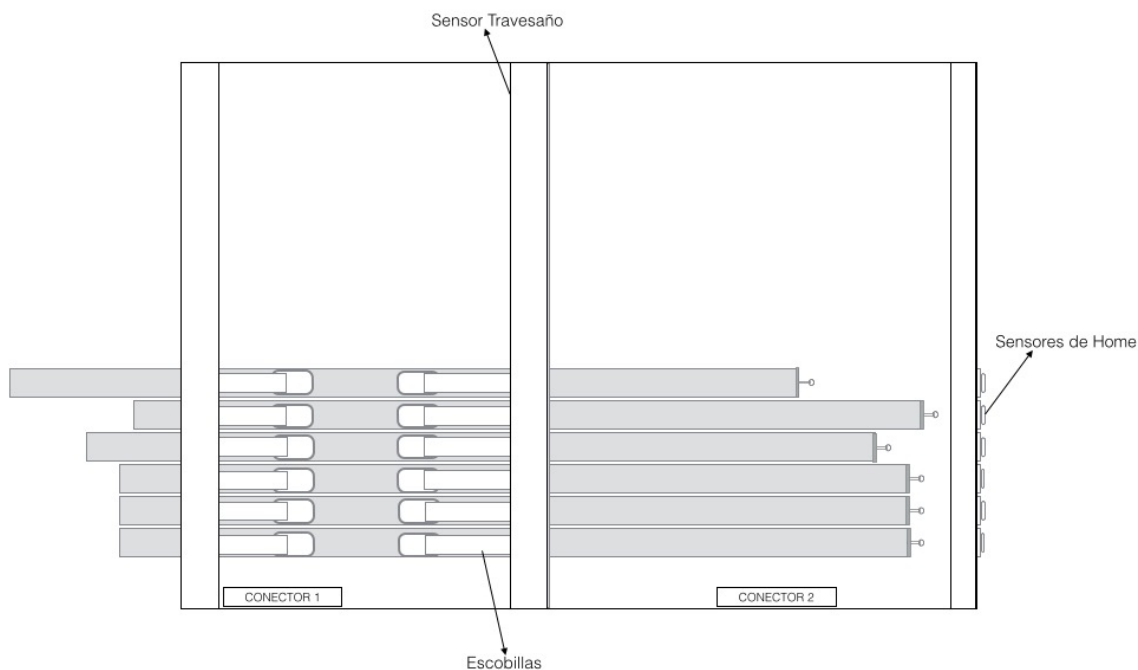


Figura 9 Esquema del alzado del conjunto de sensores

#### 4.2.2.1. Sensores de home

El conjunto de los sensores de home está compuesto por una placa perforada que permitirá el paso de las barras a su través y que se atornilla a un extremo de la placa de soporte inferior; las 20 galgas de los sensores con las que contactarán las barras y abrirán el circuito para la detección de la posición de las mismas; y los sensores MY-COM H75/80 fin de carrera. Cuando la galga pierde el contacto con el sensor, el fin de carrera abre el contacto y la electrónica de control detiene el retroceso de la barra (*Figura 10*).

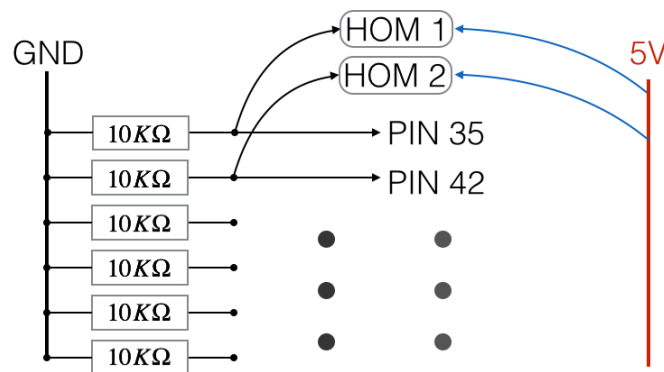


Figura 10 Sensor de home MY-COM H75/80

La función de los sensores de home es la de detener el movimiento de las barras en cuanto éstas hayan llegado a la posición de home y resetear la posición de las barras siendo esa la posición de referencia a partir de la cuál se determinará la posición de las barras contando los pasos dados por el motor (ver apartado 5.3.1).

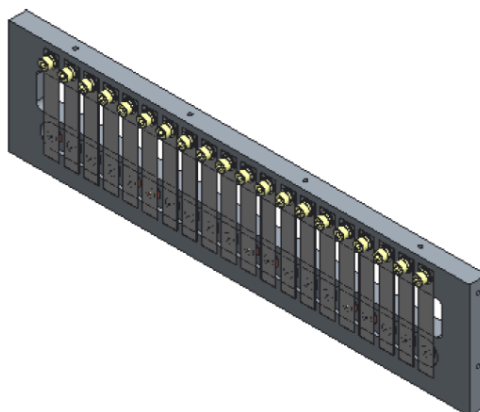
Los sensores de home MY-COM H75/80 son finales de carrera con un contacto normalmente cerrado y los cuales disponen de dos conductores. Se dispondrán en configuración pull-down, de tal manera que cuando el sensor no esté activo su estado será permanentemente de HIGH y cuando éste se active pasará al estado de LOW.

Los sensores MY-COM H75/80 estarán forzados mecánicamente por una pequeña lámina o galga metálica, la cual será deformada cuando la barra la empuje, cerrando el contacto del fin de carrera. Por lo tanto el estado del sensor de home será LOW mientras las barras permanezcan en la posición de home (sensor activo), y HIGH mientras las barras se encuentren en el campo (sensor no activo). Su configuración se describe en la *Figura 11*.



**Figura 11** Esquemas de conexión de los sensores mediante resistencias en configuración pull-down

Las barras llevan en su extremo una lámina de acero inoxidable que actuará como tope mecánico en caso de que falle el sistema de control. También esta lámina previene las sobrecargas que las barras podrían ejercer sobre las galgas de home en caso de que al llegar a la posición de home, el sistema de control no detuviese el motor a tiempo.



**Figura 12** Conjunto de sensores donde se pueden ver las láminas

#### **4.2.2.2. Sensores de contacto entre barras**

La función de los sensores de contacto entre barras es la de detener el movimiento de las barras en cuanto se produzca un contacto entre dos barras que se mueven en direcciones opuestas.

El sistema de sensores de contacto estará formado por una barra conectada por unas escobillas metálicas a 5V (HIGH) mediante configuración pull-up y su barra par conectada a 0V (LOW). Cada barra desliza sobre cuatro patines plásticos y se mantienen en su posición por ser presionadas por dos rodillos. Este conjunto incluye también los arcos a los cuales están sujetos los mencionados rodillos, los cuales mantienen a las escobillas en constante contacto con la barra. Mientras las barras no hagan contacto el sensor permanecerá HIGH (sensor no activo), en cuanto las barras hagan contacto, el sensor pasará inmediatamente al estado de LOW (sensor activo).

Ejemplo: El sensor de contacto permanecerá HIGH mientras la barra 1 y la barra 2 no hagan contacto entre sí. En el momento en que éstas hagan contacto, el sensor pasará a estar LOW ya que la barra 2 estará conectada a 0v.

En el prototipo de 6 barras, el contacto entre barras no será posible físicamente ya que se han colocado las 6 barras paralelamente, por lo que el contacto se simulará mediante el contacto de la barra con un cable conectado a una tierra cualesquiera o alguna parte del armazón metálico que se considera aislado. Esto es así porque todas las barras se encontrarán conectadas a 5V para el uso en el sensor de travesaño. Para recrear lo mejor posible las condiciones reales, al simular el contacto se detendrá dicha barra y su barra par.

Ejemplo: Al querer simular el contacto entre las barras 1 y 2, si se realiza un contacto en cualquiera de ellas se detendrán ambas.

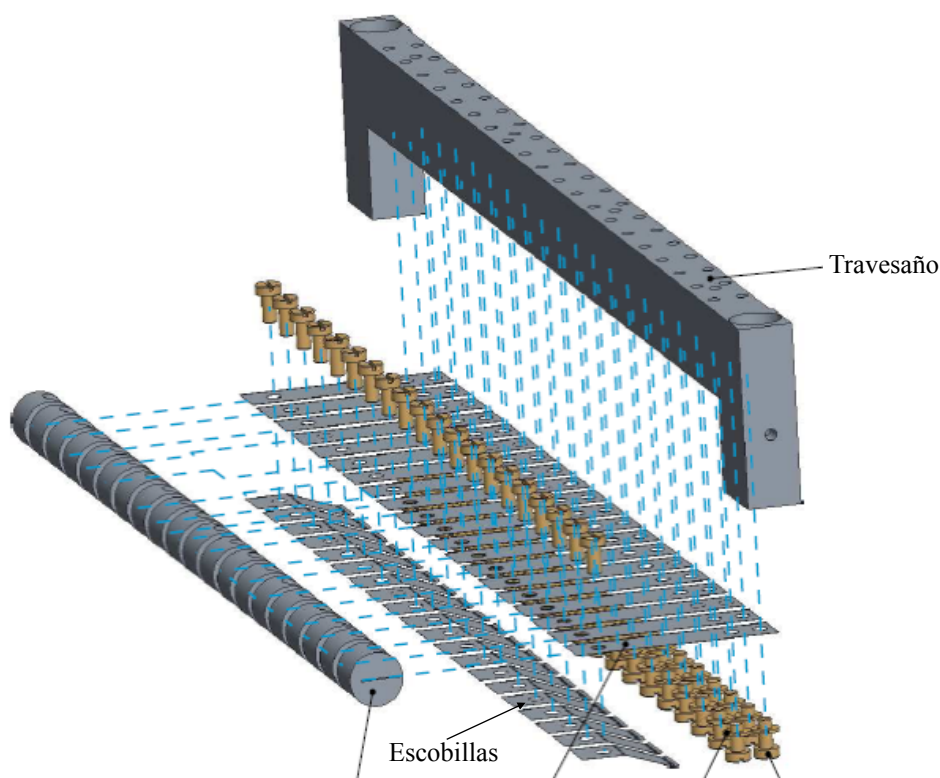


Figura 13 Conjunto de rodillos, ballestas y escobillas

#### 4.2.2.3. Sensores contacto con el travesaño

La función de los sensores de contacto con el travesaño es la de detener el movimiento de las barras en cuanto una de ellas haga contacto con el travesaño. Este sistema no permite discernir que barra ha hecho contacto con el travesaño, por lo que el sistema detendrá el movimiento de todas las barras cuando una barra haga contacto.

Para el prototipo de 6 barras el sensor de contacto con el travesaños se coloca uno en configuración pull-down, esto es así ya que se aprovecha las conexiones de los sensores de contacto para forzar las señales de contacto con el travesaño. De esta manera es posible simplificar el sistema de sensores.

El sensor del travesaño permanecerá LOW (sensor no activo) mientras ninguna de las barras haga contacto con él. En el momento en que ésta haga contacto, el sensor del travesaño pasará a estar HIGH (sensor activo) ya que la barra estará conectada a 5v.

En el prototipo de 6 barras, al encontrarse todas las barras paralelamente en el mismo armazón y conectadas a 5V, el sensor de travesaño permanecerá inactivo mientras ninguna barra haga contacto con este.

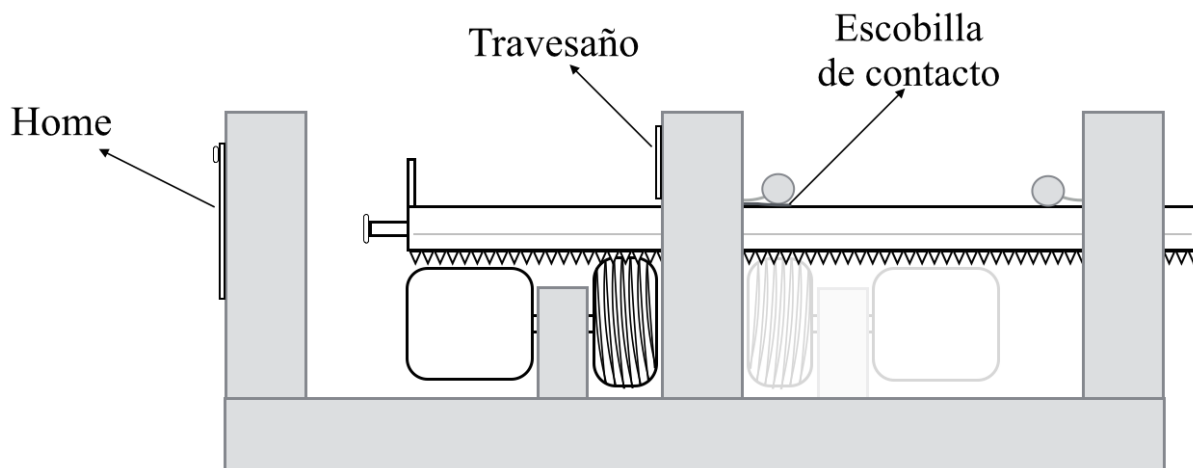


Figura 14 Esquema del perfil del conjunto de sensores

### 4.2.3. Arduino

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos. Arduino puede recoger información del entorno a través de sus pines de entrada y alterar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa arduino se programa mediante el lenguaje de programación arduino (basado en Wiring) y el entorno de desarrollo arduino (basado en Processing). Los proyectos de arduino pueden ejecutarse sin necesidad de que exista conexión a un ordenador, aunque puede existir esta comunicación para comunicarlo con diferentes tipos de software (ejemplos: Flash, Processing, MaxMSP...).

