

[https://drive.google.com/open?id=1uSQPGRAL3yldkq9UsGQ\\_ZXc1nvzyNynU](https://drive.google.com/open?id=1uSQPGRAL3yldkq9UsGQ_ZXc1nvzyNynU)

## INDICE:

- [Info Marlin pre-configurados.](#)
- [Procedimiento de flasheado.](#)
- [Recomendaciones tras flashear nuevo Marlin.](#)
  - [LCD.](#)
  - [LCD/TFT BTT V2-3.0.](#)
  - [TMC connection error.](#)
  - [Sentido de giro de ejes.](#)
  - [Ajuste PIDs.](#)
  - [Finales de carrera.](#)
  - [Ajuste del área de impresión.](#)
  - [Steps por mm.](#)
  - [Ajuste de miliamperios y modo stealchop \(TMC\).](#)
  - [Ajuste velocidades home y/o autolevel.](#)
- [Sensor de auto nivelado.](#)
- [Bl touch. \(mirar primero el punto anterior\).](#)
- [Finales de carrera con TMC2130/2209.](#)
- [Sensorless.](#)
- [Doble Z.](#)
- [G34 Auto alineamiento de Z.](#)
- [Sensor de filamento.](#)
- [Extrusor BMG o Titan.](#)
- [UBL.](#)
- [Z offset.](#)
- [EPROM EN PLACAS SIN MODULO INTEGRADO.](#)
- [Videos.](#)

## INTRODUCCIÓN:

- Marlin 2.0.0 Estable con fecha aprox del 12 de diciembre.
- Las carpetas Marlin originales tal como han sido descargadas de la página de Marlin (bugfix 2.0.0 como versión 2.0.1) se encuentran en un archivo zip en la carpeta raíz del drive.

- ***En caso de querer cargar la versión 2.0.1 es posible usar los archivos de configuración ya existentes procediendo como sigue:***

***-Descargar la versión 2.0.1 de la página de Marlin.***

***- Extraer el Zip en el escritorio.***

***-Substituir sobre la carpeta de marlin 2.0.1 los archivos siguientes de la versión 2.0.X bugfix:***

***\* platformio.ini***

***\* Config.h***

***\* Config.adv***

***Sobre este último habrá que realizar las modificaciones siguientes:***

***Cambiar la línea original:***

***1-"FILAMENT\_UNLOAD\_RETRACT\_LENGTH"***

***2-"FILAMENT\_UNLOAD\_DELAY"***

***Por:***

***1-"FILAMENT\_UNLOAD\_PURGE\_RETRACT"***

***2-"FILAMENT\_UNLOAD\_PURGE\_DELAY"***

- A menos que se indique otra cosa en el nombre de la carpeta todos los ejemplos vienen preconfiguradas con TMC2209.
- En cada carpeta se incluyen los archivos de configuración así como el correspondiente bootscreen y statusscreen si corresponde + el firmware.bin directo para flashear según dicha configuración + la carpeta marlin completa. correspondiente incluyendo todo (firmware.bin también).
- En el caso de querer modificar el firmware partiendo de una carpeta de los ejemplos se recomienda en primer lugar probar de compilar tal y como esta y asegurarse que se completa con éxito la compilación procediendo como sigue:

- 1- descargas el zip
- 2- extraes la carpeta en el escritorio
- 3- Abres la carpeta del proyecto en VS
- 4- Seleccionar algun archivo del proyecto, el configuration.h por ejemplo
- 5- pulsas compilar

Si siguiendo los pasos tal y como se ha indicado no se finaliza la compilación con éxito lo más probable es que tengáis un problema con VS, me podéis contactar para comprobarlo.

Este seria el punto de partida ya que con esto nos aseguramos de que la carpeta es correcta y de que VS compila correctamente, así si posteriormente a realizar modificaciones

hay errores de compilación sabemos que esta directamente relacionado con lo que se ha modificado.

- En todos los ejemplos se incluyen las opciones siguientes:
  - LCD en castellano.
  - PID para hotend y bed.
  - Ajustes según ejemplos base y algunos según se ha facilitado algún compañero tras hacer reajustes en su máquina (como por ejemplo el PID de la cama en la ender 3).
  - PID tuning menu.
  - Endstops menu.
  - Individual home axis menu.
  - S-curve.
  - Junction deviation.
  - Manual mesh menu en el caso de no disponer de sensor de nivelado
  - UBL y safe\_homing en el caso de disponer de sensor de nivelado.
  - Eeprom activada.
  - Bed thermal protection con ajustes menos agresivos para intentar evitar falsos positivos.
  - Modo Stealchop por defecto para todos los ejes, si es el caso que no son TMC2209 entonces el extrusor viene configurado en spreadcycle por defecto.
  - TMC debug.
  - Linear advance activado a 0 para poder activarlo con el comando M900 Kx desde el script de inicio a voluntad.

- TMC monitor (no en todos).
- Aumentado el buffer para evitar retardos en el LCD.
- SKR 1.3 función auto\_fan activado, si se desea conectar el ventilador en HE1.
- Custom commands menu en el LCD.
- 32 microsteps interpolados.
- Juegos.

-

SE RECOMIENDA FLASHEAR COMO PRIMER PASO A REALIZAR TRAS RECIBIR LA PLACA, ANTES DE REALIZAR NINGUN MONTAJE NI CONEXIÓN. PREFERIBLEMENTE ALIMENTAR LA PLACA VIA USB.

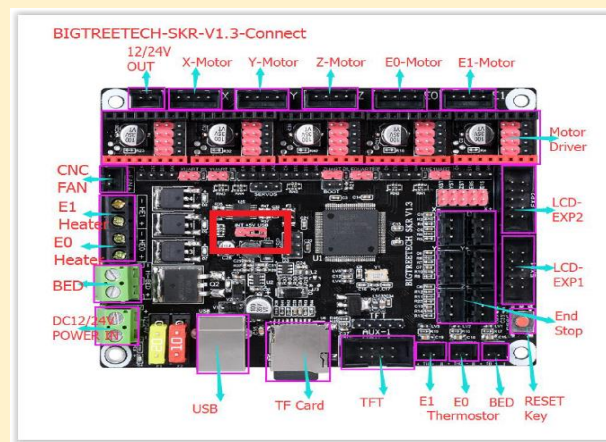
\*\*\* Todo lo señalado a continuación como operaciones a realizar via pronterface es igualmente posible desde el terminal de octoprint.

#### PROCEDIMIENTO DE FALSHEADO:

- 1- Extraer la micro SD de la placa e introducirla en el ordenador.
- 2- Copiar el archivo FIRMWARE.BIN en la raíz de la micro SD.

- 3- Introducir la micro SD en la placa y conectar alimentación, bien sea a través de USB o bien alimentación externa. (recordar la posición del jumper de alimentación si aplica ver foto).
- 4- Esperar unos segundos a que el proceso se complete, extraer la micro SD y comprobar la tarjeta en el ordenador, si el proceso se ha completado con éxito el archivo se debería haber renombrado a FIRMWARE.CUR.

***En el caso de que surja algún problema y el proceso no se complete seria conveniente probar con otra tarjeta micro SD tras haberla formateado en FAT32.***



## Pasos recomendados a seguir tras flashear un nuevo firmware:

TRAS FLASHEAR UN NUEVO MARLIN ES COVENIENTE PROSEGUIR CON UN M502 SEGUIDO DE M500 DESDE PRONTERFACE O BIEN FACTORY RESET Y POSTERIORMENTE SAVE SEETINGS DESDE EL LCD

### - LCD:

Si el LCD está definido correctamente en Marlin se han de girar 180º ambos conectores del LCD en uno solo de los extremos.

Si los conectores del LCD (EXP1 y EXP2) estuvieran intercambiados no se daña nada ni en el LCD ni en la placa, simplemente no se enciende.

En el caso de que el LCD se encienda, pero no se pueda leer nada habría que ajustar el contraste, otra nota es que cuando está bien conectado se notan los pitidos al girar y pulsar el encoder, generalmente todos los LCD tienen un pequeño potenciómetro en la parte posterior, girando dicho potenciómetro con un destornillador estrella muy fino es posible ajustar hasta que se pueda leer correctamente.

En la sección de videos he dejado un link a un video de muestra de todo lo mencionado.

- **LCD/TFT V2-3.0**

Ejemplo de conexionado V3.0: Esta pantalla tiene ambos modos LCD y TFT por lo que es necesario conectar los 3 cables EXP1/2/3.





### Ejemplo de conexionado V2.0:



V2-3.0: Para que funcione el modo TFT solo es preciso (por parte de marlin) que estén definidos correctamente los puertos de serie y que la velocidad del bus coincida tanto en Marlin como en la configuración del TFT.

Recomendaría seleccionar 115000 baudios en ambos lados, desde el menú del TFT y en Marlin como sigue:

MENU TFT:



MARLIN:

-Para una SKR 1.3:

```
#define SERIAL_PORT 0  
#define SERIAL_PORT_2 -1  
#define BAUDRATE 115200
```

Por otra parte será necesario también definir el tipo correcto de LCD en Marlin como sigue:

```
#define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER
```

Y comentar con // cualquier otro tipo de LCD que pudiese haber definido con anterioridad (si lo hubiese).

#### OTROS MODELOS DE PLACAS:

- SKR 1.4:

```
#define SERIAL_PORT -1  
#define SERIAL_PORT_2 0
```

- SKR MINI E3 V1.2:

```
#define SERIAL_PORT 2  
#define SERIAL_PORT_2 -1
```

- SKR MINI V1.1:

```
#define SERIAL_PORT -1  
#define SERIAL_PORT_2 1
```

- SKR E3 DIP

```
#define SERIAL_PORT 2  
#define SERIAL_PORT_2 -1
```

- SKR PRO 1.1

```
#define SERIAL_PORT -1  
#define SERIAL_PORT_2 1
```

- **TMC connection error:**

Comprobar que los drivers comunican correctamente con M122 desde pronterface:

<http://marlinfw.org/docs/gcode/M122.html>

<https://www.pronterface.com/>

Que veremos si hay un fallo de comunicación de alguno/s de los drivers UART y/o SPI en el LCD?



