**Autor:** Alberto García (Tach80)

**Contacto:** tach980@gmail.com

Publicado bajo licencia GPLv3

**RAMPS 1.6**

**Referencia**

Todas las indicaciones se dan para la RAMPS con los conectores hacia arriba y las entradas de corriente a la izquierda, de manera que la fila superior permita conectar dos controladoras y la parte inferior tres controladoras, como se muestra en la imagen inferior:

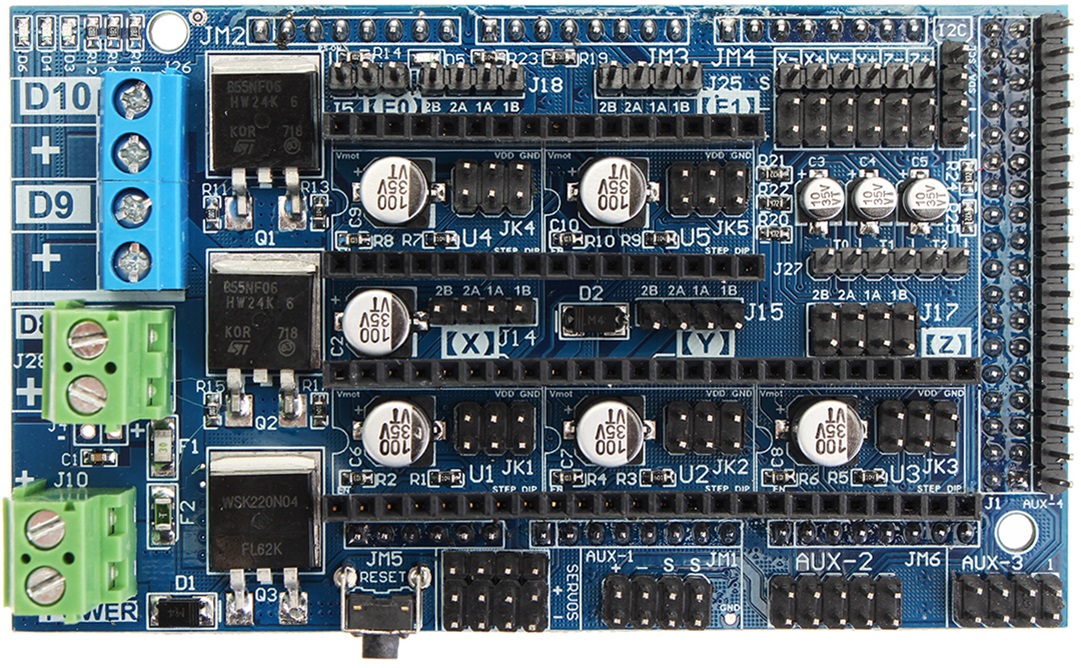


Imagen 1: muestra de la RAMPS 1.6

La placa es una shield Arduino, que requiere un chip MEGA 2560. Para la Anet A8, se requieren cuatro controladoras adicionales (una por eje más extrusor). Se recomienda la DRV8825 (foto con leyenda de pines a continuación):

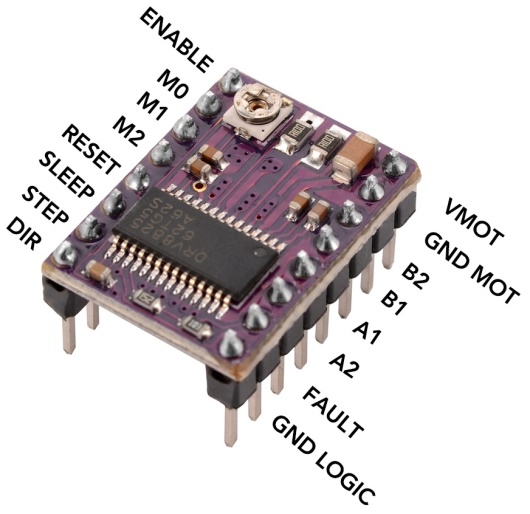


Imagen 2: imagen extraída de https://www.ebay.com/itm/5PCS-DRV8825-stepper-motor-driver-Module-3D-printer-RAMPS1-4-RepRap-StepStickvTE-/143197789220?oid=142298147577

Para la migración serán necesarios conectores Dupont, herramientas de des-crimpado de conector poste (púa fina, alicate especializado o bisturí plano), alicates y/o pelacables. El montaje de la RAMPS es el siguiente:

1. **Conectar Arduino**

La Arduino tiene dos entradas de corriente: USB y 2.1 mm jack de centro positivo, con selección automática de entrada. Sólo una entrada es necesaria. La RAMPS coge la corriente de sus propias entradas.

La orientación correcta es la que hace coincidir todas las entradas de corriente de la Arduino y la RAMPS en el mismo lateral. Con la orientación de referencia, la entrada USB y de corriente de la Arduino debería quedar a la izquierda.

La inserción debe hacerse en vertical y tratando de que todos los pines encajen a la vez: dado el tamaño de los conectores, un giro u orientación inadecuados dañará los pines.

1. **Colocar jumpers.**

**Microsteping:** el envío de un pulso que haga que el motor se mueva a una fracción de un paso completo. Es una característica de la controladora, no del stepper, y un ratio mayor implica un torque de retención menor.

Los jumpers se encuentran en la parte central, entre los conectores de las controladoras y son cinco grupos (uno por cada eje más dos para los extrusores) de 2x3 pines.