Las placas de arduino se pueden hacer a mano o comprarse montadas de fábrica, y el software se puede descargar de forma gratuita. De la gama de modelos que presenta arduino, se ha escogido el arduino Mega 2560 (Figura 15), ya que sus prestaciones son suficientes para poder implementar el sistema de control. En la Figura 16 aparecen las principales características. Arduino da la posibilidad de alimentarlo por medio de un cable USB, lo cual

es muy útil si se está programando constantemente, o por medio de un conector Jack, para un uso de arduino sin necesidad de tenerlo conectado al ordenador.



**Figura 15** Tarjeta Arduino Mega 2560

El criterio para la selección de la tarjeta Arduino ha sido el del número de pines necesarios para el sistema de control.

El número de pines necesario es de 47:

- 6 x sensores de home
- 2 x sensores de travesaño
- 3 x sensores de contacto
- 6 x pines de ENABLE
- 6 x pines de DIR
- 6 x pines de STEP

La tarjeta Arduino Mega 2560 dispone de 54 entradas/salidas digitales de las que 14 pueden proporcionarnos señales PWM de salida. Ésta tarjeta trabaja con señales entre 5V y 0V, siendo los valores entre el rango 2,5V - 5V el estado de HIGH y 0V - 2,5V el de LOW, y cuenta con varios pines que nos proporcionan las tensiones (5V y GND) (ver *Figura 16*).



Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

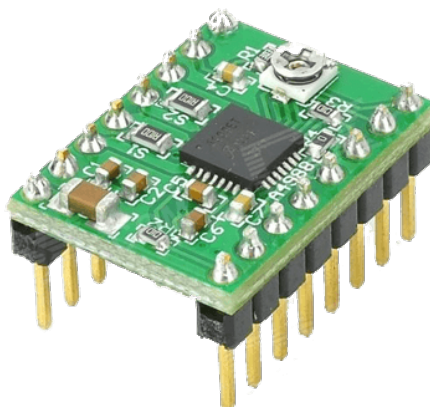
**Figura 16** Características técnicas de la tarjeta Arduino Mega 2560

Para el prototipo, los 40mA de corriente de salida que son capaces de generar los pines de la tarjeta arduino no son suficientes para hacer que los motores paso a paso giren (ver apartado 4.2.1), por lo que es necesario amplificar esa corriente. Para ello se hace uso de los drivers A4988 (ver apartado 4.2.4).

En el *ANEXO 9* se detalla la relación de pines entre arduino, drivers y sensores.

#### **4.2.4. Driver A4988**

Este producto incorpora el chip de Allegro A4988 de en una placa de fácil conexión y utilización. Es usado como controlador de motores paso a paso de hasta 2 A de corriente por bobina.



**Figura 17** Driver A4988

## Características

- Cinco resoluciones diferentes: paso completo, medio paso, un cuarto de paso, un octavo de paso, y un dieciseisavo de paso.
- Control de corriente ajustable que permite ajustar la salida de corriente máxima con un potenciómetro, que le permite utilizar tensiones superiores a la tensión nominal del motor paso a paso para lograr mayores tasas de paso.
- Protección por sobrecalentamiento térmico, cierre por baja tensión, y protección por sobrepico de corriente.

Los drivers A4988 seleccionados disponen de 16 pines, 8 pines destinados a las señales digitales de entradas y 8 pines de potencia para la alimentación de los motores. Se ha decantado por estos drivers porque permiten una resolución de hasta 1/16 de paso y limitar la corriente de alimentación de los motores.

El controlador requiere una tensión de alimentación lógica (3-5,5 V) que se conecta a través de los pines VDD y GND y una tensión de alimentación del motor (8-35 V) para ser conectada a través de VMOT y GND (Ver *Figura 18*):

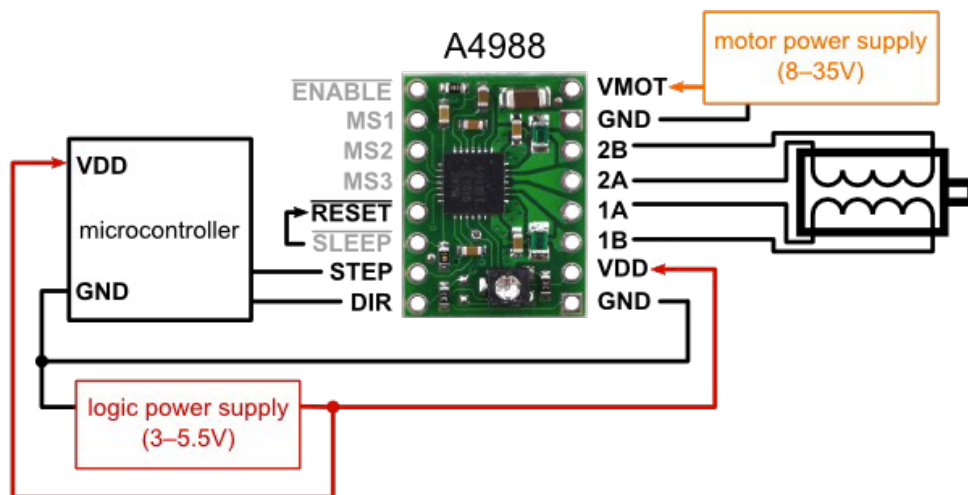


Figura 18 Esquema de conexiones del driver A4988

- **VMOT y GND:** Para la alimentación del motor, permitiendo un máximo de 35V. En nuestro caso una fuente de alimentación de 24V.

- **2B,2A,1A y 1B:** Conexión de las fases de las bobinas del motor.

- **VDD y GND:** Alimentación del pin de 5V y GND de la tarjeta Arduino.

Los motores paso a paso seleccionados tienen una especificación de tamaño de paso de 1,8° o 200 pasos por vuelta. Un controlador de microstepping tal como el A4988 permite resoluciones más altas, permitiendo ubicaciones en paso intermedio, que se consiguen por la activación de las bobinas con los niveles de corriente intermedios. Por ejemplo, controlando los motores en el modo de paso por cuartos (1/4, quarter-step) dará 800 micropasos por vuelta mediante el uso de cuatro niveles de corriente diferentes.

La resolución (tamaño de paso) puede ser seleccionada mediante los pines MS1, MS2, MS3 que permiten cinco resoluciones diferentes de paso de acuerdo con la *Tabla 3*. MS1 y MS3 tienen una resistencia pull-down interna de 100 kΩ y MS2 tiene una resistencia pull-down interna 50kΩ, así que dejar estos tres pines al aire resultará en el modo de paso completo (full-step mode).

MS1	MS2	MS3	RESOLUCION
LOW	LOW	LOW	Paso Completo
HIGH	LOW	LOW	$\frac{1}{2}$
LOW	HIGH	LOW	$\frac{1}{4}$
HIGH	HIGH	LOW	$\frac{1}{8}$
HIGH	HIGH	HIGH	$\frac{1}{16}$

**Tabla 3** Configuración de las señales MS1, MS2 Y MS3

La resolución será fija y de 1/16 de paso, por lo que los pines MS1, MS2 Y MS3 se conectarán a 5V, que equivale a un HIGH.

Cada pulso de entrada escalón o flanco de subida en el pin STEP corresponde a un micropaso del motor paso a paso en la dirección seleccionada por el pin DIR. El chip tiene tres entradas diferentes para el control de sus múltiples estados de energía: RST, SLP, y EN.

- **DIR:** Dirección de giro. Según el valor que se asigne (HIGH o LOW) hará que el motor gire en un sentido u otro.

- **STEP:** Paso. Cada flanco de subida (pasar de LOW a HIGH) hace que el motor realice un paso. Si se hace que el tiempo entre flancos sea cada vez mas corto, el motor girará con mayor velocidad.

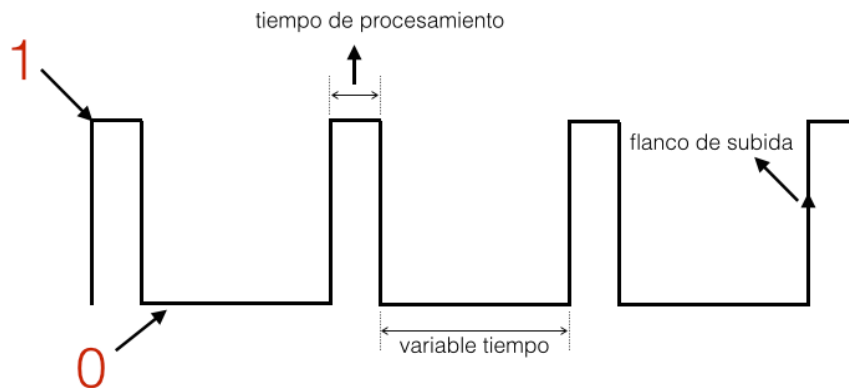


Figura 19 Esquema de la señal de STEP

Si cada flanco de subida equivale a un paso, si se reduce el tiempo entre pasos se aumenta la velocidad de giro.

- **SLEEP y RESET:** Estos pines no tienen utilidad. El fabricante indica que si éstos pines no se usan se recomienda puentearlos.

- **MS1, MS2 y MS3:** Estos pines definen la resolución del paso, permitiendo hasta un máximo de 1/16 de paso si los tres están puestos a HIGH.

- **ENABLE:** On/Off del motor. Este pin está invertido, por lo que si lo que se quiere es desmagnetizar el motor se debe colocar un HIGH.

Para lograr altas tasas de paso, la alimentación del motor es típicamente mucho más alta de lo que sería admisible sin limitación de corriente activa. Por ejemplo, nuestros motores paso a paso tiene una corriente máxima de 0,85 A (en conexión serie) y una alimentación del motor máximo de 5,6 V. El uso de un motor de este tipo con 24 V permitiría tasas de paso superiores, pero la corriente debe estar activamente limitada a menos de 0,85 A para evitar daños en el motor.

El A4988 soporta limitación activa de corriente, permitiendo mediante un potenciómetro ajustar el límite de corriente. Una forma de establecer el límite de corriente es medir el voltaje sobre el tornillo del potenciómetro y calcular el límite de la corriente resultante (las resistencias de detección actuales son  $0.05\Omega$ ). El límite de corriente se refiere a la tensión de referencia de la siguiente manera:

$$V_{medido} = \frac{I_{max}}{2.5}$$

- $V_{medido}$ : Tensión medida sobre el potenciómetro.
- $I_{max}$ : Corriente máxima a la que se quiere limitar.

Así, en éste caso, el límite de corriente es de 0,8 A y el voltaje de referencia es 0,32V.

#### 4.2.5 Fuente de alimentación

Como se señala en el esquema general de la introducción del proyecto (ver *Figura 5*), los driver son los componentes que necesitan alimentación en el sistema. Se considerará que la placa arduino permanecerá siempre conectada mediante USB al PC ya que requiere de la interacción del usuario para su funcionamiento.

La fuente de alimentación seleccionada es una MEAN WELL 24V 240W 10A IP 20 (*Figura 20*) la cual cumple con las características de los drivers A4988 y los motores seleccionados. La función de la fuente de alimentación es la de proveer a los drivers de la suficiente corriente para el movimiento de los motores.



Figura 20 Fuente de alimentación SP-320-24

La fuente de alimentación ha sido seleccionado de acuerdo a dos criterios:

- Corriente máxima de salida capaz de proporcionar.
- Tensión nominal de salida.

Por lo que para el prototipo de 6 barras, teniendo en cuenta que nuestros motores estarán limitados en corriente a 0,85 A en la configuración serie y 1,5 A en la configuración paralelo (siendo previsibles en caso de la necesidad de cambiar el tipo de configuración), se tiene que:

$$I_{\min_{Serie}} = 0,8A \cdot 6motores = 4,8A$$

$$I_{\min_{Paralelo}} = 1,7A \cdot 6motores = 9A$$

Por lo que se ha decantado por una fuente que nos pueda proporcionar al menos 10 A en el peor de los casos.

En cuanto a la tensión de salida en continua es posible ser mas flexibles, ya que los motores tienen una tensión nominal de 5,6 V pero si se limita su corriente se podrá alimentar con tensiones superiores a la nominal. Además los drivers tienen un rango de tensiones de entrada de entre 8 y 35V, por lo que se ha decantado por una fuente de tensión nominal de 24V.

La fuente que se debe seleccionar deberá contar al menos con una tensión nominal de salida de 24 V y una corriente de salida de 10A.

#### **4.2.6 Centralización de la electrónica**

La electrónica del prototipo se decidió centralizar en un mismo lugar, para facilitar su transporte y evitar daños de la misma. Se colocaron todos los componentes en el interior de una caja de derivación del tipo 921 85 de la marca Legrand. Fue bastante costoso introducir todos los componentes en la caja, debido al gran volumen de cables y componentes. La fuente de alimentación se ha colocado atornillada sobre la tapa superior de la caja (ver *Figura 22*).