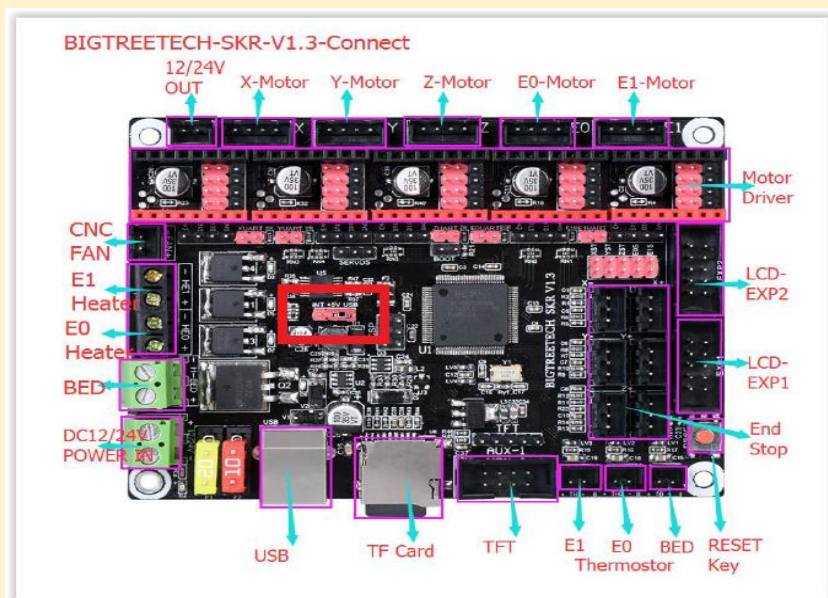
Que esperamos ver si todos los drivers están comunicando correctamente via pronterface?

```
Recv: Testing X connection... OK
Recv: Testing Y connection... OK
Recv: Testing Z connection... OK
Recv: Testing Z2 connection... OK
Recv: Testing E connection... OK
Recv: ok
```

Comprobaciones a realizar en caso de que alguno/s no comuniquen:

- 1- Los drivers solo comunican de disponer de la alimentación principal de la placa (no usb).
- 2- En caso de disponer de jumper selector de alimentación este debe encontrarse en posición +5VD.

Ejemplo:



- 3- Comprobar que los jumpers UART/SPI(STANDALONE se encuentran en la posición correcta.
- 4- Comprobar que los drivers están preparados para trabajar en modo SPI/UART ya que algunos drivers no vienen preparados de origen y hay que soldar unos puentes en la cara inferior del drive.

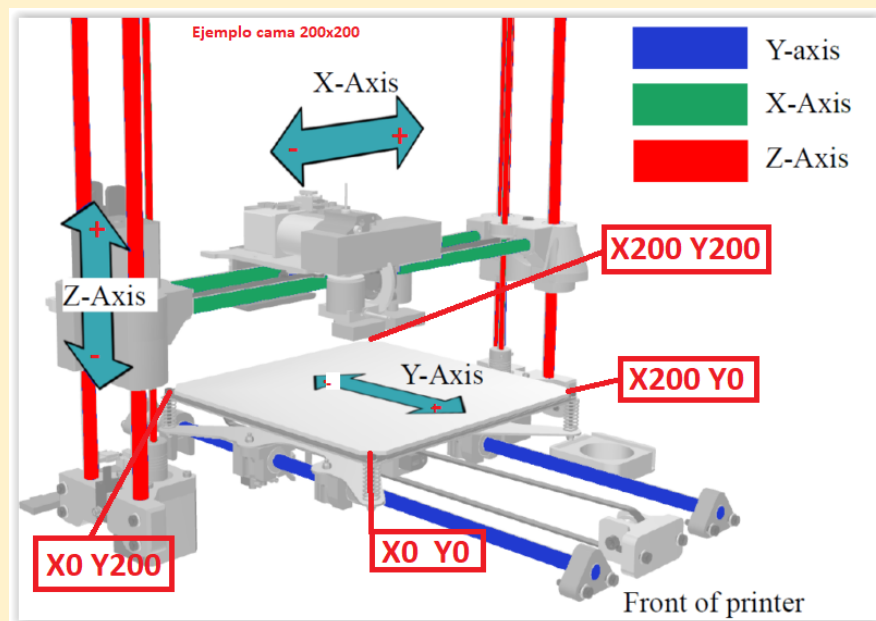
Nota: Drivers en SPI (TMC2100,2130,5160) ha de estar habilitado en Marlin la línea siguiente:

```
#define TMC_USE_SW_SP
```

- **Sentido de giro de los ejes:**

Mover ejes XYZ manualmente y comprobar si la dirección de movimiento de cada eje corresponde con lo esperado.

Normalmente en una máquina cartesiana :



- Calentar boquilla
- Mover E manualmente y comprobar si el sentido de giro es el correcto.

Si alguno es incorrecto se puede girar el conector del stepper 180º o bien modificar en Marlin cambiando true/false

```
// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.  
#define INVERT_X_DIR true  
#define INVERT_Y_DIR true  
#define INVERT_Z_DIR false
```

```
// For direct drive extruder v9 set to true, for geared extruder set to false.  
#define INVERT_E0_DIR false
```

Para ver como invertir el giro de un stepper sin necesidad de modificar Marlin [pulsa AQUI](#)

- **Ajuste PIDs:**

Calentar cama. Ajuste de PID si fuese necesario

<http://marlinfw.org/docs/gcode/M303.html>

El ajuste del PID de la boquilla se puede realizar desde el LCD sin embargo la cama se ha de realizar por fuerza desde consola, tras tener los valores correctos hay que guardar los cambios con M500 para que no se pierdan tras un reinicio.

Auto ajuste PID:

- Hotened:

- M303 E0 S210 C8
- Tras esto nos devolverá unos valores de PID que se pueden guardar con:
  - M301 Px Ix Dx (donde X es el valor que nos ha indicado).
- M500 para guardar cambios en la Eprom.

- Cama:

- M303 E-1 S60 C8.
- Tras esto nos devolverá unos valores de PID que se pueden guardar con:
  - M304 Px Ix Dx (donde X es el valor que nos ha indicado).
- M500 para guardar cambios en la Eprom.

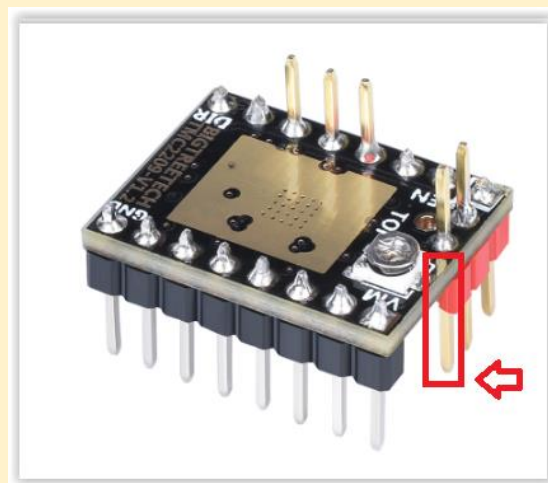


- **Home y finales de carrera:**

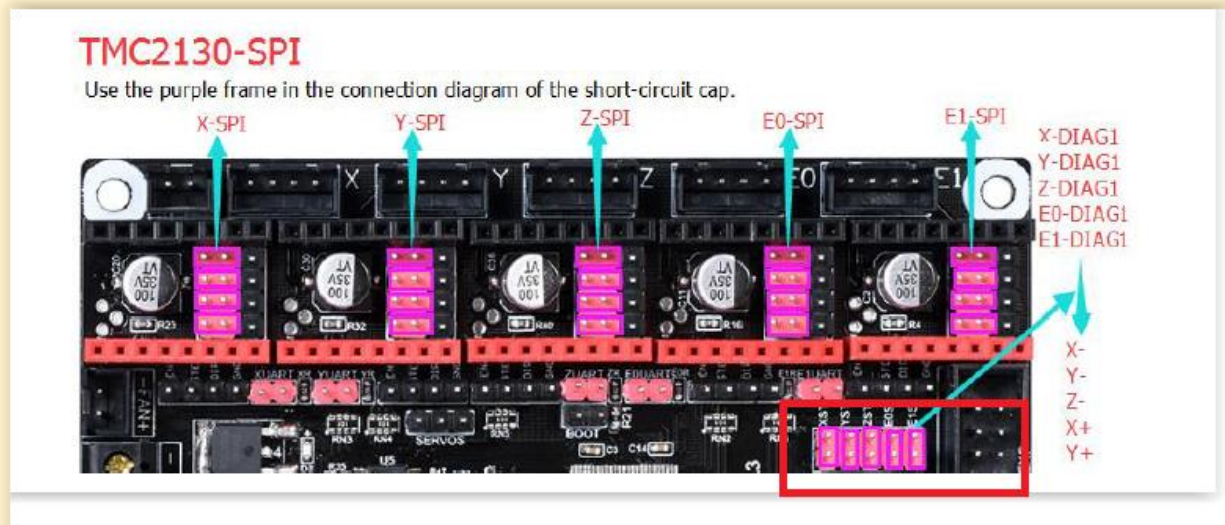
Hacer home (G28). En el caso de sensor inductivo o Bltoch debería realizar el home de Z en el centro de la cama.

-

- M119 – Para comprobar desde pronterface el estado de los endstop, debería marcar “open” cuando se encuentran libres y “triggered” cuando se pulsan.
- En el caso de sensorless debería de marcar siempre “open” en reposo.
- Ajuste de sensibilidad en caso de usar sensorless en XY desde el LCD (TMC2130-2209-5160).
- Para trabajar con finales de carrera como “endstops” en otras placas que no sea la SKR 1.3 con los drivers TMC2130/2209 o 5160 habrá que doblar o cortar el pin “diag” del driver antes de introducirlo en el zócalo.



En el caso de la SKR 1.3 habrá que quitar los jumpers de los ejes que corresponda.



Que esperamos recibir tras mandar el comando M119 si no están activados los endstops?