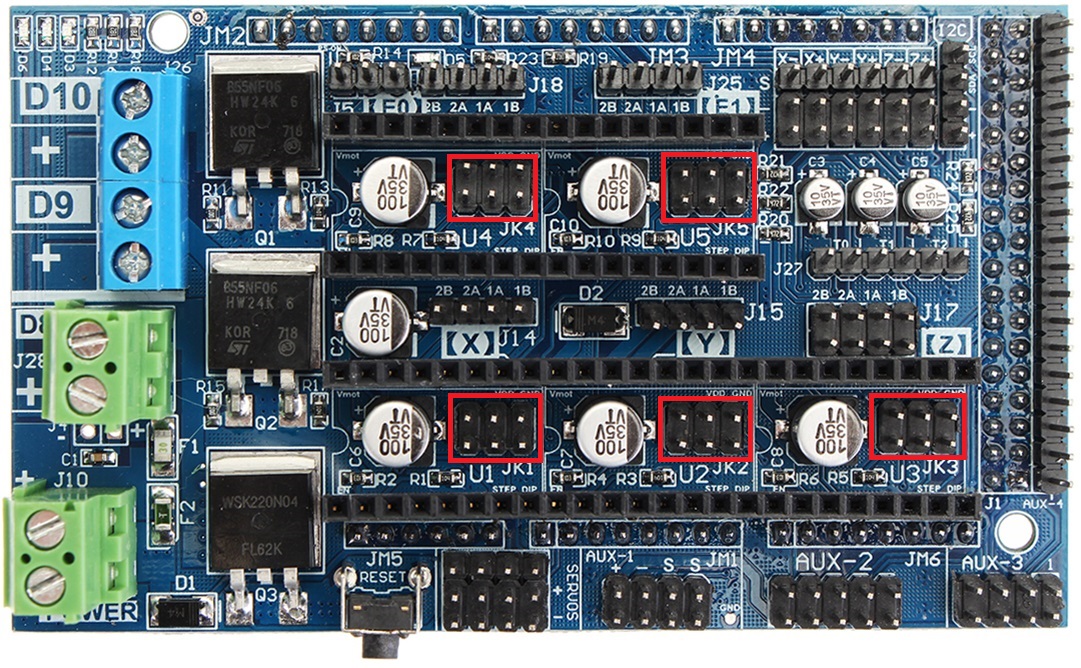


Imagen 3: ubicaciones de los jumpers. De arriba a abajo y de izquierda a derecha, extrusor 0 y extrusor 1 (fila superior) y eje e, y, z (fila inferior). Los jumpers se numeran de izquierda a derecha.

Hay que colocar verticalmente el jumper 3 para un ratio 1/16, aunque pueden conectarse para un ratio mayor sin que tenga efecto (el máximo soportado por el DRV8825 es 1/32, y subir el ratio obligará a modificar el firmware más adelante).

**Advertencia:** conectar los jumpers en horizontal puede cortocircuitar la placa.

|  |  |  |  |
| --- | --- | --- | --- |
| Jumper 1 | Jumper 2 | Jumper 3 | Tamaño del paso |
| 0 | 0 | 0 | Full step |
| 1 | 0 | 0 | Half step |
| 0 | 1 | 0 | 1/4 step |
| 1 | 1 | 0 | 1/8 step |
| 0 | 0 | 1 | 1/16 step |
| 1 | 0 | 1 | 1/32 step |
| 0 | 1 | 1 | 1/64 step |
| 1 | 1 | 1 | 1/128 step |

Tabla extraída de <https://reprap.org/wiki/RAMPS_1.4>. En verde la configuración por defecto de la Anet A8 y del extrusor Titan.

1. **Insertar controladoras**

**Peligro:** las controladoras están polarizadas. Invertir la polaridad las quema en el acto.

La RAMPS admite hasta cinco controladoras: tres de posición más dos extrusores.

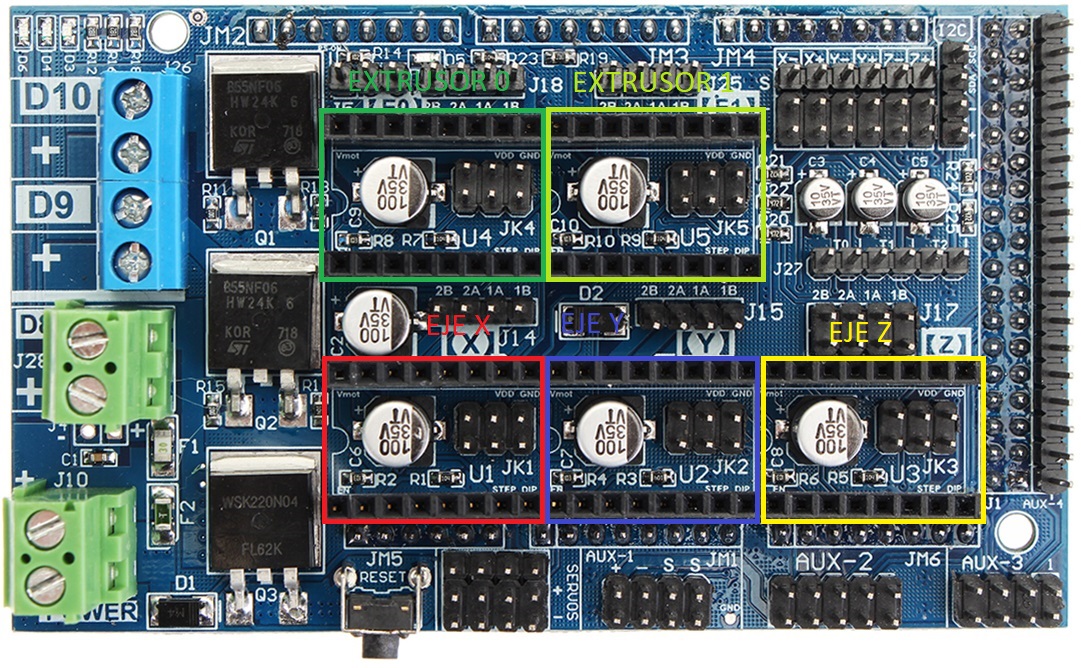


Imagen 4: pines correspondientes a cada una de las controladoras, indicando función. Nótese que los jumpers quedan debajo de las controladoras, y que el eje Z permite conectar dos motores en vez de uno.