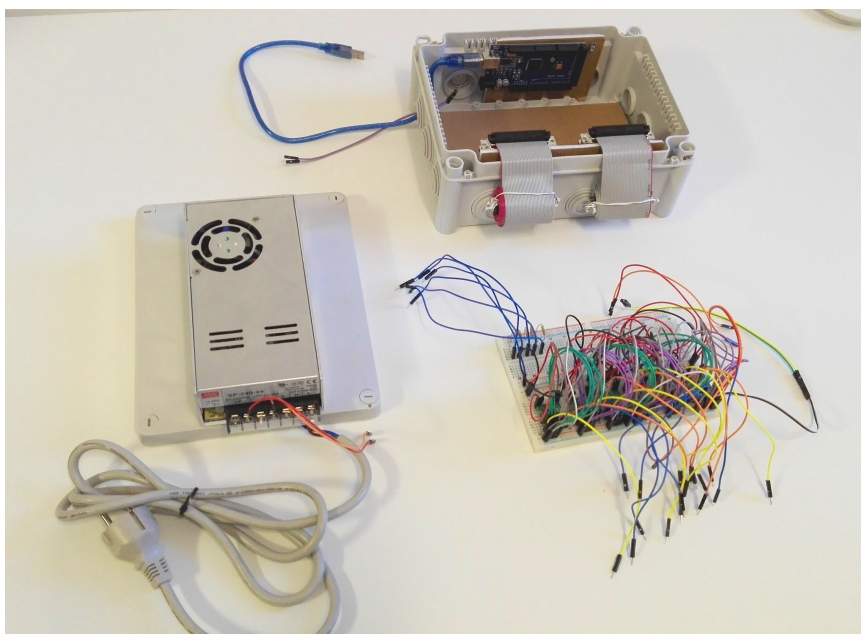


**Figura 21** Caja de registro Legrand 921 85

Las partes que quedan accesibles desde el exterior de la caja son:

- El cable USB que se conecta con el arduino para poder hacer la programación del mismo desde el ordenador.
- El conector de la toma de corriente de la fuente de alimentación para encender y apagar el sistema.
- Los dos cables planos que conectan el sistema de control con la CSUR.

Puedes ver un esquema de la centralización de la electrónica en el *ANEXO I*.



**Figura 21** Conjunto de los elementos que forman el sistema de control centralizados

## 4.2 Conexiones

En este apartado se describirá el modo en que se conectará la CSUR al sistema de control, describiendo el cableado y los conectores empleados.

### 4.2.1. Conectores

El prototipo de la CSUR llevará 2 conectores SUB-D macho de 37 pines acoplados a una bornera y se situarán atornillados en el interior. Su cometido es el del conexionado de los motores y sensores hacia el sistema de control externo.

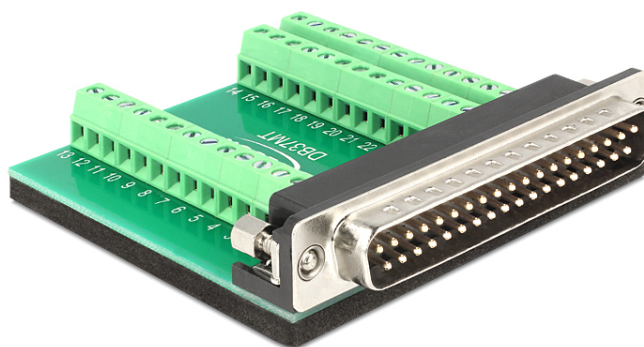


Figura 23 Conector SUB-D macho con bornera

- El **conector 1** albergará las conexiones de las 4 fases de 3 motores impares 1, 3 y 5, y las conexiones de escobillas de contacto.
- El **conector 2** albergará las conexiones de las 4 fases de 3 motores pares 2, 4 y 6, las conexiones de los sensores de home y el sensor del travesaño.

Las conexiones estarán registradas detalladamente en el *ANEXO 3*.

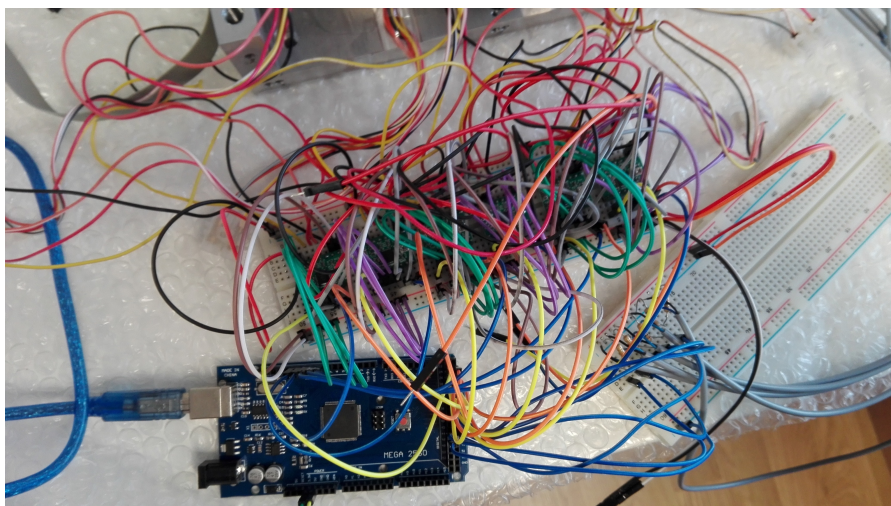
### 4.2.2. Cableado

El cableado interior de la CSUR estará únicamente formado por los conductores propios de los motores, y los conductores necesarios para cada uno de los sensores.



El cableado exterior entre CSUR y el sistema de control se realizará mediante cable plano y conectores SUB-D hembra de 37 pines. Todas las conexiones interiores entre los componentes del sistema de control se realizarán sobre placa de prototipado o protoboard mediante cables del tipo jumper y con cable USB AM/BM hacia el PC.

Ver en el *ANEXO 8* el esquema de conexiones eléctricas y en el *ANEXO 9* la relación de conexiones de pines entre arduino, drivers y sensores.



**Figura 24** Cableado de los componentes de la electrónica

## **5. DESARROLLO DEL SOFTWARE**

---

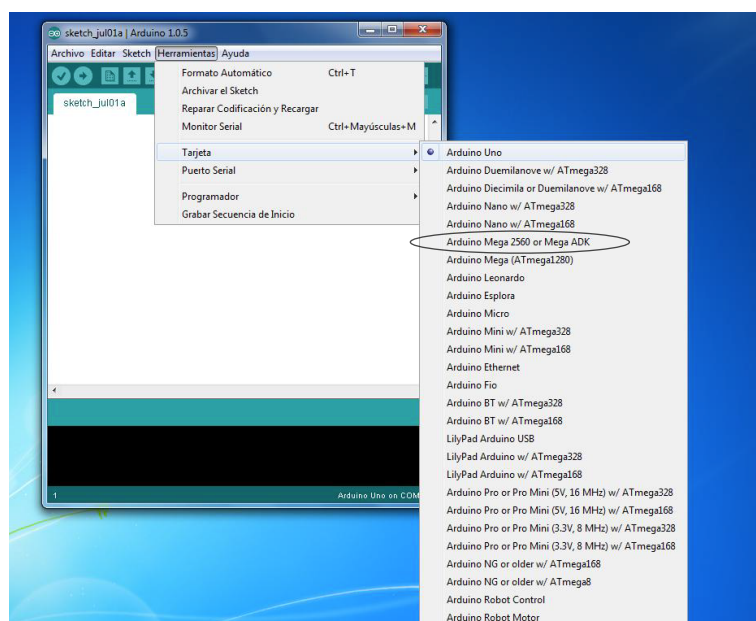
### **5.1. Lenguaje**

Como se explicó en el apartado 4.2.3, la programación de arduino está basada en Wiring [9]. Este tipo de programación permite hacer uso de un conjunto de funciones, variables y constantes que encapsulan el funcionamiento del hardware, facilitando el uso del mismo. Además, permite el uso de las características de lenguaje C++ dentro del ambiente de desarrollo, lo que permite crear funciones, punteros, clases y objetos e incluso usar el lenguaje máquina. Arduino se suele programar fundamentalmente en lenguaje C++, pero cuando se quiere contar con más utilidades, debe añadirse parte del código fuente en otros lenguajes de programación. Para este prototipo, se utilizará el lenguaje C++ para la base del programa.

## 5.2. Entorno de desarrollo

Como se comentó en el apartado 4.2.3, arduino cuenta con su propio entorno de desarrollo, el cual puede ser descargado directamente de forma gratuita desde su página web [2].

Una vez descargado e instalado, se debe configurar el entorno. En primer lugar, se debe seleccionar la tarjeta que se programará, en este caso, un arduino Mega 2560 (ver *Figura 25*).



**Figura 25** Selección de la tarjeta en el entorno de desarrollo de arduino

En segundo lugar, se debe seleccionar el puerto serial en el que se encuentra conectado el arduino UNO (ver *Figura 26*).

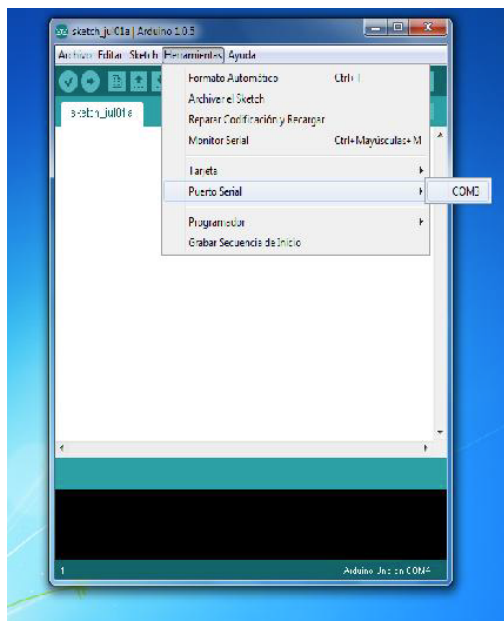


Figura 26 Selección del puerto serial

Y por último, se debe seleccionar el programador, que en este caso será el AVRISP mkII (ver Figura 27).

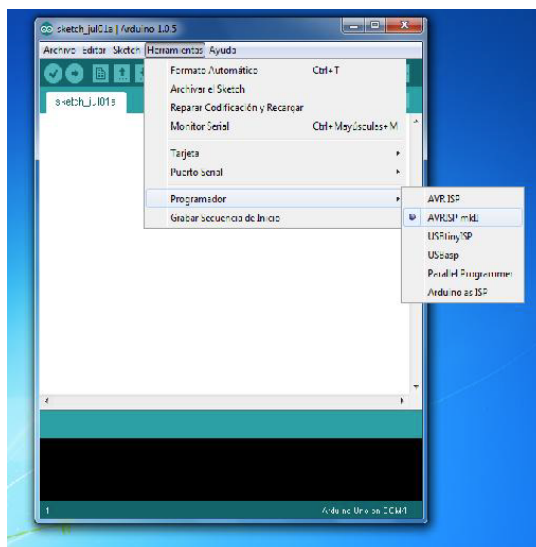


Figura 27 Selección del programador

## 5.3. Código de programación

Una vez el entorno se encontraba configurado, se pasó a la programación del código. A medida que se fue programando, se le fueron añadiendo comentarios para que se entendiera qué realizaba cada instrucción en futuras modificaciones [3].

### 5.3.1. Funciones

El programa desarrollado cuenta con distintas funciones para determinadas acciones. En el *ANEXO 7* se describe el funcionamiento global del programa:

- *setup*: configura los puertos de la tarjeta arduino como entradas y/o salidas. Sólo se ejecuta una vez al cargar el programa.

```
void setup() {  
  
  Serial.begin(38400);  
  
  pinMode(12, OUTPUT); //PWM de STEP motor 1  
  pinMode(22, OUTPUT); //ENABLE motor 1  
  pinMode(23, OUTPUT); //DIR direccion de giro. HIGH derecha (hacia home) y LOW izquierda  
  
  pinMode(11, OUTPUT); //PWM de STEP Motor 3  
  pinMode(24, OUTPUT); //ENABLE Motor 3  
  pinMode(25, OUTPUT); //DIR direccion de giro. HIGH izquierda (hacia home) y LOW derecha  
  
  pinMode(10, OUTPUT); //PWM de STEP Motor 5  
  pinMode(26, OUTPUT); //ENABLE Motor 5  
  pinMode(27, OUTPUT); //DIR direccion de giro. HIGH derecha (hacia home) y LOW izquierda  
  
  pinMode(9, OUTPUT); //PWM de STEP Motor 2  
  pinMode(28, OUTPUT); //ENABLE Motor 2  
  pinMode(29, OUTPUT); //DIR direccion de giro. HIGH izquierda (hacia home) y LOW derecha  
  
  pinMode(8, OUTPUT); //PWM de STEP Motor 4  
  pinMode(30, OUTPUT); //ENABLE Motor 4  
  pinMode(31, OUTPUT); //DIR direccion de giro. HIGH izquierda (hacia home) y LOW derecha  
  
  pinMode(7, OUTPUT); //PWM de STEP Motor 2  
  pinMode(32, OUTPUT); //ENABLE Motor 2
```

```

pinMode(33, OUTPUT); //DIR direccion de giro. HIGH izquierda (hacia home) y LOW derecha

//Sensores:

pinMode(35, INPUT); //Entrada final de carrera motor 1
pinMode(36, INPUT); //Entrada contacto barras motor 1-2

pinMode(42, INPUT); //Entrada final de carrera Motor 5
pinMode(38, INPUT); //Entrada contacto barras Motor 5-4

pinMode(39, INPUT); //Entrada final de carrera Motor 4
pinMode(40, INPUT); //Entrada contacto barras Motor 4-6

pinMode(41, INPUT); //Entrada travesaño

pinMode(37, INPUT); //Entrada final de carrera Motor 3
pinMode(43, INPUT); //Entrada final de carrera Motor 2
pinMode(44, INPUT); //Entrada final de carrera motor 6

digitalWriteFast(22,HIGH); //Enable OFF motor 1
digitalWriteFast(12,LOW); //Motor OFF motor 1

digitalWriteFast(24,HIGH); //Enable OFF Motor 3
digitalWriteFast(11,LOW); //Motor OFF Motor 3

digitalWriteFast(26,HIGH); //Enable OFF Motor 5
digitalWriteFast(10,LOW); //Motor OFF Motor 5

digitalWriteFast(28,HIGH); //Enable OFF Motor 2
digitalWriteFast(9,LOW); //Motor OFF Motor 2

digitalWriteFast(30,HIGH); //Enable OFF Motor 4
digitalWriteFast(8,LOW); //Motor OFF Motor 4

digitalWriteFast(32,HIGH); //Enable OFF Motor 4
digitalWriteFast(7,LOW); //Motor OFF Motor 4

arranque();
}