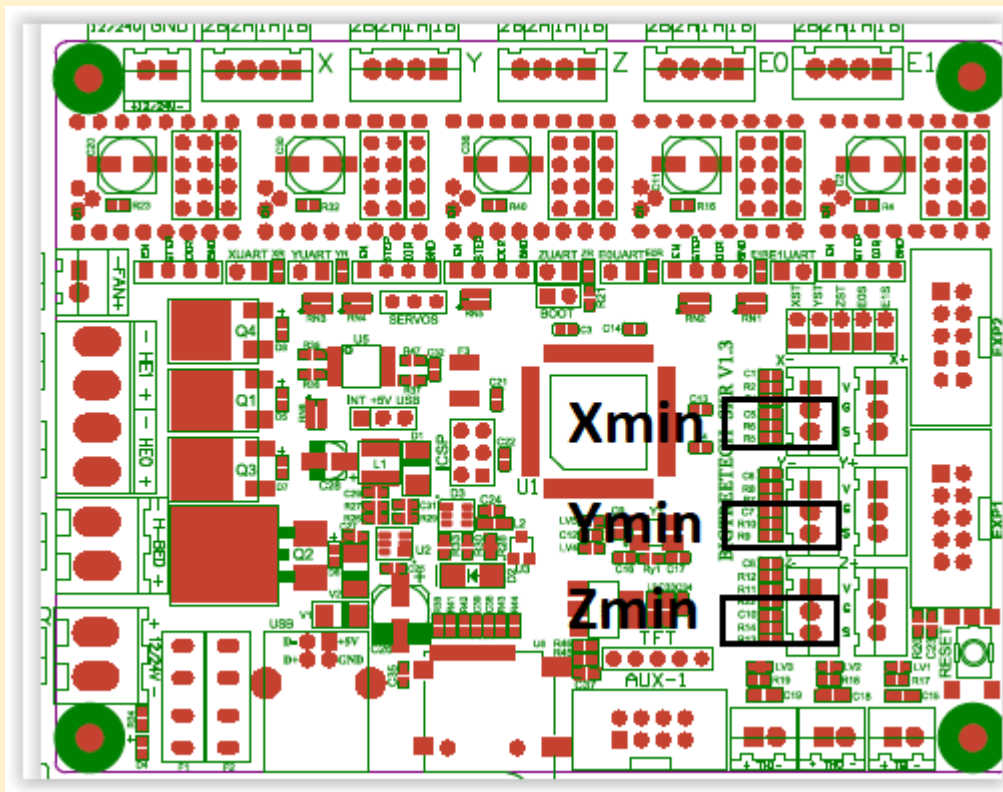
```
Recv: Reporting endstop status
Recv: x_min: open
Recv: y_min: open
Recv: z_min: open
Recv: filament: TRIGGERED
Recv: ok
```

<http://marlinfw.org/docs/gcode/M119.html>

En el caso de tener que modificar la lógica cambiar True/false en el endstop del eje correspondiente según corresponda.

```
// Mechanical endstop with COM to ground and NC to Signal us
es "false" here (most common setup).
#define X_MIN_ENDSTOP_INVERTING false
#define Y_MIN_ENDSTOP_INVERTING false
#define Z_MIN_ENDSTOP_INVERTING false
#define X_MAX_ENDSTOP_INVERTING false
#define Y_MAX_ENDSTOP_INVERTING false
#define Z_MAX_ENDSTOP_INVERTING false
#define Z_MIN_PROBE_ENDSTOP_INVERTING false
```

Nota: los finales de carrera se han de conectar entre GND y Signal, en el caso de una SKR 1.3 que tiene 3 pins en cada conector tendrá que ir como sigue:



- **Ajuste del área de impresión:**

Desplazar XY a 0 y comprobar si corresponde la posición de la boquilla con la esquina frontal-izquierda de la cama (pronterface “G1 X0 Y0 F5000”) en el caso de no corresponderse mover manualmente en ajuste mm a mm y encontrar la desviación en X y en Y respecto a la posición esperada, Habrá que sumar o restar dicho valor en:

Nota: “M211 S0” o deshabilitar end stops desde el LCD para poder mover más allá de límites de software y buscar las coordenadas correctas.

```
// Travel limits (mm) after homing, corresponding to endstop positions.  
#define X_MIN_POS  
#define Y_MIN_POS
```

Nota: Otras opciones ver M206 – Set home offsets.

- **Steps/mm, desplazamientos:**

Comprobar que el desplazamiento XY se corresponde con lo esperado, si la cama es de 220X220 con el comando “G1 X220 Y220 F5000” la boquilla debería posicionarse en la esquina posterior derecha de la cama.

- Si no coincide hay que revisar si los steps por mm corresponden a la configuración de la máquina según la definición de microsteps a que este seleccionado dicho eje.

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 160, 160, 800, 186 }
```

Nota: Ajuste desde pronterface con M92:

```
M92 [E < steps >] [T < index >] [X < steps  
>] [Y < steps >] [Z < steps >]
```

Ejemplo: M92 E205

- Comprobar calibración de extrusión y ajuste de pasos si fuese necesario. (YA hay muchas guías explicando cómo realizar este paso ejemplo:  
<https://www.youtube.com/watch?v=JJLCAdAKPkg>).
- <https://mattshub.com/blog/2017/04/19/extruder-calibration>
- **Stealchop/Spreadcycle (solo TMC UART/SPI) y ma:**

Comprobar que el valor actual corresponde con lo deseado en el modo de trabajo de los steppers (stealthchop/spreadcycle) y el amperaje max de los steppers.

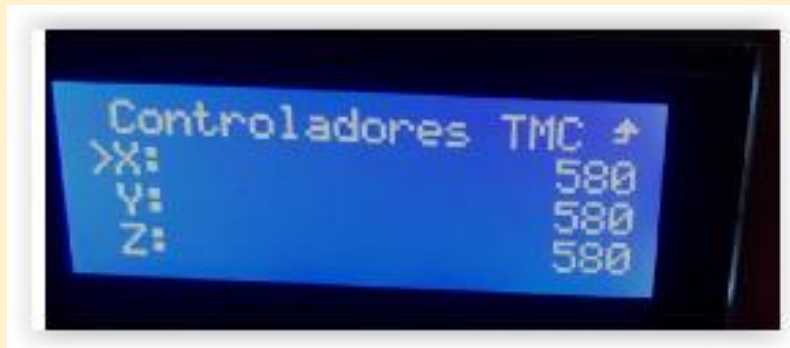
Para TMC2130 o TMC2208 se recomienda dejar el extrusor en Spreadcycle (deshabilitar stealchop desde el LCD).

Fórmula para calcular el valor max de miliamperios:

Hay que restar al menos un 10% de seguridad al resultado de dividir la corriente max del stepper según especificaciones entre 1.42

Ejemplo de un stepper de 1,5 Amp:  $(1500/1.42) - 10\% = \text{aprox } 850\text{-}900 \text{ ma}$

Ajuste desde el LCD en la ruta /configuracion/ajustes  
TMC/controladores TMC/amperaje control



Desde Marlin ajuste en:

Apartado:

```
#if HAS_TRINAMIC
```

Valor:

```
#define X_MAX_CURRENT  
#define Y_MAX_CURRENT  
#define Z_MAX_CURRENT
```

Si aplica:

```
#define Z2_MAX_CURRENT
```

Nota: TMC5160 hay que cambiar el Rsense default de:

```
#define X2_RSENSE          0.11
```

a:

```
#define X_RSENSE           0.075
```

Habilitar modo stealthchop desde pronterface: "M569 S1 X Y Z E"

Habilitar modo spreadcycle desde pronterface: "M569 S0 X Y Z E"

- Nivelación de cama y Z offset (G29 si aplica)
- Prueba de impresión (benchy, test cube).
- 

Links: Descarga Marlin website: <http://marlinfw.org/meta/download/>

Gcodes Explanation: <http://marlinfw.org/docs/gcode/G000-G001.html>

Config guides:

<http://marlinfw.org/docs/configuration/configuration.html>

**-Ajuste velocidades de home o de auto nivelado:**

El ajuste de las velocidades de los ejes cuando hace el home se realiza modificando los valores en:

```
#define HOMING_FEEDRATE_XY (20*60)  
#define HOMING_FEEDRATE_Z (4*60)
```

El ajuste de las velocidades durante el auto nivelado se realiza modificando los valores en:

```
#define XY_PROBE_SPEED 6000
```

La velocidad del eje Z durante el autolevel va originalmente referenciada a la velocidad del eje Z durante el home:

```
#define Z_PROBE_SPEED_FAST HOMING_FEEDRATE_Z
```

```
#define Z_PROBE_SPEED_SLOW (Z_PROBE_SPEED_FAST / 2)
```

Habr  seguramente m s cosas, pero as  a grandes rasgos esto es lo principal a parte de lo t pico de aceleraciones, jerk o junction deviation, K en linear advance etc...



## Como activar sensor de nivelado con un inductivo partiendo de la base de un Marlin que no lo tiene configurado?

A- Habilitar Fix mount probe:

```
#define FIX_MOUNTED_PROBE
```

B- Modificar los offset del sensor según el carro instalado en máquina.

```
/**
 * Offset de Sensor de nivelado Z a boquilla (X,Y), respecto a (0, 0).
 *
 * En el ejemplo siguiente los "offsets" para X e Y son ambos valores en
 * positivo:
 *
 * #define NOZZLE_TO_PROBE_OFFSET { 10, 10, 0 }
 *
 * +-- ATRAS ---+
 * |               |
 * I |      (+) D   | D <-- Sensor (20,20)
 * Z |               | E
 * Q |      (-) B (+) | R <-- Boquilla (10,10)
 * |               |
 * |      (-)       |
 * |               |
 * O-- FRONTAL --+
 * (0,0)
 *
 * Indicar la posición como; { X, Y, Z }
 */
```