La orientación correcta es la que hace que los pines de FAULT y GND LOGIC de la controladora coincidan con los mismos que la placa: con los datos de las imágenes, la controladora correctamente conectada encajará de manera que el tornillo regulador de corriente quede a la izquierda y el transistor quede a la derecha.

Una vez orientadas las controladoras, la inserción es directa, con una púa para cada conector.

**Cuidado:** si las obleas de las controladoras no están bien cortadas es posible que no encajen del todo. Es posible que tengas que probar diferentes combinaciones hasta dar con el juego adecuado.

1. **Conectar motores**

Los conectores habituales de la Anet son conectores de tipo poste, pero la RAMPS sólo deja espacio para conectores Dupont. Hay que crimpar nuevamente todos los cables.

Los motores de la Anet A8 motores de 6 pines, emparejados, cuatro vivos y cuatro neutros. El cuadro de polaridades es el siguiente (revisar polaridades y colores):

|  |  |  |
| --- | --- | --- |
| Pin | Polaridad | Pareja |
| 1 | + | 4 |
| 2 | Neutro |  |
| 3 | + | 6 |
| 4 | - | 1 |
| 5 | Neutro |  |
| 6 | - | 3 |

La inversión de polaridad supone inversión del giro. Puede modificarse por firmware. Los cables se ordenan de izquierda a derecha:

|  |  |  |
| --- | --- | --- |
| Eje | Métrica | Modelo |
| X | NEMA 17 | SL42STH40-1684A |
| Y | NEMA 17 | SL42STH40-1684A |
| Z (2x) | NEMA 17 | SL42STH40-1684A |
| Extrusor | NEMA 17 | 42SHDC3025-24B |

Tabla de especificaciones de los motores:

|  |  |
| --- | --- |
| Modelo | SL42STH40-1684A |
| Ángulo de paso | 1.8º ± 5% full step, sin carga |
| Voltaje | 2.8 V |
| Corriente por fase | 1.68 A |
| Resistencia por fase | 1.65 ± 10% |
| Inductancia por fase | 3.2 mH ± 20% |
| Torque estático | 3.6 Kg·cm (0.4 N·m) |
| Momento de inercia | 54 g/cm^2 |
| Peso | 280 g |
| Torque orientación | 150g/cm |
| Tamaño | 40 mm |
| Ascenso de temperatura | 80 ºC (dos fases) |
| Temperatura ambiente | -20°C ~ +50°C |
| Resistencia aislante | 100MΩ min, 500VDC |
| Juego radial del eje | 0.02 mm max (450 g de carga) |
| Juego axial del eje | 0.08 mm max (450 g de carga) |
| Máxima fuerza radial | 28 N (20 mm del borde) |
| Máxima fuerza axial | 10 N |
| Fuente | [Fuente](https://3dprint.wiki/reprap/anet/a8/steppermotor) |

**Consejo:** crimpar los cables en conectores Dupont de dos pines, poniendo en cada conector los dos pines que forman la pareja y anotar cuál es cada polo.

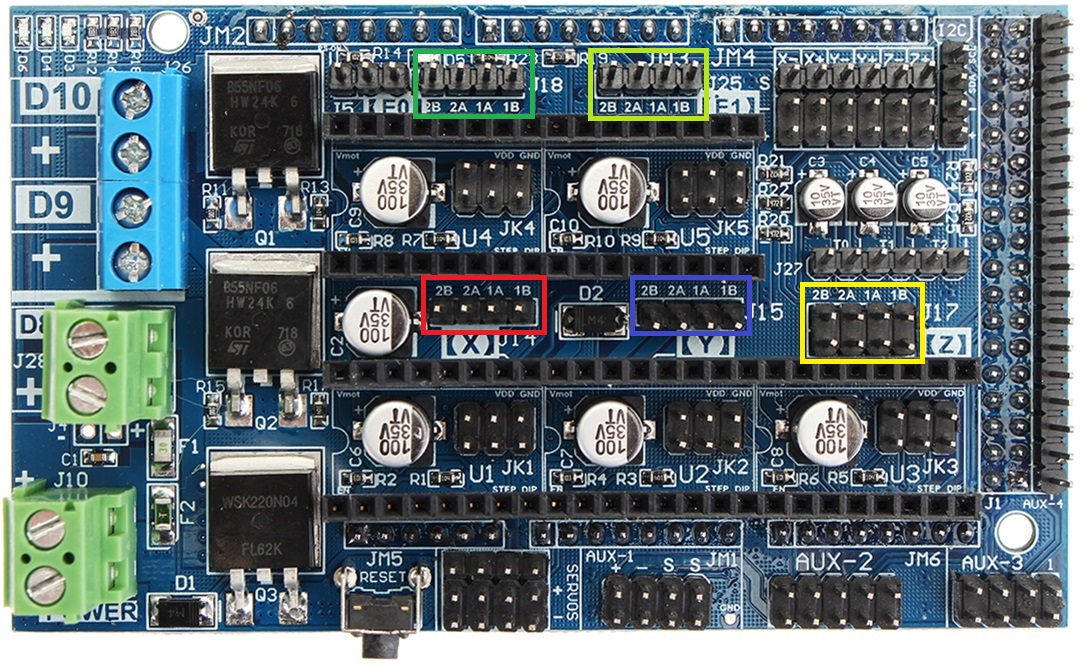


Imagen 5: pines de alimentación correspondientes a los motores de dichos ejes.

**Nota:** en el caso del eje Z, hay dos juegos de pines, siendo cada fila un juego diferente. Cada motor ha de conectar todos sus pines a la fila que corresponda.

Como referencia, conectar todos los polos positivos a los conectores XA y el otro pin a los conectores XB.