```

- arranque: mueve todas las barras hacia su posición de home. Se ejecuta inmediatamente al cargar el programa
- una barra: permite mover en cualquier sentido la barra que se desee

- *loop*: función donde se mueven el conjunto de 6 barras desde sus posiciones de home, siempre en sentido hacia el campo.
- *rendija*: se ejecuta tras finalizar el movimiento del conjunto de 6 barras. Abre una rendija del tamaño deseado.
- *menu 1*: menu con 4 opciones
  - Cerrar Campo: cierra el campo
  - Mover 6 barras: mueve las 6 barras cada una a una posición deseada
  - Mover una barra: mueve una barra hacia una posición y en la dirección deseada

```
void menu1(){
ruido1=0, ruido2=0, ruido3=0, ruido4=0, ruido5=0, ruido6=0, ruidotrav=0,
ruidocon12=0,ruidocon34=0,ruidocon56=0;

menu1_v=0;
Serial.println(" ");
Serial.println("Introduzca un numero para saber que quiere hacer ahora:");
Serial.println("1.- Cerrar por completo el campo ");
Serial.println("2.- Elegir una posición para el conjunto de las 6 barras ");
Serial.println("3.- Elegir una posición para una barra concreta ");
Serial.println("4.- Imprimir la posición de las barras");
Serial.read();

while(e==1){

if(Serial.available(>0){
menu1_v=Serial.parseInt();
Serial.println(" ");
Serial.print("Su elección ha sido: ");
Serial.println(menu1_v);
e=0;
}
}

switch(menu1_v){

case 1:

cerrar();

break;
```

```
case 2:

    loop();

break;

case 3:

    Serial.print("Elegir la barra deseada: 1-6");
    Serial.read();

    while(e==0){

        if(Serial.available(>0){
            menu1_v=Serial.parseInt();
            Serial.println(" ");

            Serial.println(menu1_v);
            e=1;
        }
    }

    una_barra();

break;

case 4:

    Serial.println(" ");
    Serial.println("La posición de la barra 1 es:");
    Serial.println(posicion1,3);

    Serial.println(" ");
    Serial.println("La posición de la barra 2 es:");
    Serial.println(posicion2,3);

    Serial.println(" ");
    Serial.println("La posición de la barra 3 es:");
    Serial.println(posicion3,3);

    Serial.println(" ");
    Serial.println("La posición de la barra 4 es:");
    Serial.println(posicion4,3);

    Serial.println(" ");
    Serial.println("La posición de la barra 5 es:");
```

```
Serial.println(posicion5,3);

Serial.println(" ");
Serial.println("La posición de la barra 6 es:");
Serial.println(posicion6,3);

break;

default:
    Serial.read();
    Serial.println(" ");
    Serial.print("Introduzca un valor válido");
    break;

}
e=1;
delay(3700);
menu1();
}
```

- Imprimir posiciones: imprime la posición en la que se encuentran las barras

- menu\_2: menu con 3 opciones
  - Llevar todas las barras a home
  - Mover una barra.
  - Imprimir posiciones.

```
void menu2(){

ruido1=0,    ruido2=0,    ruido3=0,    ruido4=0,    ruido5=0,    ruido6=0,    ruidootrav=0,
ruidocon12=0,ruidocon34=0,ruidocon56=0;

menu2_v=0;
Serial.println(" ");
Serial.println("Introduzca un numero para saber que quiere hacer ahora:");
Serial.println("1.- Llevar las barras a su posición de Home");
Serial.println("2.- Imprimir la posición de las barras");
Serial.println("3.- Elegir una posición para una barra");
Serial.read();

while(e==1){
```



```
    if (Serial.available() > 0) {
        menu2_v = Serial.parseInt();
        Serial.println(" ");
        Serial.print("Su elección ha sido: ");
        Serial.println(menu2_v);
        e = 0;
    }
}

switch (menu2_v) {

    case 1:
        arranque();
        break;

    case 2:

        Serial.println(" ");
        Serial.println("La posición de la barra 1 es:");
        Serial.println(posicion1, 3);

        Serial.println(" ");
        Serial.println("La posición de la barra 2 es:");
        Serial.println(posicion2, 3);

        Serial.println(" ");
        Serial.println("La posición de la barra 3 es:");
        Serial.println(posicion3, 3);

        Serial.println(" ");
        Serial.println("La posición de la barra 4 es:");
        Serial.println(posicion4, 3);

        Serial.println(" ");
        Serial.println("La posición de la barra 5 es:");
        Serial.println(posicion5, 3);

        Serial.println(" ");
        Serial.println("La posición de la barra 6 es:");
        Serial.println(posicion6, 3);

        break;

    case 3:
```

```
Serial.print("Elegir la barra deseada: 1-6");
Serial.read();

while(e==0){

  if(Serial.available(>0){
    menu1_v=Serial.parseInt();
    Serial.println(" ");

    Serial.println(menu1_v);
    e=1;
  }
}

una_barra();

break;

default:
  Serial.read();
  Serial.println(" ");
  Serial.print("Introduzca un valor válido");
  break;

}
e=1;
delay(4000);
menu2();
}
```

- cerrar: cierra el campo por completo. El campo no se cerrará en el centro geométrico del campo, se cerrará en los extremos de la superficie útil de espectroscopía tal y como se representa en la *Figura 28*.



**Figura 28** Representación del cierre del campo

- *travesaño*: esta función se ejecutará cuando se detecte que alguna barra ha tocado el travesaño. Parará el movimiento de todas las barras y las hará regresar a su posición de home.

Para el cálculo de las posiciones se incrementa una variable en cada flanco de subida de la señal de STEP (ver apartado 4.2.4) que equivale al número de pasos dados por el motor. Al conocer que el paso de los husillos utilizados es de 2 mm y el número de pasos necesarios para realizar una vuelta (ver apartado 8.4) con la resolución seleccionada (ver apartado 4.2.4).

$$posición = \frac{pasos}{3200} \cdot 2 [mm]$$

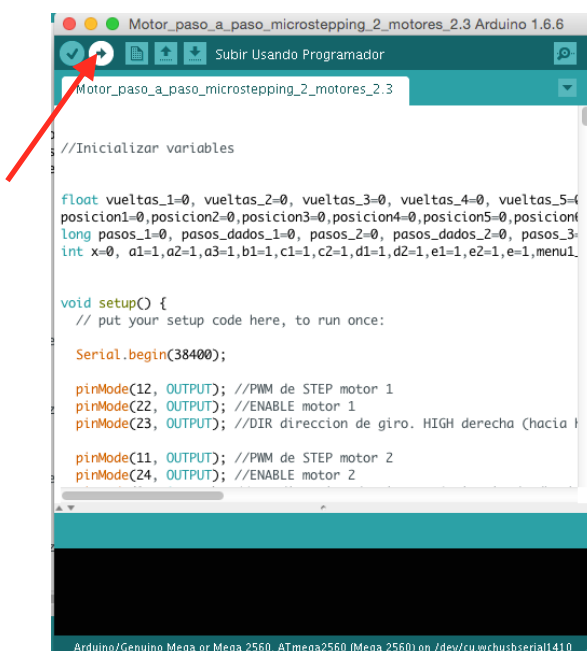
## 5.4 Carga del programa

Una vez se terminó de diseñar el código, fue cargado en arduino. Para ello es fundamental que el arduino Mega 2560 esté conectado al ordenador mediante un cable USB y que el entorno de desarrollo esté configurado como se indicaba en el apartado 4.2. Una vez comprobado esto, se carga el programa en la tarjeta (ver figura 5.4). Al seleccionar la opción de cargar, el programa compilará el código, y si se detecta algún fallo de sintaxis, lo mostrará

mediante un mensaje en la parte inferior del entorno de desarrollo. Si se carga sin errores, arduino estará listo para la ejecución del programa.

Para la puesta en marcha:

1º.- Se carga el programa en la tarjeta:



2º.- Una vez se ha cargado satisfactoriamente, se abre el monitor serie:



Los motores empezarán a girar si no se encuentran en home ya que la función arranque se pondrá en marcha nada mas cargar el programa.

3º.- El monitor serie imprimirá un mensaje si las barras han llegado a home.

4º.- El monitor serie imprimirá un menu con 4 opciones (menu\_1), a partir del cuál se podrá empezar a interactuar con él.

## 6. PRUEBAS REALIZADAS

---

Tras un diseño provisional se han realizado pruebas para conocer el comportamiento del sistema mecánico y los sensores, para luego realizar las correspondientes mejoras:

- Comprobar que el sensor de travesaño hace un buen contacto con las pletinas metálicas de las barras.
- Comprobar que el sistema detiene lo suficientemente rápido el movimiento de las barras al contactar con el travesaño, evitando el riesgo a esfuerzos mecánicos indeseados.
- Comprobar que el sistema detiene lo suficientemente rápido el movimiento de las barras al detectar los sensores de home, evitando el riesgo a esfuerzos mecánicos indeseados.
- Comprobar que el sistema detiene lo suficientemente rápido el movimiento de las barras hagan contacto entre ellas, evitando el riesgo a esfuerzos mecánicos indeseados.
- Comprobar que el sistema permite alcanzar una velocidad mínima para lograr reconfigurar las barras en un máximo de 5 minutos (ver apartado 3.5).
- Comprobar que cada una de las funciones programadas se realizan correctamente.
- Comprobar que la interfaz es intuitiva y comprensible para un usuario sin experiencia.

## 7. PROBLEMAS Y SOLUCIONES

---

Durante el procedimiento de prueba del diseño provisional surgieron algunos problemas imprevistos que no se habían tenido en cuenta.

1. El sistema se quería diseñar para que los motores pudieran desarrollar distintas velocidades a elección del usuario, por lo que una señal PWM para controlar la señal de STEP no permitiría esta opción. Esto se debe a que, el avance de cada paso de los motores se produce en cada flanco de subida de la señal STEP, por lo que una señal PWM solo permite

variar el ciclo de trabajo de la señal (el ancho del pulso durante el tiempo que está en HIGH) y no el tiempo entre cada flanco de subida.

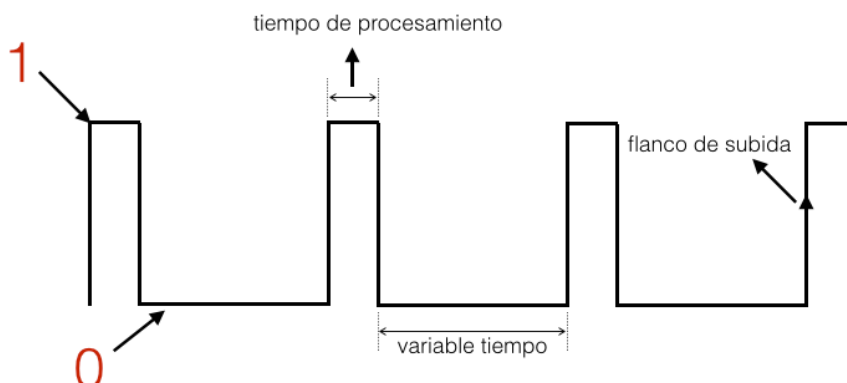


Figura 19 Esquema de la señal de STEP

Es por esto que se decidió crear una señal PWM forzada por una variable temporal a elección por el usuario para regular el tiempo entre pasos y por tanto la velocidad de los motores.

```
temp=1;
while(temp==1){
    digitalWrite(12,HIGH);
    digitalWrite(12,LOW);
    .
    .
    .
    delayMicroseconds(tiempo);
}
```

2. Una consecuencia del método que se ha utilizado para la señal STEP es la aparición de un tiempo de procesamiento que limita la velocidad de giro de los motores (ver Figura 19), quiere decir que si el tiempo introducido por el usuario fuese de 0, aún existiría el tiempo de procesamiento entre flancos de subida.