```
#define NOZZLE_TO_PROBE_OFFSET
```

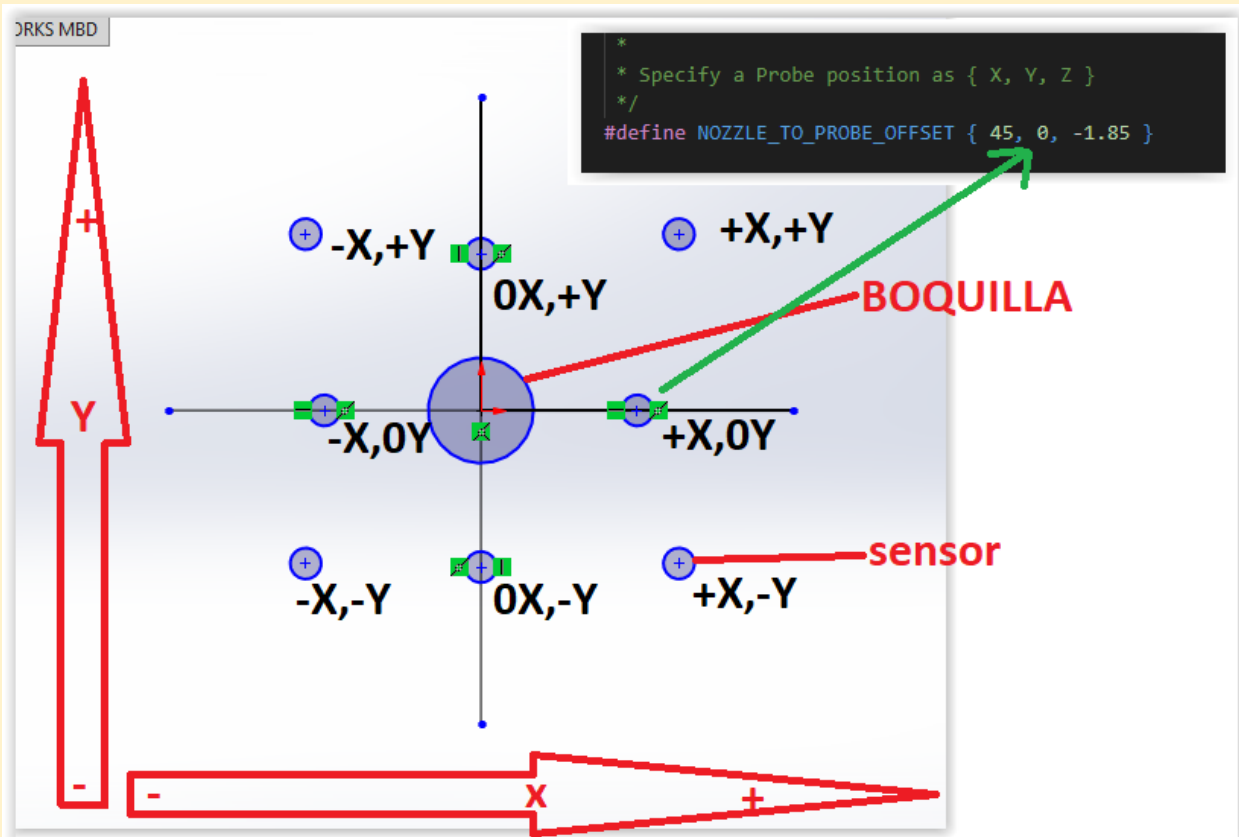
\*Nota: Este valor se puede modificar desde pronterface sin necesidad de modificar y flashear nuevamente marlin con: “ M861 X(medida) Y(medida) Z (medida) “ indicando todos o solo los ejes necesarios a modificar, por ejemplo “ M851 X50 Z-2.3 “

\*Nota 2: Para guardar cambios en Eeprom realizar un M500 después de cualquier modificación de valores.

Para determinar la posición del sensor respecto a la boquilla se puede medir con un pie de rey la distancia o bien marcando un punto en la cama donde se encuentra el centro de la boquilla y desplazando en ajuste manualmente hasta centrar sobre dicha marca el sensor, lo que hayamos desplazado en XY serán los offset.

Esto aplica para cualquier tipo de sensor de nivelado.

Para clarificar el signo de los offset referirse al grafico siguiente donde el circulo grande central es la boquilla y los círculos pequeños son posibles posiciones del sensor de nivelado:



C.- Activar:

```
#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
```

C- Desactivar MESH\_BED\_LEVELING y activar UBL

```

#define AUTO_BED_LEVELING_3POINT
// #define AUTO_BED_LEVELING_LINEAR
// #define AUTO_BED_LEVELING_BILINEAR
#define AUTO_BED_LEVELING_UBL
// #define MESH_BED_LEVELING

```

D- Activar restore after homing.

```
#define RESTORE_LEVELING_AFTER_G28
```

E- Habilitar SAFE\_HOMING para que realice el home de Z en el centro de la cama.

```
#define Z_SAFE_HOMING
```

F- Comprobar los ajustes siguientes:

```
#define Z_CLEARANCE_DEPLOY_PROBE    10 // Z Clearance for Deploy/Stow
#define Z_CLEARANCE_BETWEEN_PROBES  5 // Z Clearance between probe points
#define Z_CLEARANCE_MULTI_PROBE     5 // Z Clearance between multiple probes
#define Z_AFTER_PROBING              5 // Z position after probing is done
```

Más info: <http://marlinfw.org/docs/gcode/G029-ubl.html>

## Como activar Bltoch partiendo de la base de uno que tiene configurado sensor de nivelado? (nota sino hay que empezar por el punto anterior)

A- Desactivar Fix\_probe:

```
//#define FIX_MOUNTED_PROBE
```

B- Activar BLtouch:

```
#define BLTOUCH
```

Nota: Con M119 se espera que Z\_MIN reporte estado “triggered” con el pin introducido y “open” con el pin extraído.

Debido a esto seguramente habrá que invertir la señal respecto a lo que seria habitual en un final de carrera o un inductivo, hay que dejar configurados los endstop de Z como sigue:

```
#define Z_MIN_ENDSTOP_INVERTING true  
#define Z_MIN_PROBE_ENDSTOP_INVERTING true
```

## GCODES UTILES PARA TEST:

RESETEAR BL TOUCH M280 P0 S160

AUTO-TEST M280 P0 S120

INTRODUCIR PIN M280 P0 S90

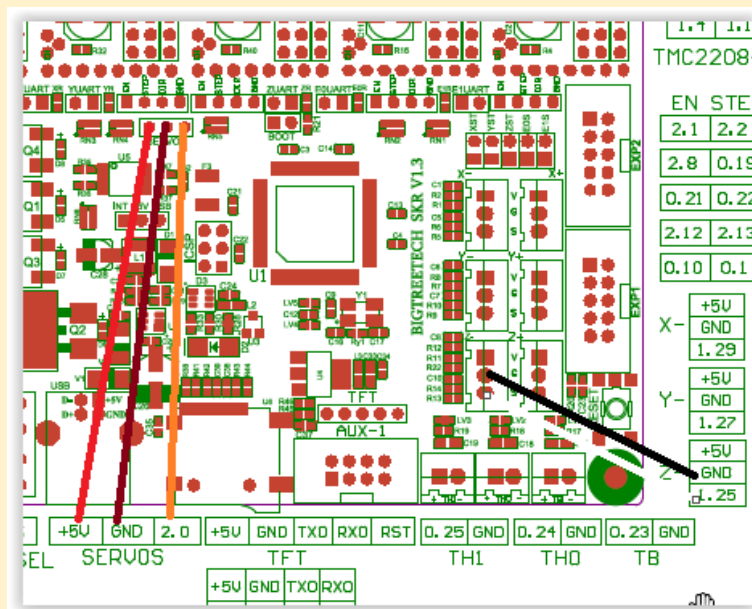
EXTRAER PIN M280 P0 S10

Nota2: En caso de comportarse a la inversa al mandar comando para extraer o introducir el pin probar a intercambiar el hilo rojo (+5VDC) por el naranja (Signal).

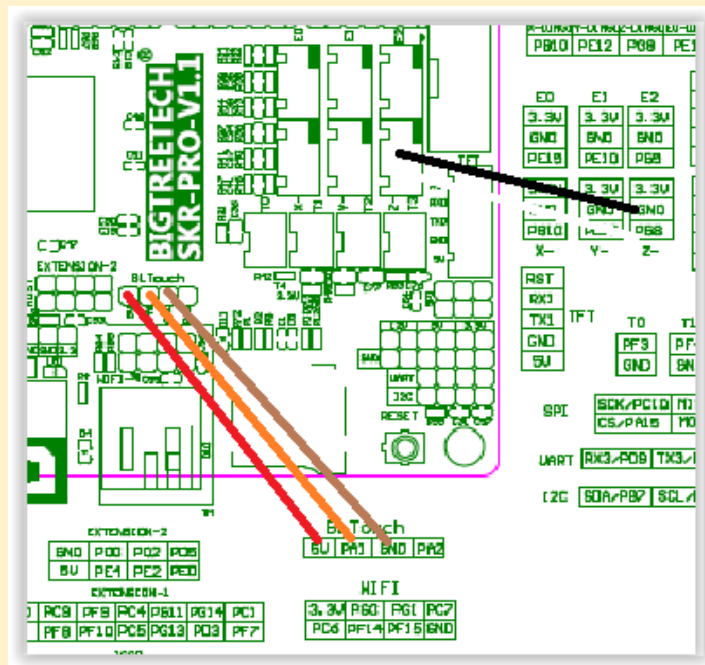
Nota3: De tener problemas de respuesta existe la posibilidad de programar un retardo en marlin en config.adv:

```
//#define BLTOUCH_DELAY 500
```

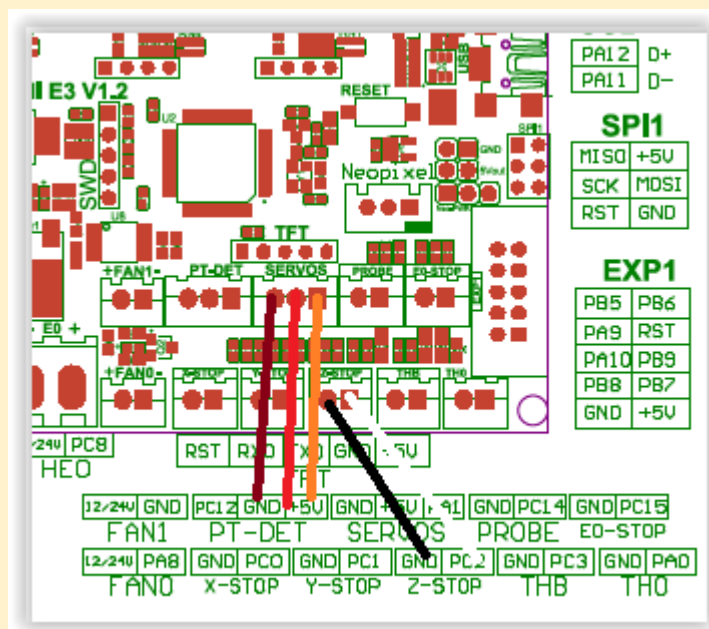
Conexionado en SKR 1.3



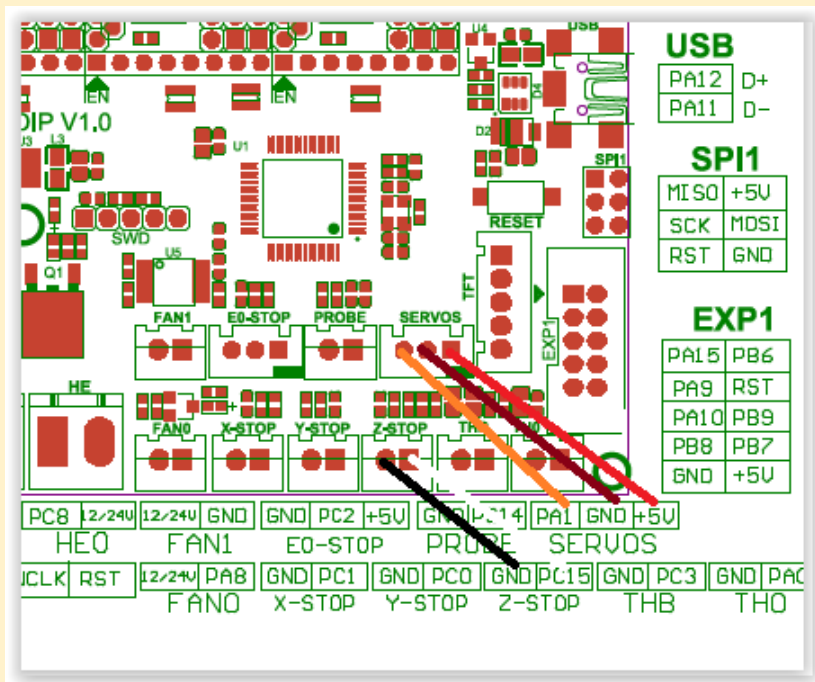
Conexionado en SKR PRO 1.1:



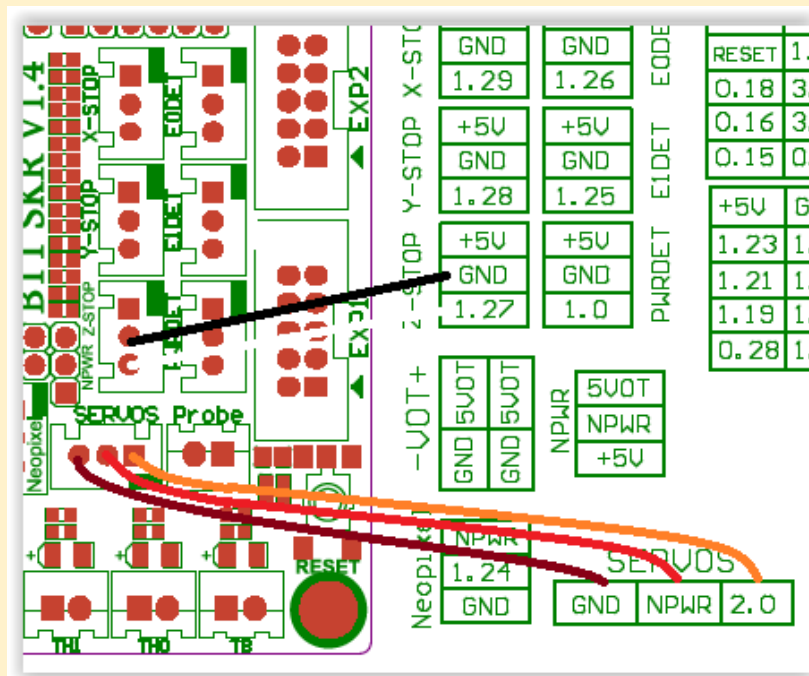
Conexionado en SKR Mini E3 V1.2:



Conexionado en SKR E3 DIP V1.0;



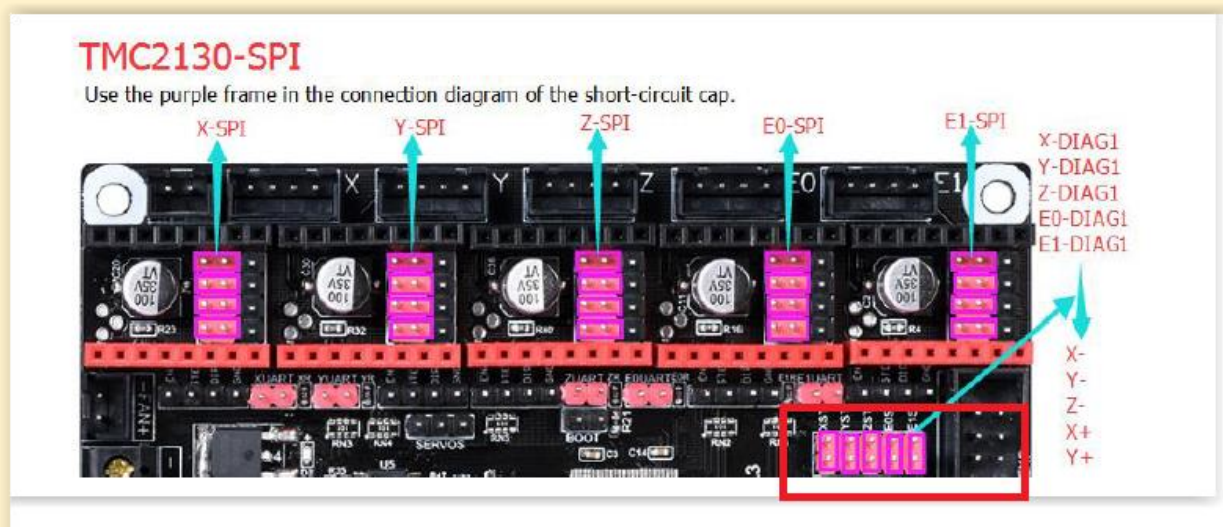
Conexionado en SKR 1.4 y 1.4 PRO:



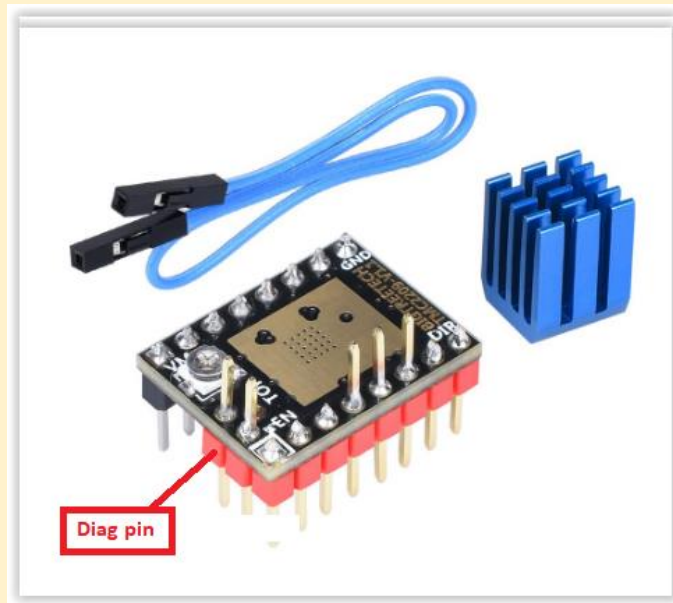


## Uso de finales de carrera físicos como endstops con TMC2130/2209

En la SKR 1.3 será necesario quitar los jumpers de los ejes correspondientes en:



En el resto de placas habrá que doblar o cortar el pin diag del driver.



## Como activar sensorless partiendo de la base de uno que no lo tiene configurado? (TMC2209-2130-5160)

A- Los endstop para XY han de estar en inverting = False sea cual sea la impresora en el caso del TMC2209 y en True de usar TMC2130 o TMC5160:

```
#define X_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.  
#define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of the endstop.
```

B- Activar sensorless homing en config.advanced:

```
#define SENSORLESS_HOMING // StallGuard capable drivers only
```

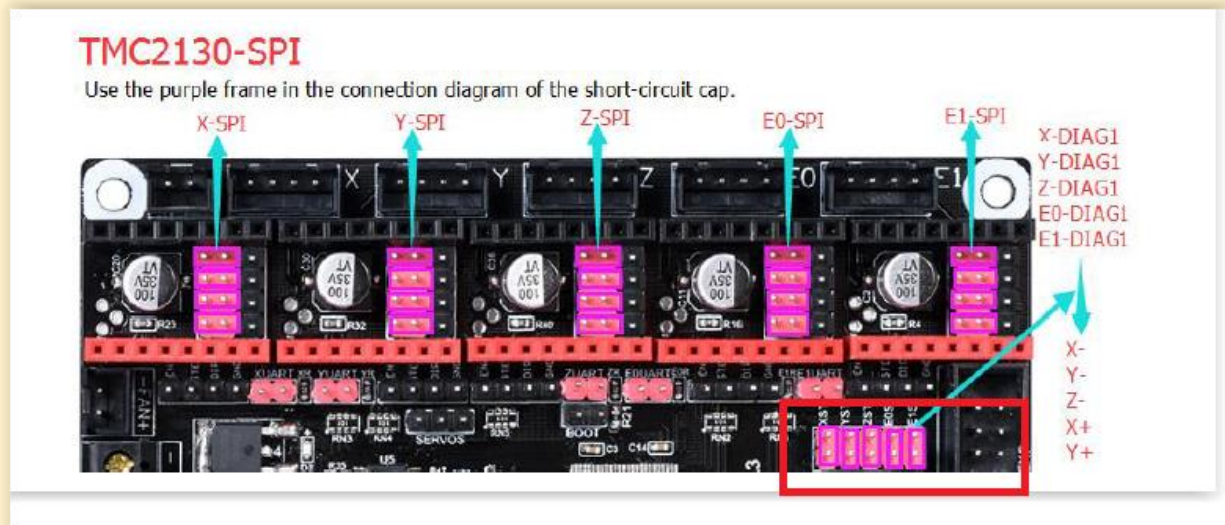
C- Otros ajustes (en función del driver, la sensibilidad se puede ajustar posteriormente desde el LCD)

```
#define X_STALL_SENSITIVITY 50
#define X2_STALL_SENSITIVITY X_STALL_SENSITIVITY
#define Y_STALL_SENSITIVITY 50
//#define Z_STALL_SENSITIVITY 8
//#define SPI_ENDSTOPS // TMC2130 only
//#define HOME_USING_SPREADCYCLE
#define IMPROVE_HOMING_RELIABILITY
```

D- Ajustar el BUMP a 0 para XY:

```
#define X_HOME_BUMP_MM 0
#define Y_HOME_BUMP_MM 0
```

Nota: En la SKR 1.3 es necesario instalar los jumpers en los ejes en que se vaya a activar sensorless.

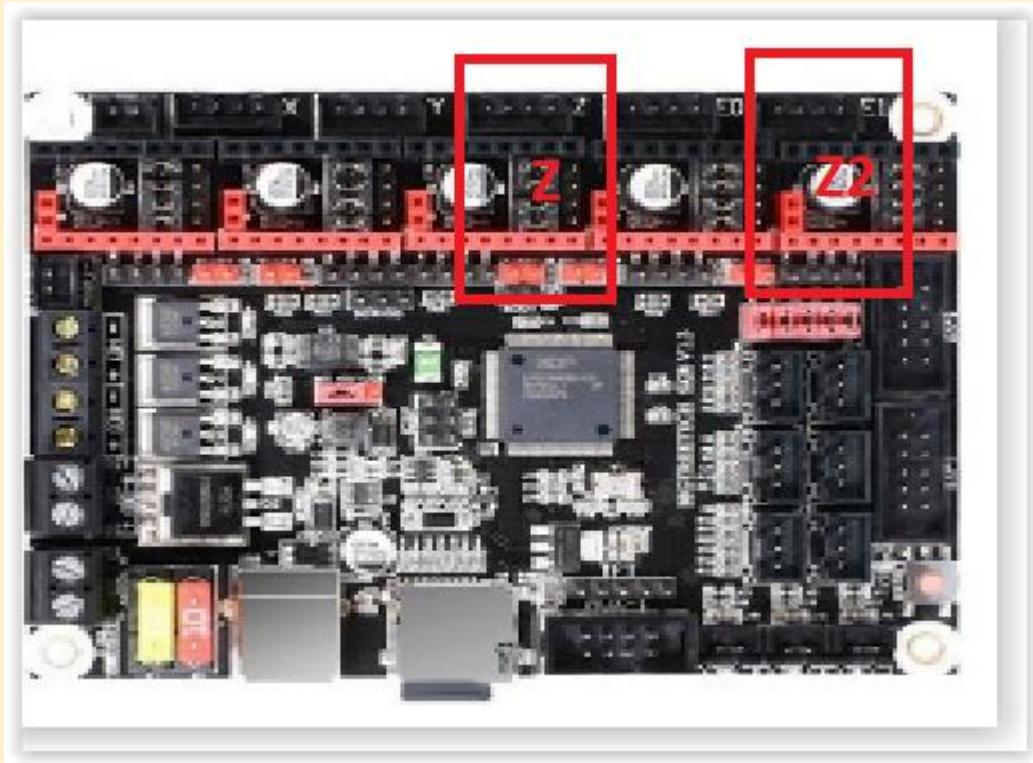


## Doble Z:

Esto solo aplica en el caso de montar drivers independientes para cada stepper de Z.

En el caso de montar ambos motores cogidos al mismo driver con un “spliter” no hay que configurar nada en Marlin.

Ejemplo en SKR 1.3.



Ejemplo de “splitter” para poner los dos steppers en el mismo driver sin configurar dual Z.



A- Habilitar Z2 y definir el driver.

```
#define Z2_DRIVER_TYPE TMC2209
```

B- Habilitar dual Z:

```
#define Z_DUAL_STEPPER_DRIVERS
```

C- Ambos ejes deberían de estar configurados con el mismo modo de microsteps y de ma (en principio).

Sección:

```
#if HAS_TRINAMIC
```

Ajustes:

```
#if AXIS_IS_TMC(Z)
#define Z_CURRENT      850
#define Z_CURRENT_HOME Z_CURRENT
```

```

#define Z_MICROSTEPS      16
#define Z_RSENSE           0.11
#define Z_CHAIN_POS       -1
#endif

#if AXIS_IS_TMC(Z2)
#define Z2_CURRENT         850
#define Z2_CURRENT_HOME   Z2_CURRENT
#define Z2_MICROSTEPS      16
#define Z2_RSENSE           0.11
#define Z2_CHAIN_POS       -1

```

D- Si se desean usar finales de carrera hay que habilitar el Segundo final de Carrera, en donde se conecta y si aplica algún offset de software:

```

#define Z_DUAL_ENDSTOPS
#if ENABLED(Z_DUAL_ENDSTOPS)
#define Z2_USE_ENDSTOP _XMAX_
#define Z_DUAL_ENDSTOPS_ADJUSTMENT 0

```

E- Habilitar auto ajuste de Z y sus ajustes correspondientes:

```

#define Z_STEPPER_AUTO_ALIGN

```

Definir los puntos de test en función del tamaño de la cama y del offset del sensor de nivelado ya que los puntos han de ser accesibles para el sensor, por ejemplo si XY\_MIN=0 XY\_MAX=220 Probe offset=(50,0,0) Edge=10

```

#define Z_STEPPER_ALIGN_XY { { 50, 110 }, { 210, 110 } }

```

Subir el máximo de itinerancias ya que originalmente es demasiado bajo a mi parecer.

```
#define Z_STEPPER_ALIGN_ITERATIONS 15
```

Reducir la precisión si se desea:

```
#define Z_STEPPER_ALIGN_ACC 0.05
```



## G34 auto-alineamiento de los husillos en Z

El auto alineamiento de husillos se puede realizar bien con el sensor de nivelado o bien con finales de carrera en ambos husillos como Z\_MIN en ambos lados.

Esto sirve para configuraciones con dos o tres husillos, en el caso de tres husillos se puede realizar el ajuste automático de la cama modificando la altura de esta para que no sea necesario realizar correcciones en Z durante la impresión como se haría habitualmente con la función “auto-level”.

Parto de la base de que dual (o triple) Z ya se encuentran activados y configurados según se menciono en otro apartado anteriormente.

### 1 Opción de sensor de nivelado (inductivo/Bltouch etc) para el alineamiento:

A Activar la opción de auto-align

```
#define Z_STEPPER_AUTO_ALIGN
```

B Definir los puntos de test:

```
// Define probe X and Y positions for Z1, Z2 [, Z3]  
#define Z_STEPPER_ALIGN_XY { { 45, 150 }, { 290, 150 } }
```

Donde se definiran los puntos a realizar la comprobación, en el caso de dos husillos entiendo que lo conveniente seria realizar dicha comprobación sobre la mitad de la cama en el eje Y y cerca de ambos extremos de la cama en el eje X dejando un margen de seguridad de unos 10 mm.

Para definir las posiciones es conveniente tener en consideración las posiciones limite donde nuestro sensor puede alcanzar, en función del recorrido max-min posible de cada eje y del offset del sensor respecto a la boquilla.

En el caso del ejemplo defino una cama de 300x300, el offset del sensor respecto a la boquilla es de 45 mm a la derecha de la misma y el min alcanzable en X para la boquilla es de 0 mm por lo que:

POSICION 1		POSICION2
{X,Y}		{X,Y}
{{45,150}}	,	{290,150}}

En el caso de disponer de triple Z habría que definir puntos cercanos a donde se encuentra cada uno de los husillos y añadir el tercer punto.

Nota: es importante que la posición 1 se corresponda con el punto cercano a Z y la posición 2 se corresponda con donde se encuentra Z2, de lo contrario la máquina hará lo opuesto a lo esperado y no será capaz de realizar el alineamiento, lo mismo aplica en el caso de disponer de tres husillos.

C- Por último y por preferencias personales yo modifíco las líneas siguientes:

```
#define Z_STEPPER_ALIGN_ITERATIONS 15  
#define RESTORE_LEVELING_AFTER_G34  
#define Z_STEPPER_ALIGN_ACC 0.05
```

## 2 Opción de finales de carrera para los husillos.

A- En dicho caso y como es evidente el sensor de nivelado no será el mismo que el final de carrera que se usa para los husillos por lo que habrá que deshabilitar.

```
#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
```

B- Será necesario definir donde ira conectado el sensor de nivelado, por ejemplo:

```
#define Z_MIN_PROBE_PIN _YMAX_
```

C- Habilitar los endstops a usar, en este caso 3, Y\_MAX para el sensor de nivelado (si corresponde) y dos para cada uno de los husillos (Z\_MIN y Z\_MAX):

```
#define USE_XMIN_PLUG
#define USE_YMIN_PLUG
#define USE_ZMIN_PLUG
//#define USE_XMAX_PLUG
#define USE_YMAX_PLUG
#define USE_ZMAX_PLUG
```

D- Se asumirá con esto que el primer final de carrera para el Z será en Z\_MIN y quedará por definir el final de carrera para el segundo Z que quedará como sigue:

```
#define Z_DUAL_STEPPER_DRIVERS
#if ENABLED(Z_DUAL_STEPPER_DRIVERS)
  #define Z_DUAL_ENDSTOPS
  #if ENABLED(Z_DUAL_ENDSTOPS)
    #define Z2_USE_ENDSTOP _ZMAX_
```

E- Por último definir si hay algún offset del ajuste de los finales de carrera, que o bien se realizará mecánicamente o bien por software en:

```
#define Z_DUAL_ENDSTOPS_ADJUSTMENT
```

## Añadir sensor de filamento:

A- Habilitar sensor de filamento en config:

```
#define FILAMENT_RUNOUT_SENSOR
```

B- Cambiar la lógica del sensor si aplica (la mayoría de los sensores hay que dejarlo en true, también se puede modificar desde el LCD):

```
#define FIL_RUNOUT_INVERTING true // Set to true to invert the logic of the sensor.
```

C- Opcional, usar el pin por defecto o definir el pin donde se va a conectar, esto viene normalmente definido por defecto en el archive pins de la placa.

Para SKR 1.3 la ruta del archive pins es: \Marlin\src\pins\lpc1768

Para el resto de placas la ruta es: \Marlin\src\pins\stm32

Pins por defecto:

- SKR 1.3: #define FIL\_RUNOUT\_PIN P1\_28 // (X\_MAX)
- SKR E3 DIP: #define FIL\_RUNOUT\_PIN PC2
- SKR MINI E3: #define FIL\_RUNOUT\_PIN PC15
- SKR Mini E3 V1.1 y V1.2: Sin definir
- SKR Pro 1.1: sin definir.
- SKR V1.4: P1\_25 o P1\_26

En el caso de placas que venga sin definir se puede añadir la siguiente línea bien en config como en el archive pins:

- SKR Mini E3 V1.0 & V1.2: `#define FIL_RUNOUT_PIN PC15  
//(E0_STOP)`
- SKR Mini V1.1: `#define FIL_RUNOUT_PIN PC13 //(Z_MAX_PIN)`
- SKR Pro 1.1: `#define FIL_RUNOUT_PIN PE15 //(X_MAX/E0)`

\*Nota: el sensor de filamento se puede activar o desactivar desde el LCD, desde pronterface “M412 S0” desactivar detector; “M412 S1” activar detector; M412 reportar estado actual.

\*Nota2: Se puede monitorizar el estado del sensor de filamento con M119 (triggered si está detectando filamento y open si no tiene introducido filamento).

\*Nota3: En el caso de que marque a la inversa se puede invertir la lógica desde el LCD o bien desde Marlin cambiando true/false en:

```
#define FIL_RUNOUT_INVERTING
```

## Cambio de extrusor estándar tipo MK8 o similar a BMG o Titan:

A parte de los cambios de hardware específicos de cada impresora hay dos cosas a modificar, invertir el sentido de giro del stepper del extrusor y el cambio de los pasos por mm.

Ambos pasos se pueden realizar haciendo las consiguientes modificaciones en Marlin o bien sin llegar a tocar Marlin ni tener que flashear nuevamente, vamos a ver ambos modos:

### Modificando Marlin:

- Invertir el sentido de giro del stepper del extrusor (intercambiar true o false según corresponda):

```
#define INVERT_E0_DIR
```

- Cambio de los pasos del extrusor.

Suponiendo que se parte de una configuración de 16 microsteps y stepper de 1,8º en dicho eje habrá que cambiar el valor por 415

para un BMG y de 418.5 para un Titan, en el caso de estar configurado a 32 microsteps habrá que doblar la cantidad de pasos por mm y así sucesivamente.

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 100, 100, 1600, 415 }
```

### Sin Modificar Marlin:

- **Invetir giro del stepper:**

En conectores tipo duppont se podra rotar 180º el conector del extrusor en la placa con lo cual quedará directamente invertido el sentido de giro, en conectores tipo XH no es posible girar el conector por lo que habrá que cortar los cables y empalmar nuevamente habiendo invertido los 4 cables de orden respecto a lo original.