**Nota:** aunque no hay peligro por invertir la polaridad de un motor, complica la definición de los mismos en el firmware. Usando un único criterio se tarda menos en configurar y probar después.

1. **Conectar sensores de posición**

Como con los motores, los conectores son de tipo poste, así que hay que volver a crimpar a conectores Dupont de dos o tres pines.

Los sensores tienen tres patillas, pero solamente son necesarias dos: una de corriente y una de tierra. Como se trata de un interruptor sencillo no tiene polaridad, de modo que pueden crimparse indistintamente (aunque se recomienda hacerlos todos con el mismo criterio).

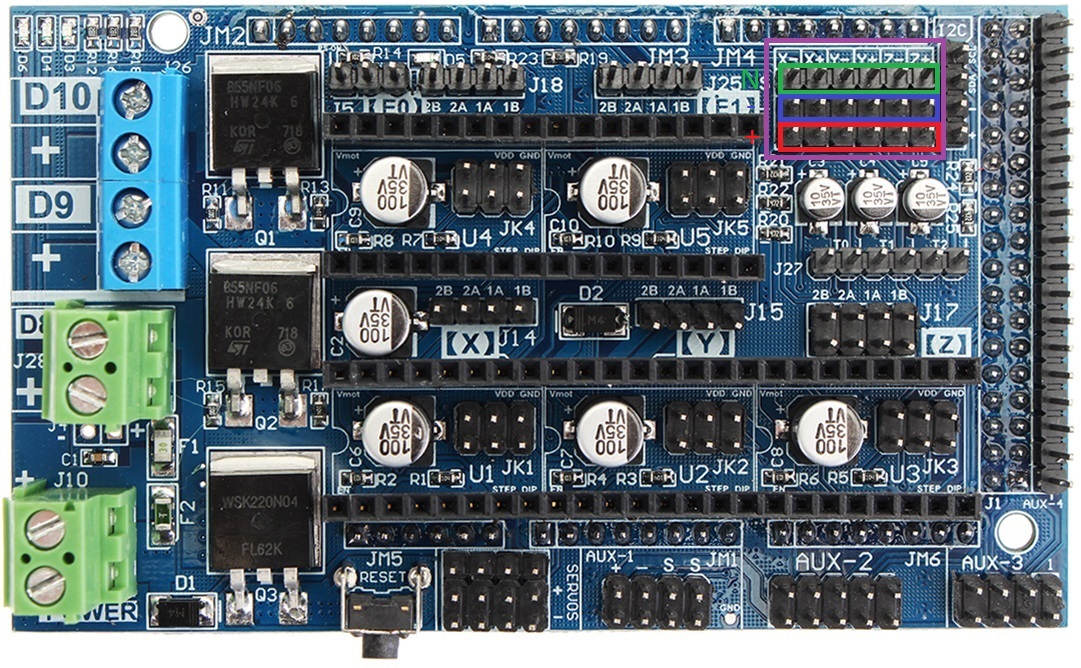


Imagen 6: conectores para los topes de eje.

Las conexiones deben realizarse al menos desde una línea positiva o negativa a la toma neutra (que recibe la señal del interruptor). Dado que los topes de la Anet A8 son NO (*Normally Open*), no es necesario conectar los tres pines.

Aunque se puede modificar por firmware (como el sentido de giro de los motores), se aconseja conectar todos los topes al mismo polo: todos en A+ a neutro o en A- a neutro.

**Nota:** por simplicidad, se recomienda conectar todos los topes de la misma manera, sean todos positivos o negativos, a neutro.

**Advertencia:** en el caso de tener conectores de dos pines, aunque no tiene por qué pasar nada, se aconseja no conectar los dos pines a corriente por dos motivos: primero, que al no conectarse a neutro la placa no recibe la señal de interrupción; segundo, que puedes dañar el interruptor.

1. **Conectar controladora de pantalla**

Hay que pinchar el conector específico de la pantalla a la RAMPS. Los pines se encuentran en la fila vertical de la parte superior derecha, más a la derecha de los pines usados para los endstops.

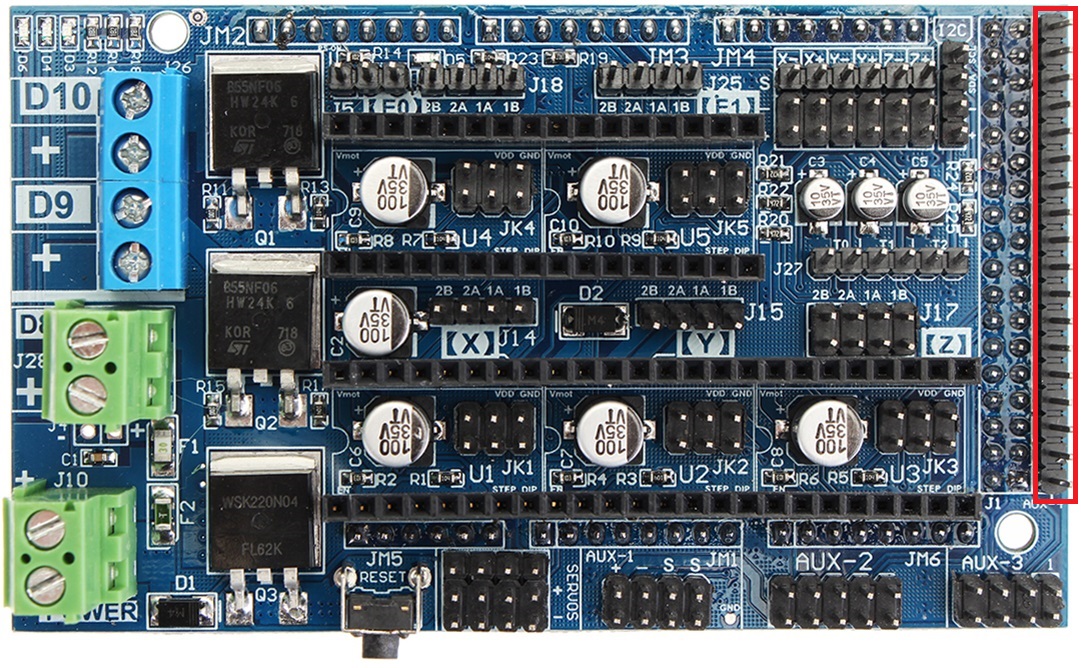


Imagen 7: pines para la conexión de la interfaz de pantalla.