Ya que el tiempo de procesamiento es inevitable, lo que se ha hecho es minimizarlo lo máximo posible mediante una librería que realiza la operación de escritura de los pines en

menor tiempo (`digitalWriteFast`) [8] y optimizar los procesos que se ejecutan al mismo tiempo.

```
#include <digitalWriteFast.h>
temp=1;
while(temp==1){
    digitalWriteFast(12,HIGH);
    digitalWriteFast(12,LOW);
    .
    .
    .
    delayMicroseconds(tiempo);
}
```

3. A la hora de realizar algunas de las funciones de movimiento de las barras se producía una falsa detección de los distintos sensores, llegando a la conclusión de que era causa de un ruido electrónico por el acoplamiento entre los conductores de alimentación de los motores y los conductores de los sensores en los conectores SUB-D. Por lo que se procedió, como primera medida de corrección, a separar físicamente los conductores en lo máximo posible y conectar los pines desocupados a GND.

Ésta medida no terminó por solucionar el problema por lo que se intentó solucionarlo mediante software con un sistema antirebote. Esto es, el programa no tomará la primera lectura de los sensores como la absoluta sino que se realizará una segunda lectura en el siguiente ciclo de procesamiento. Si la primera lectura es igual a la segunda, se tomará ésta como la lectura absoluta, en cambio si son distintas querrá decir que la lectura no es más que ruido.

```
#include <digitalWriteFast.h>
temp=1;
while(temp==1){
    digitalWriteFast(12,HIGH);
    digitalWriteFast(12,LOW);

    if(digitalRead(35)==HIGH){
        ruido=ruido + 1;
        if(digitalRead(35)==HIGH && ruido >=2){
            digitalWrite(12, LOW); //Motor 1 OFF
            Serial.println(" ");
            Serial.println("La barra 1 ha llegado a HOME");
            ruido=0;
        }
    }
    delayMicroseconds(tiempo);
}
```

El sistema antirebote solucionó por completo el problema del ruido.



## 8. CÁLCULOS

### 8.1. Pasos por vuelta

Considerando una resolución de 1/16 paso y teniendo en cuenta que el paso del motor es de  $1'8^\circ$ , por tanto 200 pasos por vuelta, entonces con una resolución de 1/16 de paso el motor dará 3200 micropasos por vuelta.

$$\left. \begin{array}{l} 200 \text{ pasos} \rightarrow 1'8^\circ \\ x \rightarrow \frac{1'8^\circ}{16} \end{array} \right\} 200 \text{ pasos} \cdot 16 = 3200 \text{ micropasos/vuelta}$$

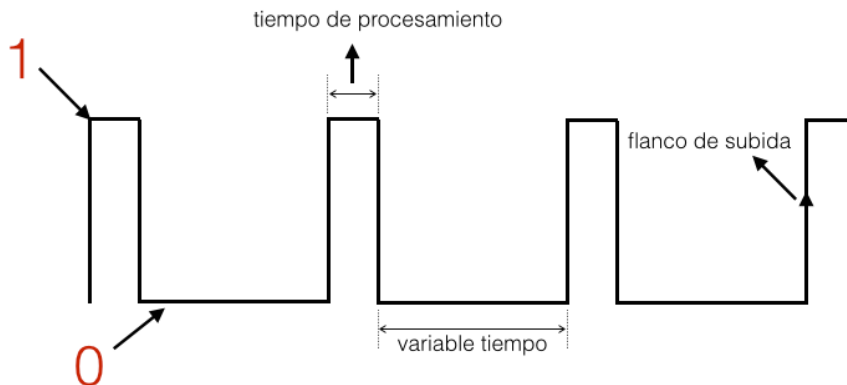
Se tiene que para una vuelta los motores necesitarán dar 3200 micropasos.

### 8.2. Velocidad de giro

Si se tiene en cuenta que en cada flanco de subida de la señal de STEP se realiza un paso, y entre cada flanco de subida transcurre una variable de tiempo determinado que se denominará  $t_{paso}$ :

$$t_{paso} = t_{proc} + t_{variable}$$

- $t_{variable}$ : Es una variable en nuestro programa con la que se varía la velocidad de giro de nuestros motores. Al ser menor esta velocidad menor es el tiempo entre pasos por lo que mayor es la velocidad.
- $t_{proc}$ : Es el tiempo de procesamiento del propio software. Es un tiempo fijo y el cuál se debe tener en cuenta a la hora del cálculo de la velocidad de giro.



El tiempo de procesamiento ha sido calculado experimentalmente. Éste tiempo se ha medido contando el tiempo transcurrido entre el inicio y final de un ciclo, obteniendo un valor promedio de unos 46µs y por lo tanto:

$$t_{paso} = 46\mu s + t_{variable}$$

Finalmente la velocidad de giro se obtiene de la relación de los pasos dados por vuelta y el tiempo empleado en ello. Por lo tanto si se multiplica los pasos por vuelta (3200 micropasos) por el tiempo por paso ( $t_{paso}$ ) se obtiene el tiempo que emplean los motores en dar una vuelta, y a la inversa las vueltas que dan por unidad de tiempo (rps)

$$v_{giro} = \frac{1}{3200 \cdot (t_{paso} \cdot 10^{-6})} [rps]$$

### 8.3. Velocidad de avance horizontal

La velocidad de avance horizontal de una barra se calcula a partir de la velocidad de giro del motor y el paso de los husillos. Al ser el paso de los husillos de 2mm, la velocidad de avance se obtiene al multiplicar la velocidad de giro por el paso del husillo:

$$v_{avance} = v_{giro} \cdot 2 \left[ \frac{mm}{s} \right] = v_{giro} \cdot 0.002 \left[ \frac{m}{s} \right]$$

### 8.4. Tiempo en llegar a una posición concreta

Dada una posición de destino, el tiempo en alcanzar dicha posición se calcula dividiendo la posición deseada por la velocidad de avance:

$$t_{avance} = \frac{posición}{v_{avance}} [s]$$

## 8.5. Posición de una barra

Como el sistema es un sistema en lazo abierto, la posición de las barras se podrá conocer únicamente contando los pasos dados por el motor y por tanto, las vueltas dadas. Se asigna esa cuenta de vueltas dadas a una variable y conociendo el paso de los husillos, la posición relativa de la barra es:

$$\text{posición} = \text{vueltas} \cdot 2[\text{mm}]$$

*NOTA: La posición es la distancia desde extremo de la barra hasta la posición de home.*

## 9. PRESUPUESTO MATERIAL

Artículo	Precio (€)	Unidades	Total (€)
<b>Sistema de control</b>			
Arduino Mega 2560 + cable USB	15,00	1	15,00
Driver A4988	2,00	6	12,00
Placas protoboard	3,60	2	7,20
Fuente de alimentación 24V 10A / 240W	65,00	1	65,00
Caja de registro Legrand	19,97	1	19,97
SUBTOTAL			119,17
<b>CSUR</b>			
Sensores Home MY-COM H75/80	112,06	6	672,36
Conectores SUB-D macho de 37 pines	10,38	2	20,76
Motores paso a paso Applied Motion HT17-275	58,38	6	350,28
SUBTOTAL			1043,04
<b>TOTAL</b>			<b>1162,57</b>

## 10. ABSTRACT AND FUTURE WORK

### 10.1 Abstract

The main objective of this project was to design a system for the control bars of a CSUR prototype. The CSUR moves the bars and according to the position in which they are, block or unblock the nominal field of view, allowing the light beam reach or not the detector. Therefore, the design was focused on engines, how to connect and control. The engines were selected were stepper motors, controlled by the Arduino open source platform. Although Arduino did not have the capacity to give the enough current directly to the engine, because they demanded more current than Arduino could produce, it could be solved through the use of drivers A4988 which have been connect with a power supply.

As a final result, the assembled prototype is able to perform all the functions required, including the formation of slits with a width desired and positioning of the bars with a resolution of up to 1  $\mu\text{m}$ . Also the ability to bring to home position all the bars, to move the set of 6 bars or even move a single bar to a desired position.

After conducting tests, the following results were obtained:

<i>Check the crossbar sensor makes good contact with the metal plates of the bars.</i>	It does not occur a good contact and it proceeded to cut the film of insulating material Isolating crossbar frame, achieving the desired contact.
<i>Check that the system stops moving fast enough bars to contact the crossbar, avoiding the risk of unwanted mechanical stresses.</i>	The test was positive
<i>Check system fast enough stops the rod movement sensors to detect the home, avoiding the risk to undesired mechanical stresses.</i>	After several tests that the system does not stop the movement of the bars once had reached its physical limit, it was due calibrate gauges sensors home.
<i>Test system fast enough stops the movement of the bars making contact between them, avoiding the risk to undesired mechanical stresses.</i>	The test was positive
<i>Check that the system allows a minimum speed to achieve reconfigure the bars in a maximum of 5 minutes (see section 3.5).</i>	The test was positive

<i>Check that each of the programmed functions are performed correctly.</i>	After the respective modifications to the initial system was achieved that all functions work without problems
<i>Check that the interface is intuitive and understandable to an inexperienced user.</i>	After a short explanation of the interface is intuitive and understandable

In this project they have been applied knowledge acquired in numerous subjects of Grade in Electronic Industrial Engineering and Automation. Some of them have been present more directly, such as Computing, Electronic Engineering Fundamentals, Fundamentals of Electrical Engineering, Digital Electronics, and Industrial Automation. In addition, they had to acquire knowledge that were not present in the curriculum Grade, such as using and programming Arduino and their complements.

## 10.2 Future works

The control system designed will be able to test the 6 bars prototype, allowing to observe the behavior of all mechanical components. Thanks to this it can set for optimum working and maximum performance.

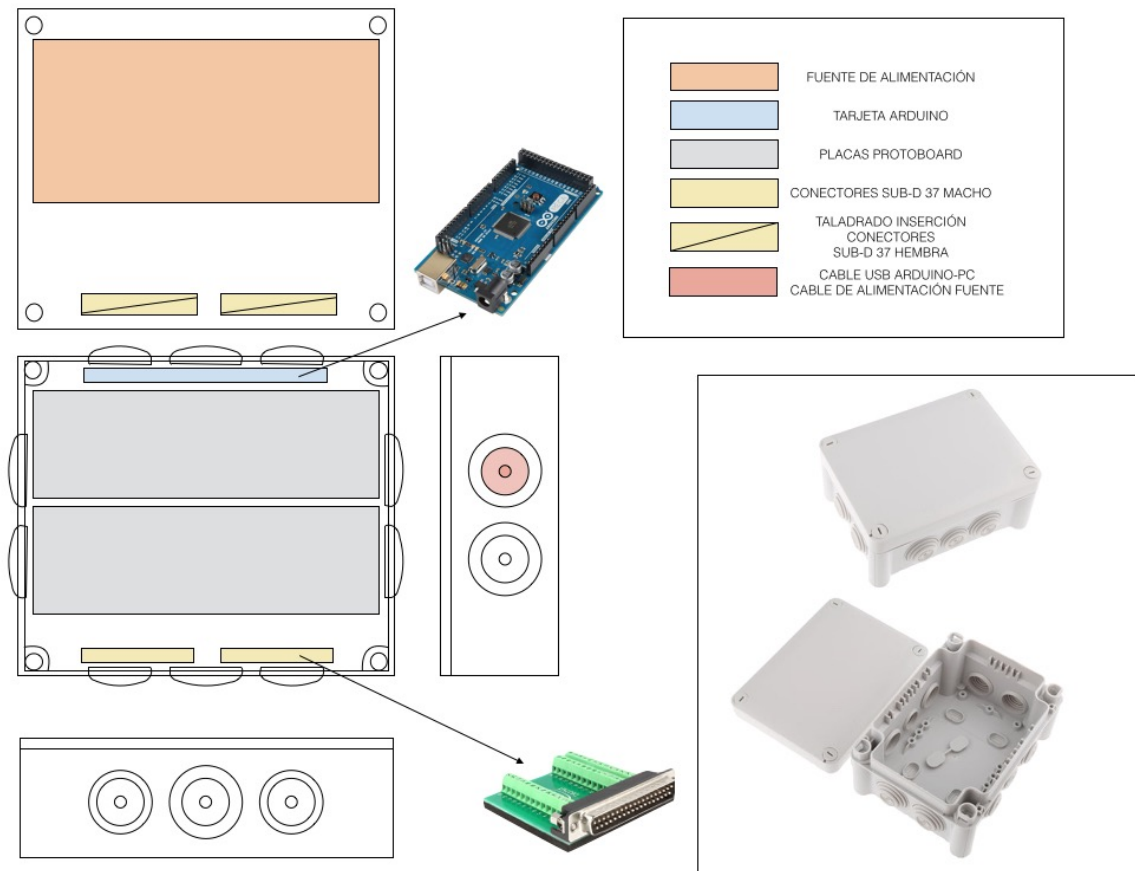
The control system is easily adaptable to give, if it is desired, a different use, always within the framework of motor control.