- **Cambio de pasos del extrusor:**

Esta operación se puede realizar a través del LCD en el menú de configuración avanzada, steps por mm, extrusor. Hay que recordar posteriormente guardar los cambios en la Eprom desde el LCD para que se conserve el cambio tras un reinicio.

*Referirse a lo mencionado anteriormente para determinar el número de pasos a ajustar.*

Otra opción sería cambiarlo desde pronterface con el comando:

“M92 Ex” donde x será el número de pasos a ajustar. Seguido de guardar en memoria “M500” (save eeprom).

Se puede comprobar que dicho cambio se ha realizado con éxito comprobando lo que devuelve el comando M501 seguido de M503 esto recargará los ajustes de la Eprom y nos reportará los ajustes actuales.

```
Recv: echo: M92 X100.00 Y100.00 Z1600.00 E415.00
```

**Nota:** tras este proceso sería conveniente realizar un ajuste fino de los pasos del extrusor tal y como se explica al principio de este documento en las recomendaciones tras flashear un nuevo firmware.

## **Ajuste de nivelado automático (bed level bilinear y/o UBL)**

El bilinear lo considero obsoleto pudiendo disponer de UBL, este tiene muchas más opciones de ajuste, en una placa con suficiente memoria como es el caso de las actuales, al no haber problemas de espacio, para mi deja de tener sentido decantarse por bilinear frente a UBL, así que lo siguiente aplica solo para UBL.

Activación de los ajustes siguientes en Marlin:

```
#define AUTO_BED_LEVELING_UBL
```

```
#define DEBUG_LEVELING_FEATURE
```

Opcionalmente yo prefiero activar la siguiente opción de cara a facilitar un primer nivelado manual de ser necesario:

```
#define LEVEL_BED_CORNERS
```

Otras opciones interesantes a considerar:

```
#define ENABLE_LEVELING_FADE_HEIGHT
```

Esto nos permitirá definir una altura vía LCD o pronterface (normalmente sobre los 10 mm), con dicha opción se realizará una corrección de nivelado hasta la altura deseada “difuminando” el desnivel progresivamente para finalmente llegar a la cota deseada a un nivel 0, en el que ya no será necesario seguir realizando auto corrección y reduciendo por tanto la carga sobre los steppers en Z.

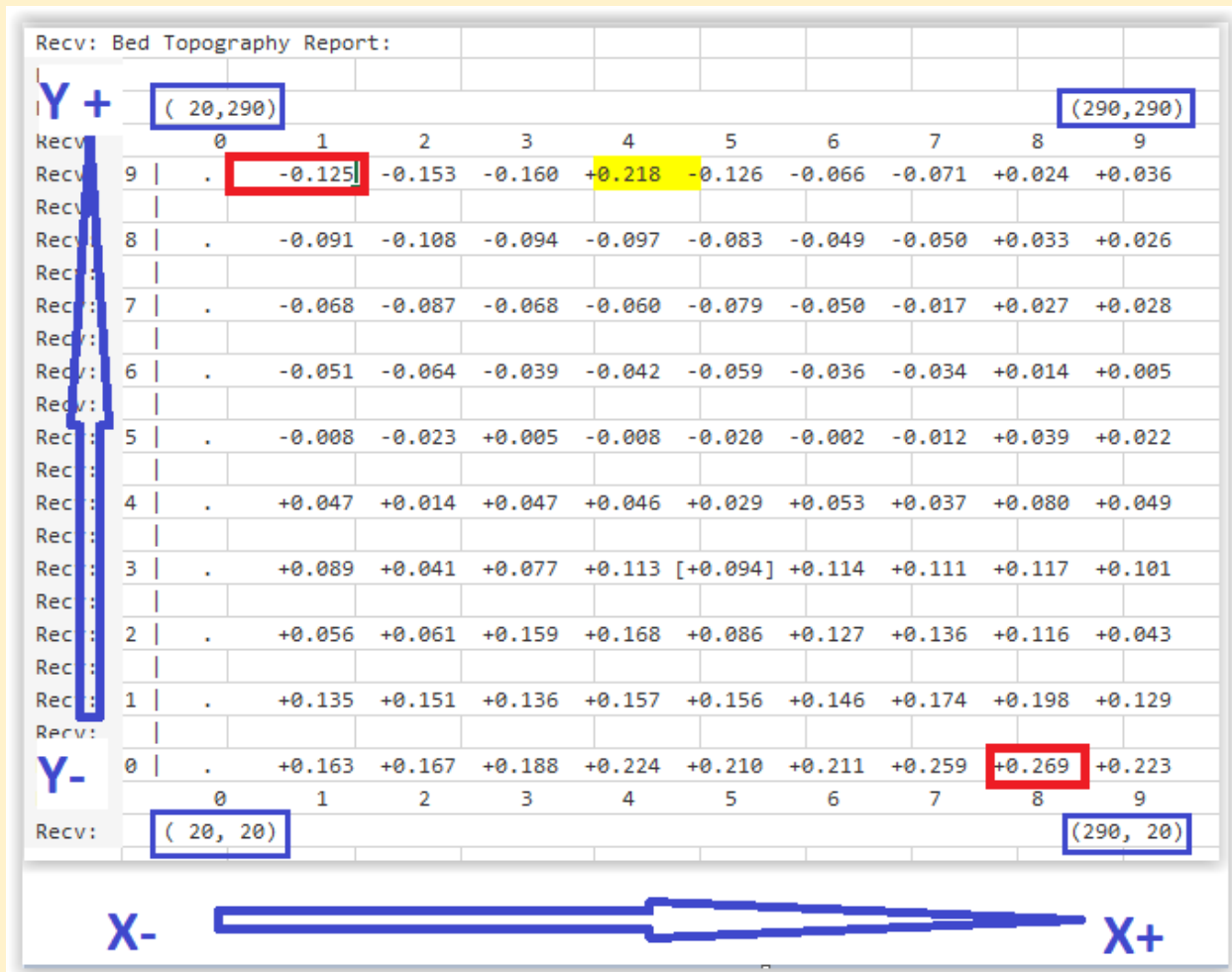
```
#define G26_MESH_VALIDATION
```

Todo y haberlo configurado, nunca me ha acabado de gustar los resultados y prefiero otras plantillas de comprobación, hay multitud de ellas en thingiverse y se pueden escalar en % para que encaje con el tamaño de nuestra cama.

## PASOS A SEGUIR PARA REALIZAR UN NIVELADO DESDE PRONTERFACE O LCD:

- 1- Ajuste manual del nivelado de la cama con la opción de “bed level corners” desde el LCD o bien cualquier otro método de nivelado manual.
- 2- “G28” (home todos los ejes)
- 3- Comprobar que no hay obstrucciones en la zona de test del sensor, que cualquier superficie de impresión (cristal, fleje ...) sen encuentre bien emplazado y limpio.
- 4- Calentar cama y boquilla a temperaturas de trabajo.
- 5- “G29 P1” o bien (build cold mesh desde el LCD).
- 6- “G29 T” Mostramos mesh (no tengo ahora mismo constancia de si se puede actuar desde el LCD y tampoco tendría sentido ya que no podríamos visualizar el resultado).

Que esperamos como respuesta tras este comando? Algo similar a lo siguiente donde se muestran las coordenadas de nuestra cama, y el valor medido en cada uno de los puntos de la rejilla definida, en función del XY\_MAX, XY\_MIN, Z probe offsets y Edge valores que delimitarán los puntos que son “alcanzables” para medida.



En el ejemplo he marcado en amarillo uno de los puntos de comprobación que parece ser una inconsistencia en la lectura respecto a los puntos a su alrededor, en este caso estoy casi seguro que ha tomado la cota sobre la pinza que sujeta el cristal y ese debe ser el motivo, al ser en un extremo no me preocupa ya que evitaremos esa zona de impresión.

He marcado en rojo los valores max y min . En el ejemplo la diferencia es de  $+0,269 / -0,125 = \underline{0,394 \text{ mm}}$  lo que encuentro excesivo y creo

conveniente corregir a ser posible de cara a trabajar a largo plazo hasta unos valores en inferiores a 0,15 aprox.

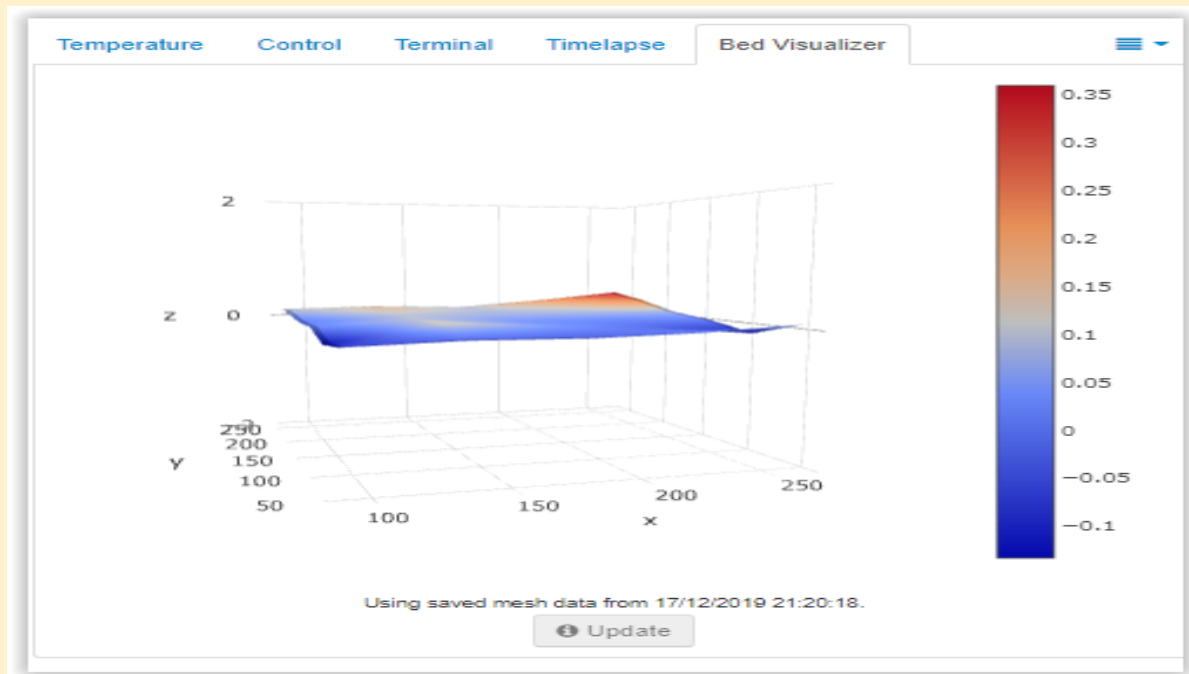
Marcado en azul los puntos de referencia y los ejes.