Una vez conectado, hay que enchufar las dos cintas al conector, y éstas a la pantalla., respetando el número indicado en la pantalla y en el conector.

1. **Conectar calentadores (ver sondas más adelante)**

**Nota:** aunque son resistencias no polarizadas, se aconseja conectarlas como si lo fueran.

* + **Cama caliente:** las tomas de corriente se conectan a D8.
  + **Resistencia del extrusor:** se conecta a D10.

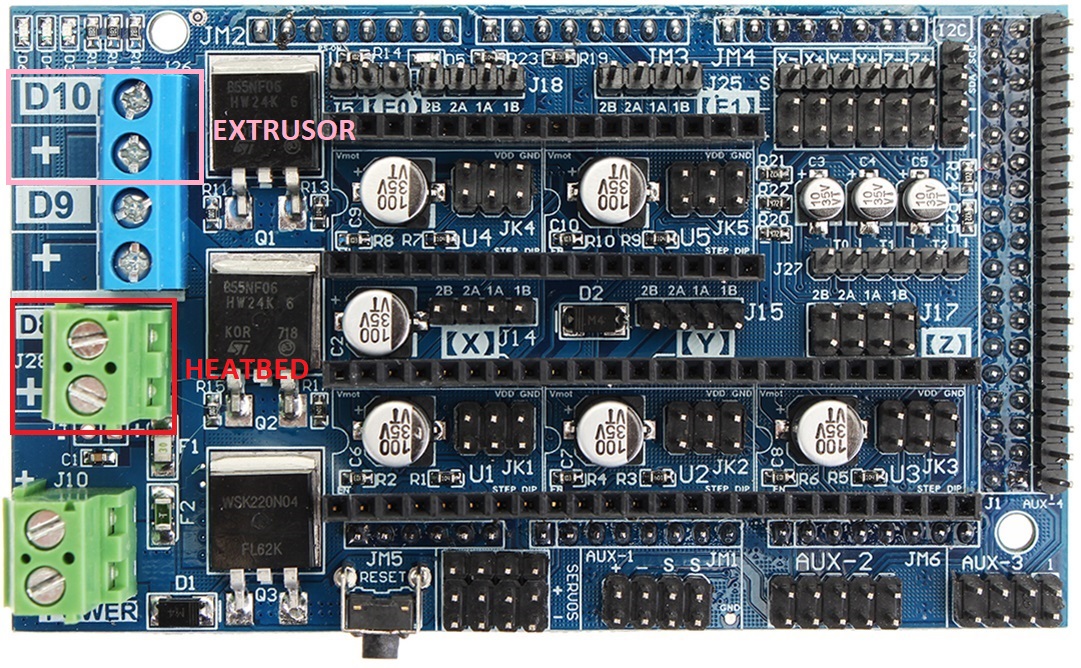


Imagen 8: conectores de extrusor y cama caliente.

1. **Conectar los ventiladores**

**Notas importantes**

* Todos los ventiladores son polarizados a 12 V. Prueba con el tester primero (o prueba con una pila de 9 V: arrancará el ventilador a baja velocidad sin quemarlo inmediatamente en caso de error).
* NO HAY salida de 12 V aparte de las D8-D10.
* **Peligro:** conectar mal a 12 V un ventilador lo quema inmediatamente.
  + **Ventilador-disipador:** se conecta directamente a la fuente de alimentación.
  + **Soplador de aire:** se conecta al D9.

1. **Conectar sensores de temperatura**

Las conexiones se hacen a los pines indicados. Ya que los termistores son resistencias variables, no están polarizados y se pueden conectar en cualquier orden (siempre que sea en la pareja correcta de pines).

**Nota:** aunque un error no causa problemas inmediatamente, conectar las sondas en los pines equivocados puede sobrecalentar la cama y dañarla.