## 11. REFERENCIAS

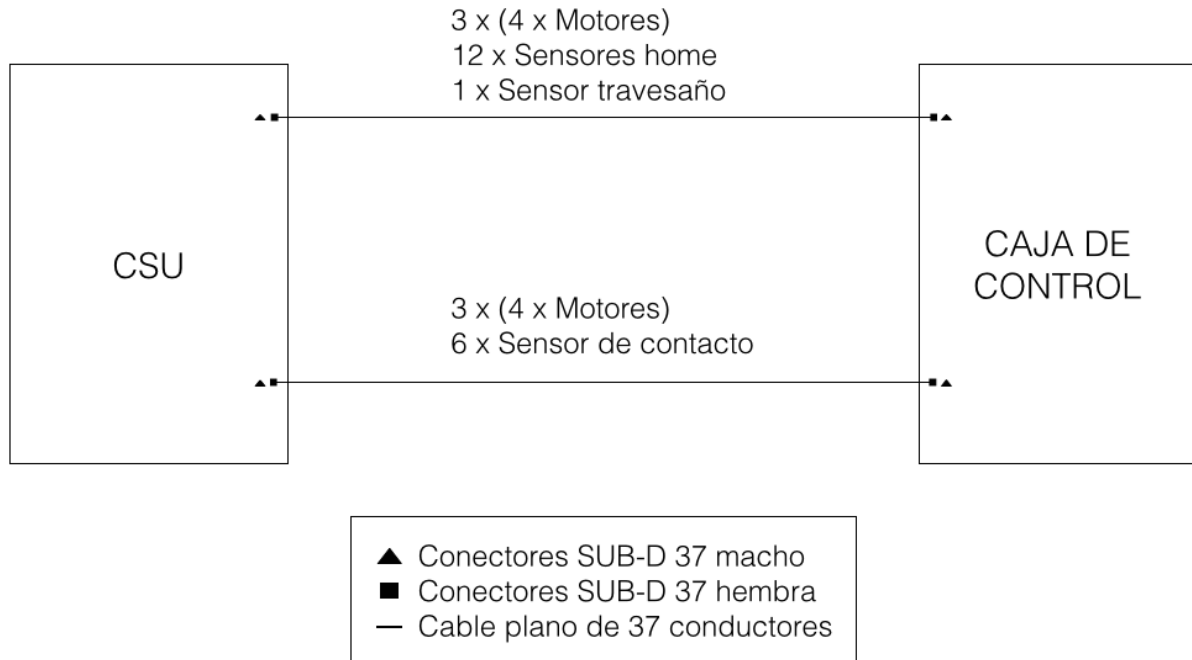
---

- [1] Información de Arduino. [www.arduino.cc/es](http://www.arduino.cc/es)
- [2] Arduino website. [www.arduino.cc/Main/Software](http://www.arduino.cc/Main/Software)
- [3] Programa y librería. <https://www.dropbox.com/sh/tz6hd7qv9h0mbfp/AAD-QPayEP1SPxIFTna4s-sFa?dl=0>
- [4] Documento oficial de especificaciones del EMIR. [https://www.dropbox.com/s/cizl855ehcb5sh8/Ref.%201%20DMSREMI\\_577v4.pdf?dl=0](https://www.dropbox.com/s/cizl855ehcb5sh8/Ref.%201%20DMSREMI_577v4.pdf?dl=0)
- [5] Página web del IAC. <http://www.iac.es/>
- [6] Información sobre EMIR. <http://www.gtc.iac.es/instruments/emir/emir.php>
- [7] Información sobre espectrógrafo. [http://www.ecured.cu/Espectr%C3%B3grafo#Componentes\\_del\\_Espectr%C3%B3grafo](http://www.ecured.cu/Espectr%C3%B3grafo#Componentes_del_Espectr%C3%B3grafo)
- [8] Librería digitalWriteFast. <https://github.com/watterott/Arduino-Libs/blob/master/digitalWriteFast/digitalWriteFast.h>
- [9] Información sobre el lenguaje C++/Wiring. <http://wiring.org.co/learning/tutorials/es/C++/>

## 12. ANEXO 1: Esquema de centralización de la electrónica



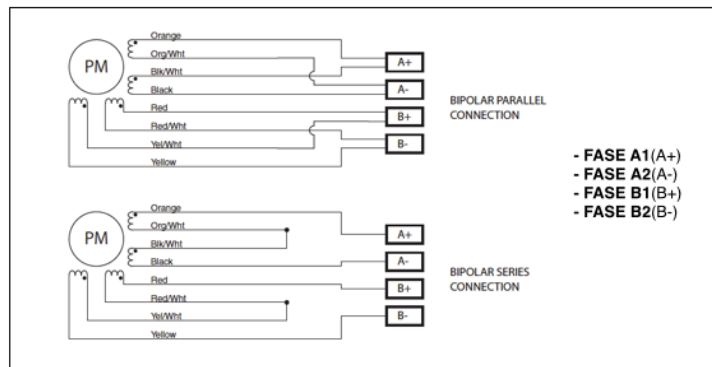
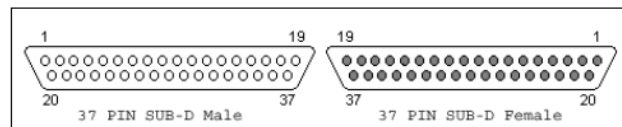
## 13. ANEXO 2: Esquema de conexiones CSUR-Caja



## 14. ANEXO 3: Disposición de conexiones en los conectores

Conector 1	
1	BARRA 01 MOTOR FASE A1
2	BARRA 01 MOTOR FASE A2
3	BARRA 01 MOTOR FASE B1
4	BARRA 01 MOTOR FASE B2
5	BARRA 03 MOTOR FASE A1
6	BARRA 03 MOTOR FASE A2
7	BARRA 03 MOTOR FASE B1
8	BARRA 03 MOTOR FASE B2
9	BARRA 05 MOTOR FASE A1
10	BARRA 05 MOTOR FASE A2
11	BARRA 05 MOTOR FASE B1
12	BARRA 05 MOTOR FASE B2
13	BARRA 01 ESCOBILLA
14	BARRA 02 ESCOBILLA
15	BARRA 03 ESCOBILLA
16	BARRA 04 ESCOBILLA
17	BARRA 05 ESCOBILLA
18	BARRA 06 ESCOBILLA
19	-
20	-
21	-
22	-
23	-
24	-
25	-
26	-
27	-
28	-
29	-
30	-
31	-
32	-
33	-
34	-
35	-
36	-
37	-

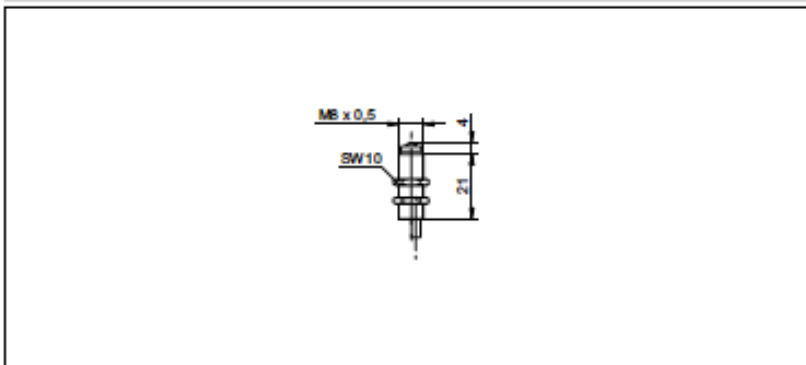
Conector 2	
1	BARRA 02 MOTOR FASE A1
2	BARRA 02 MOTOR FASE A2
3	BARRA 02 MOTOR FASE B1
4	BARRA 02 MOTOR FASE B2
5	BARRA 04 MOTOR FASE A1
6	BARRA 04 MOTOR FASE A2
7	BARRA 04 MOTOR FASE B1
8	BARRA 04 MOTOR FASE B2
9	BARRA 06 MOTOR FASE A1
10	BARRA 06 MOTOR FASE A2
11	BARRA 06 MOTOR FASE B1
12	BARRA 06 MOTOR FASE B2
13	BARRA 01 SENSOR HOME 1
14	BARRA 01 SENSOR HOME 2
15	BARRA 02 SENSOR HOME 1
16	BARRA 02 SENSOR HOME 2
17	BARRA 03 SENSOR HOME 1
18	BARRA 03 SENSOR HOME 2
19	BARRA 04 SENSOR HOME 1
20	BARRA 04 SENSOR HOME 2
21	BARRA 05 SENSOR HOME 1
22	BARRA 05 SENSOR HOME 2
23	BARRA 06 SENSOR HOME 1
24	BARRA 06 SENSOR HOME 2
25	TRAVESAÑO 1
26	-
27	-
28	-
29	-
30	-
31	-
32	-
33	-
34	-
35	-
36	-
37	-





## 15. ANEXO 4: Hoja de datos sensores de home MY-COM H75/80

### dimension drawing



### general data

repeat accuracy	< 0,001 mm
activating force	75 cN
mech. pre-run / overrun	- / 0,6 mm approx.
measurement type	contact with medium
direction of approach	frontal

### electrical data

DC voltage max.	15 VDC
switch. current DC max.	2 mA
AC voltage max.	24 VAC
switch. current AC max.	50 mA
output circuit	break function (NC) mechanical

### mechanical data

activating pin	ruby
housing material	brass nickel plated
dimension	8 mm
type	cylindrical threaded
housing length	21 mm
connection types	cable, 80 cm

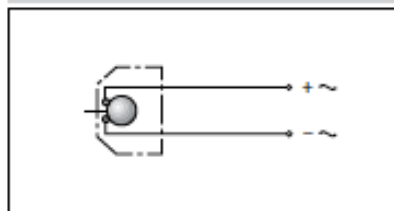
### ambient conditions

operating temperature	-20 ... +75 °C
protection class	IP 67

### photo



### connection diagram



## 16. ANEXO 5: Hoja de datos de la fuente de alimentación RSP-320-24



### ■ Features :

- Universal AC input / Full range
- Built-in active PFC function, PF>0.95
- High efficiency up to 90%
- Protections: Short circuit / Overload / Over voltage / Over temperature
- Forced air cooling by built-in DC Fan with fan speed control function
- 1U low profile 30mm
- Optional conformal coating models (RSP-320-□CC)
- LED indicator for power on
- 3 years warranty

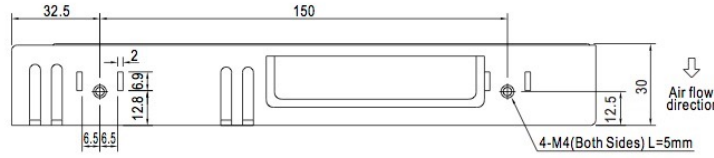
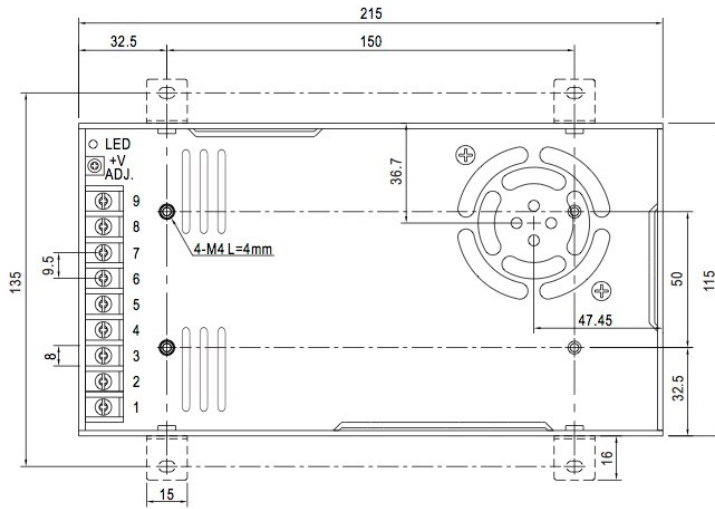