Una vez realizada una corrección volver al paso 5, volver a comprobar los valores al llegar al paso 6 y pasar al paso 7 una vez conforme con los valores deseados.

Para el que tenga Octoprint:

<https://plugins.octoprint.org/plugins/bedlevelvisualizer/>

Octoprint bed visualizer plugin return ejemplo:



- 7- "G29 P3" (Fill unpopulated) repetir esta operación dos veces para completar auto completado de los puntos no alcanzables del mesh.
- 8- "G29 S1" Guardar mesh en slot 1 de la memoria.
- 9- "G29 F10.0" Seleccionar fade a 10 mm desde el LCD (opcional).
- 10- "G29 A" o Activar auto-nivelado desde le LCD.
- 11- "M500" , Guardar ajustes (yo repito esta operación dos veces)

Tras esto cada vez que reiniciemos la máquina el auto-nivelado debería de quedar activado por defecto con el mesh guardado.

## 12- Ajuste de Z offset.

Nota: Se puede automatizar el proceso modificando en Marlin la línea Z\_PROBE\_END\_SCRIPT y dejándola como sigue:

```
#define Z_PROBE_END_SCRIPT "G29 P3 \n G29 P3 \n G29 S1 \n G29 F10.0 \n G29 A \n M500 \n G29 T \n M300 S440 P200 \n M300 S660 P250 \n M300 S880 P300"
```

## Ajuste del Z offset

Recomendaría realizar esta operación o la re-comprobación tras el auto-nivelado.



- 1- "G28" home all axis.
- 2- "M851 Z0" de creerse conveniente, esto ajustara el Z offset al 0.
- 3- "M211 S0" o deshabilitar end stops desde el LCD.
- 4- "G1 X150 Y150 Z0 F7000" "go to center desde el custom menú del LCD (ejemplo para una cama de 300x300 se trata de enviar la boquilla al centro de la cama y Z=0.
- 5- Desde el LCD o pronterface ir moviendo manualmente reduciendo progresivamente la posición de Z hasta encontrar el punto en que se engancha el papel entre la boquilla y la cama.
- 6- Una vez en la posición deseada ver cuanto hemos desplazarlo y ajustar dicho valor como Z offset con "M851 Z-x" donde X es el valor desplazado en negativo anteriormente. Esta operación se puede realizar desde el LCD en ajustar Z offset.
- 7- "G28" home all axis.
- 8- "G1 X150 Y150 Z0 F7000" "go to center desde el custom menú del LCD (ejemplo para una cama de 300x300 se trata de enviar la boquilla al centro de la cama y Z=0.

- 9- Comprobar si sigue quedando en la posición deseada y realizar alguna corrección si es preciso volviendo al punto 5, si no es preciso, pasar al punto 10.
- 10- “M500” o guardar memoria desde el LCD (nota, yo repito esta operación por seguridad).
- 11- Prueba de impresión, al inicio de una impresión de prueba y con las opciones que hay pre-configuradas en los Marlin de los ejemplos se puede realizar una corrección del Z offset “on the fly” al pulsar dos veces sobre la rueda del LCD nos permitirá mediante el giro de la rueda desplazar el Z offset hasta que consideremos correcto.
- 12- En el caso de haber realizado corrección en el punto anterior es preciso guardar dicho cambio nuevamente en la Eeprom con “M500” o “store settings” desde el LCD.

# EPROM EN PLACAS SIN MODULO INTEGRADO.

## Por ejemplo SKR Pro 1.1

En algunas placas estas no llevan eprom integrada por lo que para poder disponer de esta funcionalidad hay que considerar una de las dos opciones siguientes:

- Emulación Eprom:

La funcionalidad se puede emular de manera que se reserve un espacio en la memoria flash y esta se use como Eprom.

Para esto es necesario acceder al archivo pins correspondiente a la placa con la que estemos trabajando y habilitar la función:

```
#define FLASH_EEPROM_EMULATION // Use Flash-  
based EEPROM emulation
```

Esta es la única opción posible en las placas mini con base STM32 sin módulo Eprom integrado.

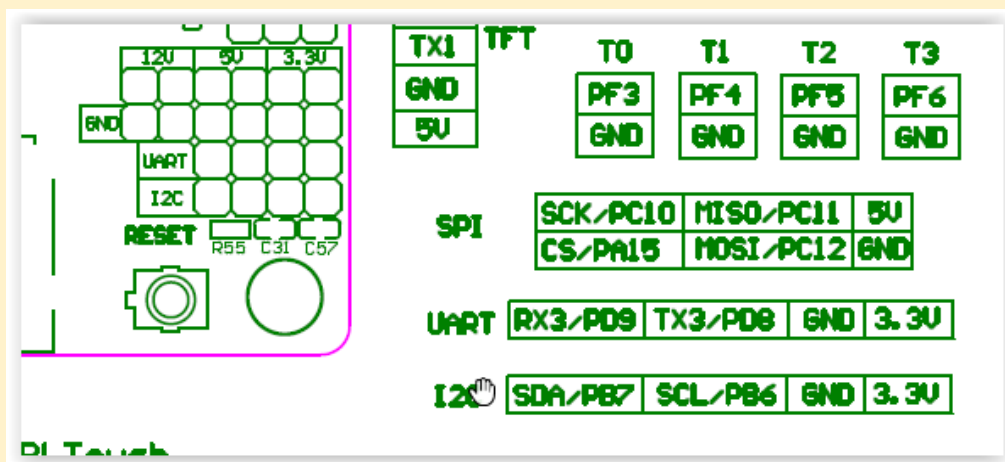
- Añadir un módulo Eprom:

Otra opción es añadir físicamente una eprom y conectarla vía I2C a la placa, a día de hoy esto es factible al venir ya preparadas de origen con el bus en la SKR PRO 1.1 solo.

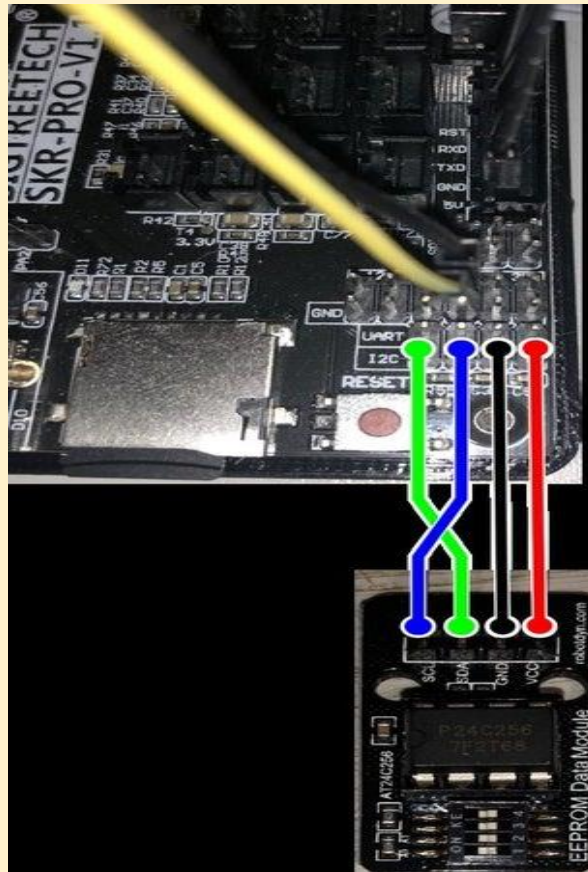
Estos módulos Eprom se pueden adquirir en Aliexpress por cerca de 1€ por ejemplo:

<https://es.aliexpress.com/item/32966161068.html?spm=a2g0s.9042311.0.0.274263c00N1o0m>

Se debería conectar dicho módulo siguiendo el pinout del módulo adquirido y en la placa como sigue:



Hay que fijarse en el sentido de las conexiones SDA/SCL ya que según el módulo que hayamos adquirido es posible que haya que girar el orden de los cables en una de las puntas.



Tras esto habría que añadir las líneas siguientes en el archivo pins correspondiente:

```
#define I2C_EEPROM
#ifdef E2END
#undef E2END
#endif
#define E2END 0x7FFF // EEPROM end address AT24C256 (32kB)
```

En la última línea se define la dirección I2C y el tamaño de la memoria, es posible que haya que modificarlo según el módulo que hayamos adquirido.

Como es lógico habrá que habilitar la función Eeprom en el archivo configuración de Marlin.

# VIDEOS DE SOPORTE

- 1- M500-501-502-503: <https://www.youtube.com/watch?v=R0fKG0ANCuo>
- 2- UBL: <https://www.youtube.com/watch?v=KamOhRwdNPY&t=3s>
- 3- Conectar LCD 2004: <https://photos.app.goo.gl/hb59rXReZYLMMQv8>
- 4- Llebar boquilla al centro para ajuste del Z offset según el menú preconfigurado en custom commands. <https://photos.app.goo.gl/gdibpS66p7EouXaw7>
- 5- Cargar material manualmente según el menú preconfigurado en custom commands. (load, para descargar seria la función unload).  
<https://photos.app.goo.gl/oKpdhHH7Tw7Prwg96>
- 6- G34, auto-alineamiento Z. <https://photos.app.goo.gl/cvsJoZHxDc2EKEGYA>