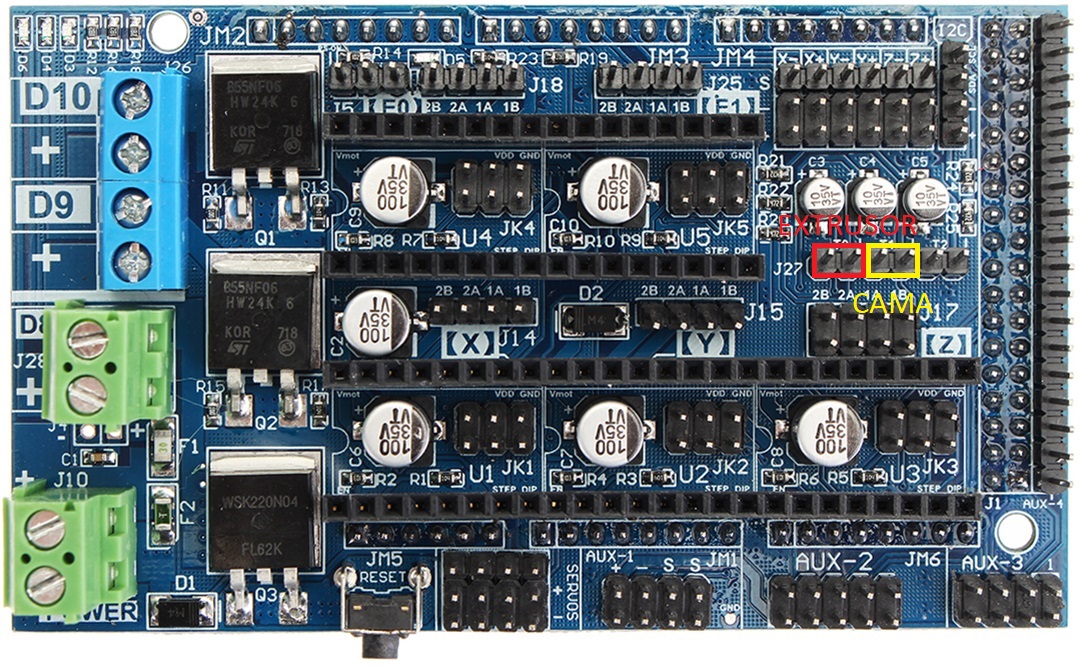


Imagen 9: conectores no polarizados para los termistores.

1. **Conexiones eléctricas**

Hay que poner una línea polarizada desde la fuente de alimentación a POWER.

**Peligro:** una inversión en la polaridad quemará la placa inmediatamente.

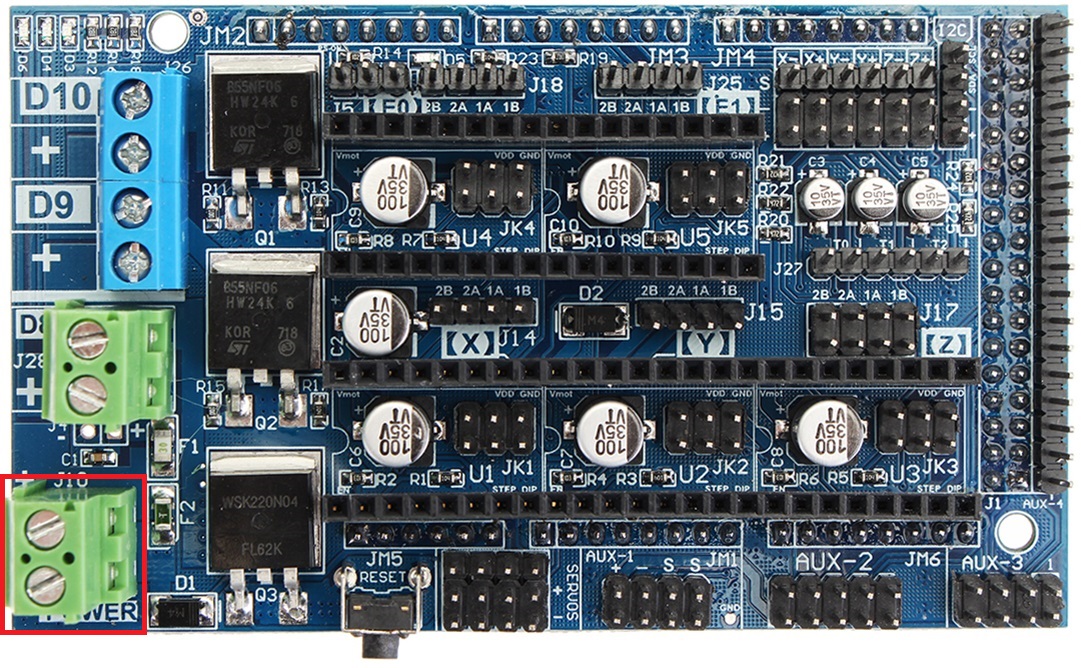


Imagen 10: conector de entrada de corriente para la RAMPS.

**Configuración del firmware**

La RAMPS 1.6 es idéntica (con alguna modificación menor) de la RAMPS 1.4: las configuraciones de la 1.4 deberían valer para la 1.6.

<https://www.my-home-fab.de/en/documentations/reprap-firmware/installing-marlin-firmware-on-ramps-1.4>

1. Preparar y configurar el entorno de Arduino.
   * Baja e instala el IDE de Arduino.

<https://www.arduino.cc/en/Main/Software>

* + Baja y descomprime el proyecto de Marlin.

<https://github.com/MarlinFirmware/Marlin>

* + (Opcional) Instala la librería u8glib (obsoleta, usar u8g2) para soporte de pantalla LCD, directamente en el IDE de Arduino.

En *Sketch -> Librerías -> Gestionar librerías* busca “u8g2” e instala la última versión. La página del proyecto es la siguiente:

<https://github.com/olikraus/u8g2/wiki>

* + Conectar la placa.

En primer lugar, conecta la placa Arduino y espera que el ordenador la reconozca. Es posible que tengas que instalar drivers.

Una vez el equipo la haya reconocido hay que seleccionarla en el IDE de Arduino. Para ello, ve a *Herramientas -> Placa* y selecciona *Arduino Mega 2560*. Después, hay que seleccionar el puerto con el que el equipo se comunicará con la placa. Para saberlo, hay dos posibilidades:

* A través del propio IDE de Arduino.

En *Herramientas -> Puerto* aparecerá un listado de los puertos que el IDE detecta como ocupados. Selecciona el que contenga la placa que has conectado.

* A través del Administrador de dispositivos de Windows.

Yendo al *Administrador de dispositivos* de Windows, encontrarás en la parte inferior de la lista los dispositivos conectados a puertos COM. Busca el que corresponda con la placa que has conectado, donde se indicará entre paréntesis el número del puerto que le corresponde. Después, selecciona ese puerto en el IDE.

* ¿Hay alguna posibilidad de hacerlo en Linux? A través del propio IDE.

1. Importa, modifica y vuelca el firmware.
   * Importar el proyecto.

Una vez instalada la placa, importa el proyecto, haciendo clic en *Archivo -> Abrir* y busca la carpeta del proyecto, donde la hayas descomprimido. Busca y abre el archivo *marlin.ino*, para que el IDE abra todos los archivos relacionados con el proyecto.