### SPECIFICATION

MODEL	RSP-320-13.5	RSP-320-15	RSP-320-24	RSP-320-27	RSP-320-36	RSP-320-48	
OUTPUT	DC VOLTAGE	13.5V	15V	24V	27V	36V	48V
	RATED CURRENT	23.8A	21.4A	13.4A	11.9A	8.9A	6.7A
	CURRENT RANGE	0 ~ 23.8A	0 ~ 21.4A	0 ~ 13.4A	0 ~ 11.9A	0 ~ 8.9A	0 ~ 6.7A
	RATED POWER	321.3W	321W	321.6W	321.3W	320.4W	321.6W
	RIPPLE & NOISE (max.) Note.2	150mVp-p	150mVp-p	150mVp-p	200mVp-p	220mVp-p	240mVp-p
	VOLTAGE ADJ. RANGE	12 ~ 15V	13.5 ~ 18V	20 ~ 26.4V	26 ~ 31.5V	32.4 ~ 39.6V	41 ~ 56V
	VOLTAGE TOLERANCE Note.3	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%	±1.0%
	LINE REGULATION	±0.3%	±0.3%	±0.2%	±0.2%	±0.2%	±0.2%
	LOAD REGULATION	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%	±0.5%
	SETUP, RISE TIME	1500ms, 50ms/230VAC 3000ms, 50ms/115VAC at full load					
HOLD UP TIME (Typ.)	8ms at full load 230VAC /115VAC						
INPUT	VOLTAGE RANGE Note.4	88 ~ 264VAC	124 ~ 370VDC				
	FREQUENCY RANGE	47 ~ 63Hz					
	POWER FACTOR (Typ.)	PF>0.95/230VAC		PF>0.98/115VAC at full load			
	EFFICIENCY (Typ.)	88%	88.5%	89%	89%	89.5%	90%
	AC CURRENT (Typ.)	4A/115VAC	2A/230VAC				
	INRUSH CURRENT (Typ.)	20A/115VAC	40A/230VAC				
LEAKAGE CURRENT	<1mA / 240VAC						
PROTECTION	OVERLOAD	105 ~ 135% rated output power Protection type : Hiccup mode, recovers automatically after fault condition is removed					
	OVER VOLTAGE	15.7 ~ 18.4V	18.8 ~ 21.8V	27.6 ~ 32.4V	32.9 ~ 38.3V	41.4 ~ 48.6V	58.4 ~ 68V
	OVER TEMPERATURE	Shut down o/p voltage, recovers automatically after temperature goes down					
ENVIRONMENT	WORKING TEMP.	-30 ~ +70°C (Refer to "Derating Curve")					
	WORKING HUMIDITY	20 ~ 90% RH non-condensing					
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH					
	TEMP. COEFFICIENT	±0.03%/°C (0 ~ 50°C)					
VIBRATION	10 ~ 500Hz, 2G 10min./1cycle, 60min. each along X, Y, Z axes						
SAFETY & EMC (Note 5)	SAFETY STANDARDS	UL60950-1, TUV EN60950-1, CCC GB4943 approved					
	WITHSTAND VOLTAGE	I/P-O/P:3KVAC I/P-FG:2KVAC O/P-FG:0.5KVAC					
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms / 500VDC / 25°C / 70% RH					
	EMC EMISSION	Compliance to EN55022 (CISPR22) Class B, EN61000-3-2,-3, GB9254 class B, GB17625.1					
EMC IMMUNITY	Compliance to EN61000-4-2,3,4,5,6,8,11, EN55024, light industry level, criteria A						
OTHERS	MTBF	206.5K hrs min. MIL-HDBK-217F (25°C)					
	DIMENSION	215*115*30mm (L*W*H)					
	PACKING	0.9Kg; 15pcs/14.5Kg/0.78CUFT					
NOTE	1. All parameters NOT specially mentioned are measured at 230VAC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance : includes set up tolerance, line regulation and load regulation. 4. Derating may be needed under low input voltages. Please check the derating curve for more details. 5. The power supply is considered a component which will be installed into a final equipment. The final equipment must be re-confirmed that it still meets EMC directives. For guidance on how to perform these EMC tests, please refer to "EMI testing of component power supplies." (as available on <a href="http://www.meanwell.com">http://www.meanwell.com</a> ) 6. For charging related applications, please consult Mean Well for details.						

**Mechanical Specification**

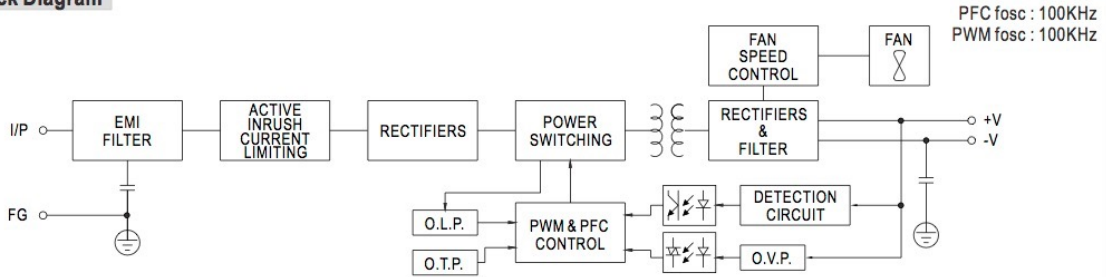
Case No.207A Unit:mm



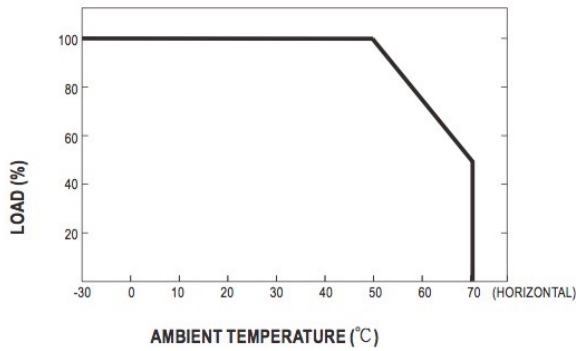
Terminal Pin No. Assignment :

Pin No.	Assignment	Pin No.	Assignment
1	AC/L	4-6	DC OUTPUT -V
2	AC/N	7-9	DC OUTPUT +V
3	FG $\oplus$		

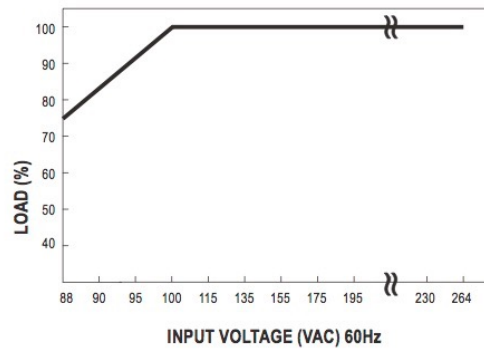
**Block Diagram**



**Derating Curve**



**Static Characteristics**



## 17. ANEXO 6: Hoja de datos del driver A4988



**A4988**

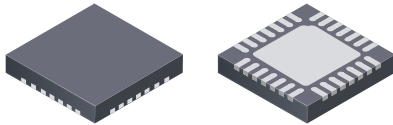
### *DMOS Microstepping Driver with Translator And Overcurrent Protection*

#### Features and Benefits

- Low  $R_{DS(ON)}$  outputs
- Automatic current decay mode detection/selection
- Mixed and Slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full,  $1/2$ ,  $1/4$ ,  $1/8$ , and  $1/16$

#### Package:

28-contact QFN  
with exposed thermal pad  
 $5\text{ mm} \times 5\text{ mm} \times 0.90\text{ mm}$   
(ET package)



Approximate size

#### Description

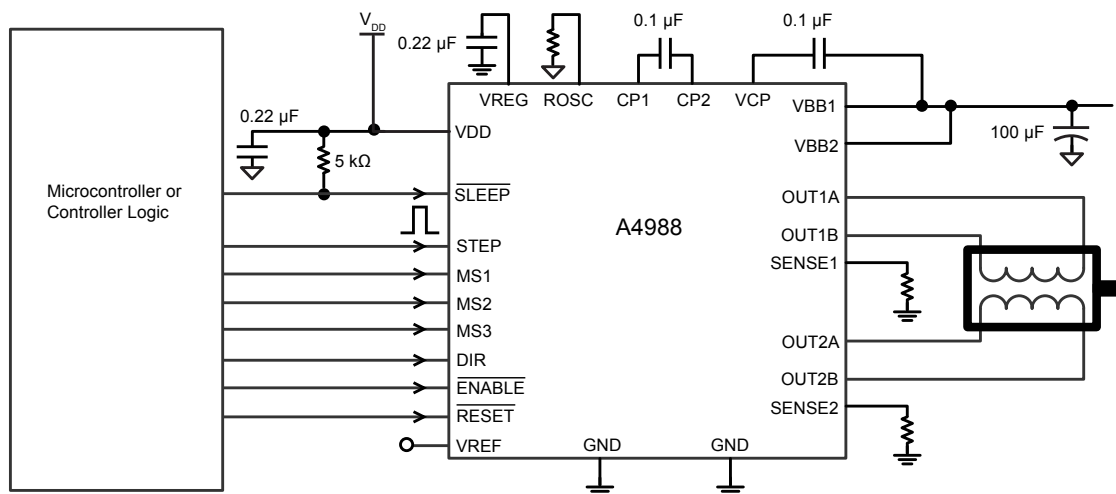
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and  $\pm 2$  A. The A4988 includes a fixed off-time current regulator which has the ability to operate in Slow or Mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

During stepping operation, the chopping control in the A4988 automatically selects the current decay mode, Slow or Mixed. In Mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

*Continued on the next page...*

#### Typical Application Diagram



## A4988

## *DMOS Microstepping Driver with Translator And Overcurrent Protection*

### Description (continued)

Internal synchronous rectification control circuitry is provided to improve power dissipation during PWM operation. Internal circuit protection includes: thermal shutdown with hysteresis, undervoltage lockout (UVLO), and crossover-current protection. Special power-on sequencing is not required.

The A4988 is supplied in a surface mount QFN package (ES), 5 mm × 5 mm, with a nominal overall package height of 0.90 mm and an exposed pad for enhanced thermal dissipation. It is lead (Pb) free (suffix -T), with 100% matte tin plated leadframes.

### Selection Guide

Part Number	Package	Packing
A4988SETTR-T	28-contact QFN with exposed thermal pad	1500 pieces per 7-in. reel

### Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Load Supply Voltage	$V_{BB}$		35	V
Output Current	$I_{OUT}$		±2	A
Logic Input Voltage	$V_{IN}$		-0.3 to 5.5	V
Logic Supply Voltage	$V_{DD}$		-0.3 to 5.5	V
Motor Outputs Voltage			-2.0 to 37	V
Sense Voltage	$V_{SENSE}$		-0.5 to 0.5	V
Reference Voltage	$V_{REF}$		5.5	V
Operating Ambient Temperature	$T_A$	Range S	-20 to 85	°C
Maximum Junction	$T_J(max)$		150	°C
Storage Temperature	$T_{stg}$		-55 to 150	°C

# A4988

## DMOS Microstepping Driver with Translator And Overcurrent Protection

### ELECTRICAL CHARACTERISTICS<sup>1</sup> at T<sub>A</sub> = 25°C, V<sub>BB</sub> = 35 V (unless otherwise noted)

Characteristics	Symbol	Test Conditions	Min.	Typ. <sup>2</sup>	Max.	Units
<b>Output Drivers</b>						
Load Supply Voltage Range	V <sub>BB</sub>	Operating	8	–	35	V
Logic Supply Voltage Range	V <sub>DD</sub>	Operating	3.0	–	5.5	V
Output On Resistance	R <sub>DSON</sub>	Source Driver, I <sub>OUT</sub> = –1.5 A	–	320	430	mΩ
		Sink Driver, I <sub>OUT</sub> = 1.5 A	–	320	430	mΩ
Body Diode Forward Voltage	V <sub>F</sub>	Source Diode, I <sub>F</sub> = –1.5 A	–	–	1.2	V
		Sink Diode, I <sub>F</sub> = 1.5 A	–	–	1.2	V
Motor Supply Current	I <sub>BB</sub>	f <sub>PWM</sub> < 50 kHz	–	–	4	mA
		Operating, outputs disabled	–	–	2	mA
Logic Supply Current	I <sub>DD</sub>	f <sub>PWM</sub> < 50 kHz	–	–	8	mA
		Outputs off	–	–	5	mA
<b>Control Logic</b>						
Logic Input Voltage	V <sub>IN(1)</sub>		V <sub>DD</sub> ×0.7	–	–	V
	V <sub>IN(0)</sub>		–	–	V <sub>DD</sub> ×0.3	V
Logic Input Current	I <sub>IN(1)</sub>	V <sub>IN</sub> = V <sub>DD</sub> ×0.7	–20	<1.0	20	μA
	I <sub>IN(0)</sub>	V <sub>IN</sub> = V <sub>DD</sub> ×0.3	–20	<1.0	20	μA
Microstep Select	R <sub>MS1</sub>	MS1 pin	–	100	–	kΩ
	R <sub>MS2</sub>	MS2 pin	–	50	–	kΩ
	R <sub>MS3</sub>	MS3 pin	–	100	–	kΩ
Logic Input Hysteresis	V <sub>HYS(IN)</sub>	As a % of V <sub>DD</sub>	5	11	19	%
Blank Time	t <sub>BLANK</sub>		0.7	1	1.3	μs
Fixed Off-Time	t <sub>OFF</sub>	OSC = V <sub>DD</sub> or GND	20	30	40	μs
		R <sub>OSC</sub> = 25 kΩ	23	30	37	μs
Reference Input Voltage Range	V <sub>REF</sub>		0	–	4	V
Reference Input Current	I <sub>REF</sub>		–3	0	3	μA
Current Trip-Level Error <sup>3</sup>	err <sub>I</sub>	V <sub>REF</sub> = 2 V, %I <sub>TRIPMAX</sub> = 38.27%	–	–	±15	%
		V <sub>REF</sub> = 2 V, %I <sub>TRIPMAX</sub> = 70.71%	–	–	±5	%
		V <sub>REF</sub> = 2 V, %I <sub>TRIPMAX</sub> = 100.00%	–	–	±5	%
Crossover Dead Time	t <sub>DT</sub>		100	475	800	ns
<b>Protection</b>						
Overcurrent Protection Threshold <sup>4</sup>	I <sub>OCST</sub>		2.1	–	–	A
Thermal Shutdown Temperature	T <sub>TSD</sub>		–	165	–	°C
Thermal Shutdown Hysteresis	T <sub>TSDHYS</sub>		–	15	–	°C
VDD Undervoltage Lockout	V <sub>DDUVLO</sub>	V <sub>DD</sub> rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	V <sub>DDUVLOHYS</sub>		–	90	–	mV

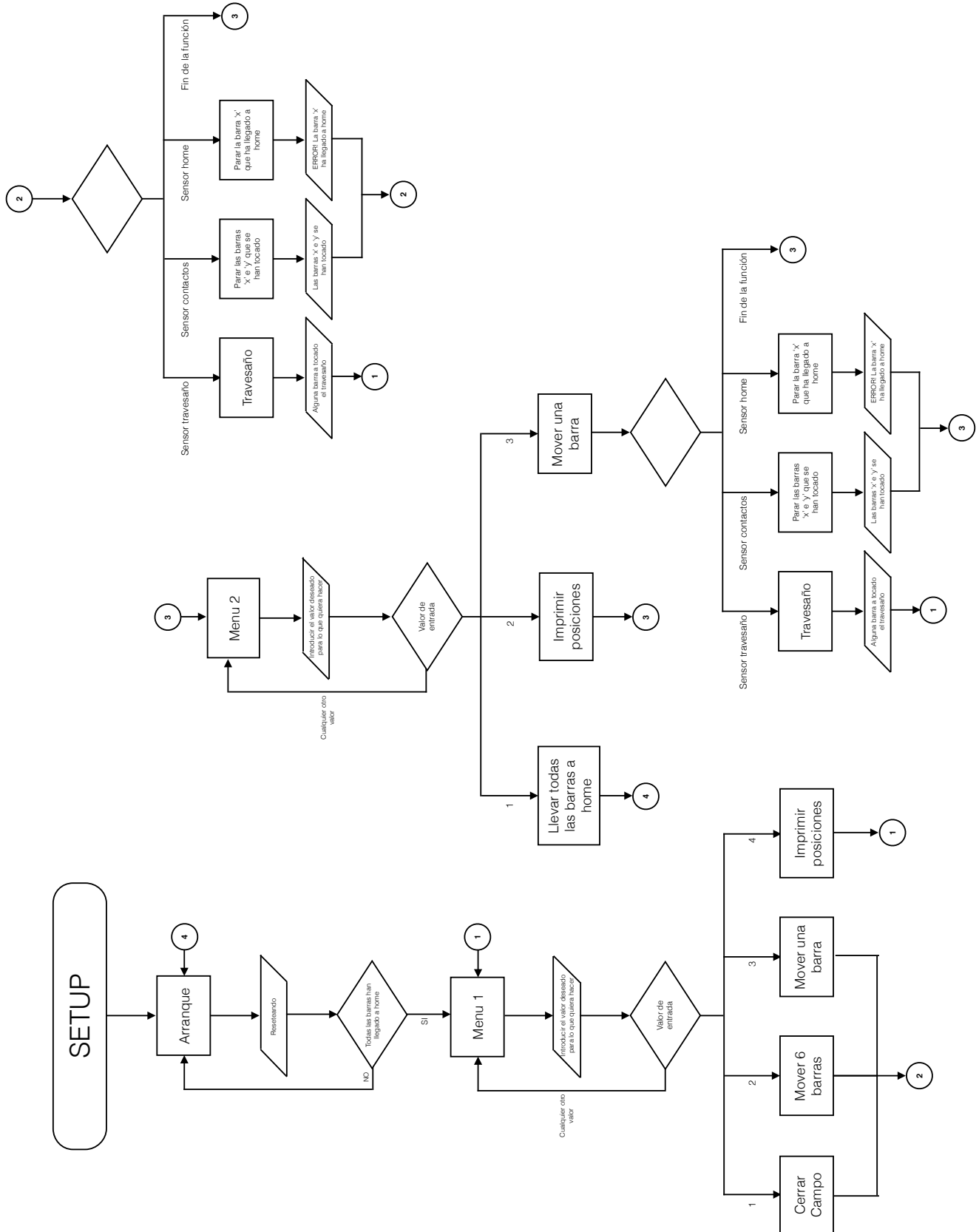
<sup>1</sup>For input and output current specifications, negative current is defined as coming out of (sourcing) the specified device pin.

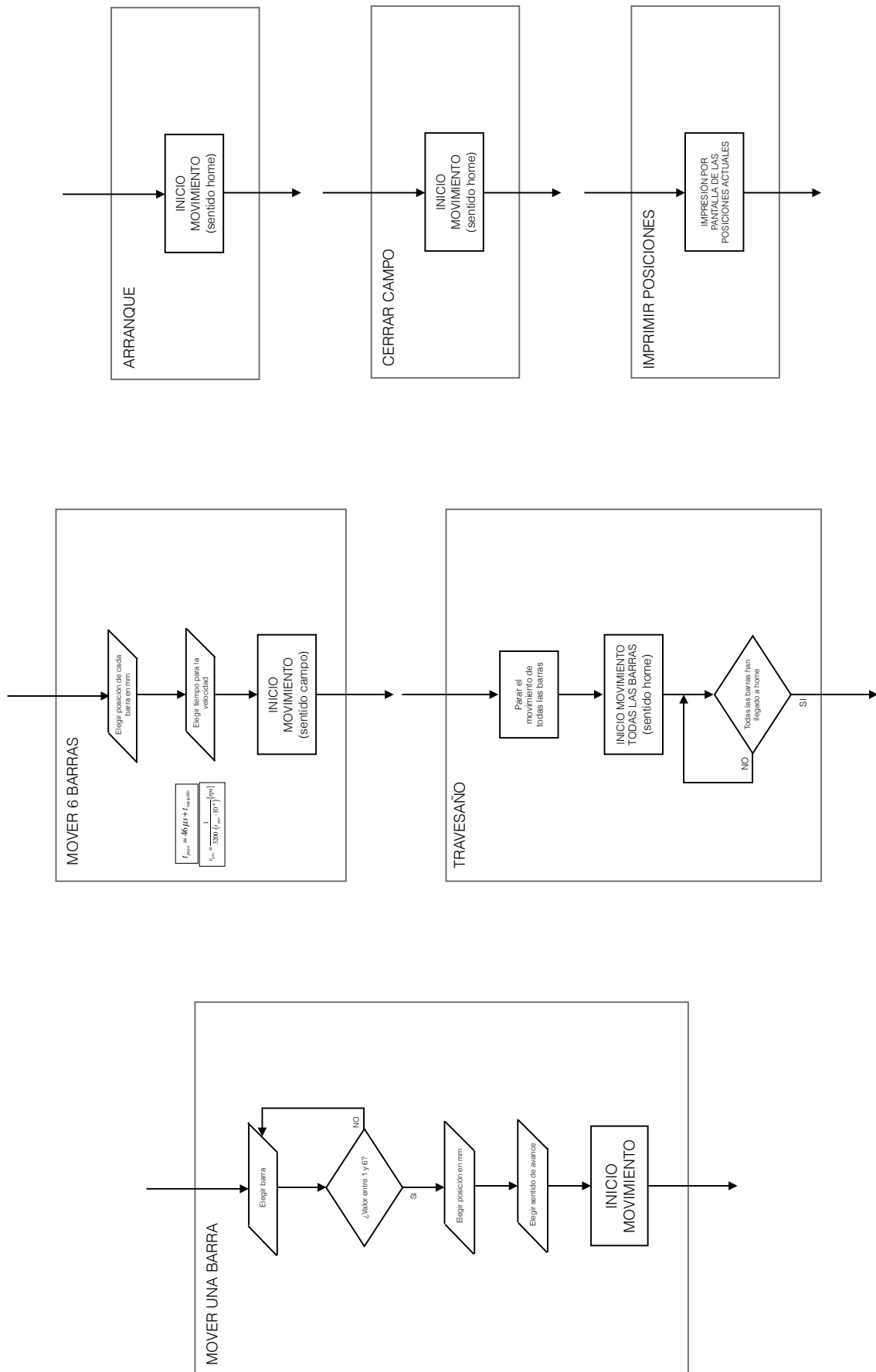
<sup>2</sup>Typical data are for initial design estimations only, and assume optimum manufacturing and application conditions. Performance may vary for individual units, within the specified maximum and minimum limits.

<sup>3</sup>V<sub>ERR</sub> = [(V<sub>REF</sub>/8) – V<sub>SENSE</sub>] / (V<sub>REF</sub>/8).

<sup>4</sup>Overcurrent protection (OCP) is tested at T<sub>A</sub> = 25°C in a restricted range and guaranteed by characterization.

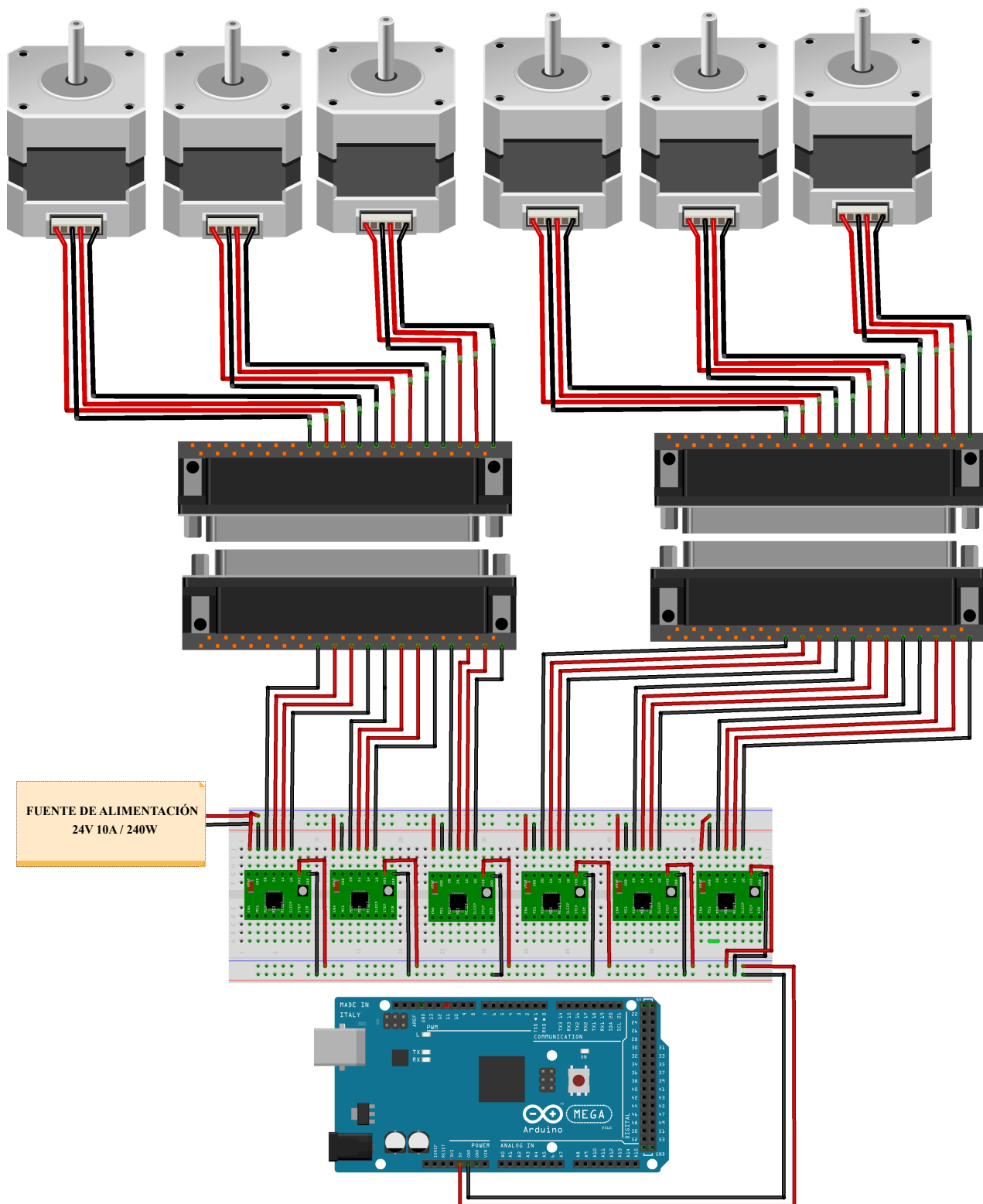
## 18. ANEXO 7: Diagrama de flujo del programa del sistema de control





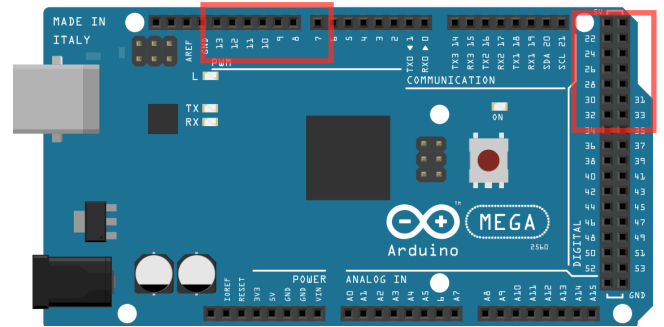


## 19. ANEXO 8: Conexiones eléctricas



## 20. ANEXO 9: Relación de pines Arduino-Divers y Arduino-Sensores

Pines Arduino-Divers(A4988)			
	ENABLE	DIR	STEP
Motor 1	22	23	12
Motor 2	28	29	9
Motor 3	24	24	11
Motor 4	30	31	8
Motor 5	26	27	10
Motor 6	32	33	7



Pines Arduino-Sensores			
	HOME	TRAVESAÑO	CONTACTO
Motor 1	35	41	36
Motor 2	43		
Motor 3	37		38
Motor 4	39		
Motor 5	42		40
Motor 6	44		