* + Modificaciones al proyecto.

**Importante:** es posible que tengas que hacer más modificaciones que las que aparecen aquí. Una vez acabes y estés satisfecho con los resultados, te recomiendo mantener una copia del código en lugar seguro, para no tener que repetir el proceso.

**Nota:** si las modificaciones que describo aquí no funcionan o no te convencen, no te empeñes en ellas. Parte del encanto de este tipo de proyectos está en encontrar la solución que mejor te funcione.

**Configuración de velocidad de interfaz serie**

Abre la pestaña que contiene el archivo *Configuration.h*. Verás un sinfín de líneas. La primera modificación que vamos a hacer es modificar la velocidad de transferencia sobre la interfaz USB. Esto es importante para conseguir estabilidad en la conexión y que la interfaz no se caiga en medio de la impresión por una sobrecarga de datos.

Por defecto, este parámetro tiene un valor de 250000, que modificaremos por uno un poco más conservador. La línea debería quedar de la siguiente manera:

// This determines the communication speed of the printer

#define BAUDRATE 115200

**Definición, BAUDRATE:** velocidad de transferencia de símbolos sobre una interfaz serie. En este caso, se monta una interfaz serie sobre un puerto USB (Aviso: no es lo mismo “interfaz serie” que “puerto serie”). Es un “ancho de banda”, la velocidad de transmisión estándar en puertos serie.

**Configuración de salidas de corriente (extrusor, cama, soplador)**

La siguiente modificación es la que indica la configuración de los puertos de alimentación para cama caliente, extrusor y ventilador. Tal y como lo hemos montado, la configuración es Extrusor (D10), Ventilador (D9) y Cama caliente (D8).

Podemos ver diferentes configuraciones en el archivo *boards.h*, donde se definen diferentes códigos para diferentes configuraciones. En nuestro caso, hemos de buscar los códigos correspondientes a la RAMPS 1.4. En el commit de 09/06/18 las definiciones son las siguientes:

#define BOARD\_RAMPS\_14\_EFB 43 // RAMPS 1.4 (Power outputs: Hotend, Fan, Bed)

#define BOARD\_RAMPS\_14\_EEB 44 // RAMPS 1.4 (Power outputs: Hotend0, Hotend1, Bed)

#define BOARD\_RAMPS\_14\_EFF 45 // RAMPS 1.4 (Power outputs: Hotend, Fan0, Fan1)

#define BOARD\_RAMPS\_14\_EEF 46 // RAMPS 1.4 (Power outputs: Hotend0, Hotend1, Fan)

#define BOARD\_RAMPS\_14\_SF 48 // RAMPS 1.4 (Power outputs: Spindle, Controller Fan)

De nuevo en *Configuration.h*, hemos de modificar las siguientes líneas por las que corresponda a nuestra configuración. El aspecto final es el siguiente:

#ifndef MOTHERBOARD

#define BOARD\_RAMPS\_14\_EFB

#endif

**Configuración de pantalla LCD**

El siguiente paso si tienes pantalla LCD es configurarla. En el commit de 18/09/18 hay una sección sobre pantallas LCD en torno a la línea 1520. Nuestra pantalla es una RepRap Discount, cuya definición se encuentra en torno a la línea 1530. Hay que quitar las dos barras (“//”) al inicio de la línea (descomentarla):

// RepRapDiscount Smart Controller.

// http://reprap.org/wiki/RepRapDiscount\_Smart\_Controller

//

// Note: Usually sold with a white PCB.

//

#define REPRAP\_DISCOUNT\_FULL\_GRAPHIC\_SMART\_CONTROLLER

**Configuración de lector de tarjeta SD**

uncomment #define SDSUPPORT for SD CARD support

uncomment #define SD\_CHECK\_AND\_RETRY

set #define ENCODER\_PULSES\_PER\_STEP 4

set #define ENCODER\_STEPS\_PER\_MENU\_ITEM 2

choose if you need #define SPEAKER

http://www.rodic.si/wordpress/reprapdiscount-full-graphic-smart-controller-lcd-sd-card-for-ramps-1-4-to-any-3d-printer/

Con estas modificaciones ya hemos terminado la etapa “base” de la impresora. Si quemases este firmware a la placa la impresora ya daría las primeras señales de vida (aunque no estaría operativa aún).

**Configuración de extrusores**

Puede que una de las características más definitorias de la impresora sea el número de extrusores que tiene. En el commit de 18/09/18 se define por defecto un extrusor en la línea 149. No es necesario hacer modificaciones, aunque lo dejo anotado por si quieres añadir un segundo extrusor más adelante:

// This defines the number of extruders

#define EXTRUDERS 1

**Configuración de temperaturas**

Una vez definido el número de extrusores, el siguiente paso es indicar las temperaturas de trabajo de la cama caliente y del extrusor. En el commit de 18/09/18 se definen las sondas a partir de la línea 260 en una sección específica. Te recomiendo que leas con detenimiento los comentarios de los desarrolladores para ver los detalles sobre los diferentes sensores que admite Marlin, y elegir el que corresponda a tu máquina.

**Nota:** en el siguiente apartado aparecen múltiples definiciones que sólo se diferencian en el índice final. Es para indicar el número de extrusor al que se refieren. Como la siguiente configuración es para un único extrusor (con índice 0), el resto de definiciones han de dejarse con los valores por defecto. Lógicamente, si montas más de un extrusor, habrás de rellenar más campos, repitiendo las instrucciones para cada uno de los extrusores.

* **Hotend**

Tras revisar la documentación y los archivos de ejemplo, veo que el resistor es un *ATC Semitec 104GT-2/104NT-4-R025H42G* para la Anet A8. En mi caso, el hotend es el E3D V6, con el mismo modelo de resistor, como aparece indicado en el siguiente enlace:

<https://e3d-online.dozuki.com/Guide/V6+Marlin+Configuration/5?lang=en>

Para la configuración de un solo extrusor, hemos de modificar el siguiente parámetro:

#define TEMP\_SENSOR\_0 5

Faltan por definir las temperaturas máximas y mínimas que puede/debe alcanzar el resistor. La máxima temperatura que el resistor del V6 puede soportar es de 285ºC, así que debemos introducirlo en el campo apropiado, que quedará de la siguiente manera:

#define HEATER\_0\_MAXTEMP 285

Para la temperatura mínima, requiere una breve explicación: no se trata de una temperatura mínima, sino un valor que el firmware utilizará como referencia para detectar un mal funcionamiento del sensor. En principio, dado que la impresora se encuentra en el interior de una vivienda, la temperatura no debería caer demasiado, así que se puede emplear el valor por defecto (5ºC).

#define HEATER\_0\_MINTEMP 5

* **Cama caliente**

Queda definir los valores para la cama caliente. Tomando como referencia los valores del archivo de configuración de ejemplo para la Anet A8 (de la que usaremos la cama), el resistor empleado vuelve a ser el *ATC Semitec 104GT-2/104NT-4-R025H42G*. Por tanto, el parámetro quedará de la siguiente manera:

#define TEMP\_SENSOR\_BED 5

De nuevo, hay que definir las temperaturas máxima y mínima que la cama puede/debe alcanzar. En este caso, la cama caliente de la Anet A8 por defecto alcanza los 60ºC de manera habitual, pudiendo llegar hasta los 80ºC, aunque no se recomienda. Por ello, bajaremos la temperatura máxima hasta los 85ºC:

#define BED\_MAXTEMP 85

Empleando el mismo razonamiento que para el extrusor, dejaremos la temperatura mínima en 5ºC:

#define BED\_MINTEMP 5

Con estos datos damos por finalizada esta sección.

**Configuración de los motores stepper**

**Peligro:** no utilices la opción “home” o “homing” de tu impresora hasta terminar complemente esta sección. Usa los controles del display o del servidor de impresión que utilices para las pruebas.

**Advertencia:** un fallo en la configuración de esta sección puede dañar los componentes mecánicos de la impresora, ya que los stepper podrían no detectar el fin de carrera y forzar alguno de sus elementos. Estate preparado para desconectar la impresora en cualquier momento cuando hagas las pruebas.

**Sentido de giro de los motores stepper**

En esta sección veremos la configuración del sentido de giro de los motores stepper. Como ya hemos comentado, es posible invertir su sentido de giro por software empleando esta sección.

Encontrarás la configuración del sentido de giro de los stepper hacia la línea 850, donde se puede invertir su sentido de giro cambiando el booleano:

// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.

#define INVERT\_X\_DIR false

#define INVERT\_Y\_DIR true

#define INVERT\_Z\_DIR false

Un poco más abajo encontrarás las mismas opciones para el stepper del extrusor. Como ya hemos comentado antes, modifica el valor “0” para el primer extrusor.

// For direct drive extruder v9 set to true, for geared extruder set to false.

#define INVERT\_E0\_DIR false

#define INVERT\_E1\_DIR false

#define INVERT\_E2\_DIR false

#define INVERT\_E3\_DIR false

#define INVERT\_E4\_DIR false

Es posible que tengas que regresar a esta sección varias veces hasta que aciertes con los sentidos correctos.

**Configuración de paso de los motores stepper**

La siguiente sección consiste en calibrar los ejes de los diferentes motores, indicando los pasos que gira cada motor por unidad de desplazamiento (Marlin emplea “mm”). Esto requiere algunas matemáticas y un conocimiento consistente del hardware de la impresora aunque, por suerte, los archivos de ejemplo nos vendrán muy bien.

En general, los valores que hemos de introducir en la variable obedecen a la siguiente fórmula:

En mi caso, conservo el hardware de los ejes de movimiento, de manera que podemos tomar como referencia los datos del archivo de configuración que Marlin nos proporciona sobre la Anet A8. Sin embargo, esto no es cierto para el extrusor, ya que lo he sustituido por el E3D Titan, a cuya página te dirijo para los detalles sobre la configuración de Marlin, en el siguiente enlace:

<https://e3d-online.dozuki.com/Guide/Titan+Marlin+Configuration/35?lang=en>

La Anet A8 utiliza un microstepping de 1/16 en todos los motores, y es el valor por defecto que espera el extrusor E3D Titan. Combinando los valores del archivo de ejemplo y de la página anterior, la variable quedará de la siguiente manera:

#define DEFAULT\_AXIS\_STEPS\_PER\_UNIT { 100, 100, 400, 419 }

No obstante, si especificamos un ratio de 1/32, habrá que multiplicar por dos los valores aquí indicados, de modo que pueden ser:

#define DEFAULT\_AXIS\_STEPS\_PER\_UNIT { 200, 200, 800, 837 }

**Nota:** no descartes tener que corregir estos valores más de una vez. Haz unas cuantas pruebas hasta que des con la combinación correcta.

**Final de las modificaciones**

Una vez hayas completado todos los pasos, ya es posible compilar y subir el firmware a la placa, lo que puedes hacer pulsando el botón con la flecha apuntando a la derecha. Si quieres comprobar que no haya errores, a la izquierda de ese botón encontrarás otro que te permite hacerlo antes de subirlo (aunque el IDE lo comprueba por defecto antes de compilar).

**Configuración del servidor de impresión (OctoPrint)**

1. Descarga la imagen de OctoPrint en:

<https://octoprint.org/download/>

1. Quema la imagen en la tarjeta SD.
2. Introduce la tarjeta SD en las Raspberry y arranca.
3. Configura la red para que sea accesible desde el equipo principal.
4. Conecta la Raspberry a la Arduino.
5. Configura la impresora desde el panel de administración web.