

# G0, G1 - Linear Move

[↩ 1.0.0-beta](#) [🔗 planner](#) [➡ Add a straight line movement to the planner](#)

The `G0` and `G1` commands add a linear move to the queue to be performed after all previous moves are completed. These commands yield control back to the command parser as soon as the move is queued, but they may delay the command parser while awaiting a slot in the queue.

A linear move traces a straight line from one point to another, ensuring that the specified axes will arrive simultaneously at the given coordinates (by linear interpolation). The speed may change over time following an acceleration curve, according to the acceleration and jerk settings of the given axes.

A command like `G1 F1000` sets the feedrate for all subsequent moves.

Marlin treats `G0` (rapid linear movement) as an alias to `G1` (rapid movement).

By convention, most G-code generators use `G0` for non-extrusion movements (those without the E axis) and `G1` for moves that include extrusion. This is meant to allow a kinematic system to, optionally, do a more rapid uninterpolated movement requiring much less calculation.

## Usage

`G0 [E<pos>] [F<rate>] [X<pos>] [Y<pos>] [Z<pos>]`

Argument	Description
[E<pos>]	The length of filament to feed into the extruder between the start and end point
[F<rate>]	The maximum movement rate of the move between the start and end point. The feedrate set here applies to subsequent moves that omit this parameter.
[X<pos>]	A coordinate on the X axis
[Y<pos>]	A coordinate on the Y axis
[Z<pos>]	A coordinate on the Z axis

## Notes

- Coordinates are given in millimeters by default. Units may be set to inches by `G20`.
- In Relative Mode (`G91`) all coordinates are interpreted as relative, adding onto the previous position.
- A single linear move may generate several smaller moves due to kinematics and bed leveling compensation.
- Developers, keep using `G0` for non-print moves. It makes G-code more adaptable to lasers, engravers, etc.
- In a future version of Marlin, `G0` will do rapid movement, optionally, on SCARA machines.

## Examples

The most basic move sets a feedrate and moves the tool to the given position.

```
G0 X12 ; move to 12mm on the X axis
G0 F1500 ; set the feedrate to 1500mm/minute
G1 X90.6 Y13.8 ; move to 90.6mm on the X axis and 13.8mm on the Y axis
```

There are some caveats related with feedrates. Consider the following:

```
G1 F1500 ; set the feedrate to 1500mm/minute
G92 E0
G1 X50 Y25.3 E22.4 ; move while extruding
```

In the above example the feedrate is set to 1500mm/minute, then the tool is moved 50mm on the X axis and 25.3mm on the Y axis while extruding 22.4mm of filament between the two points.

```
G1 F1500
G92 E0
G1 X50 Y25.3 E22.4 F3000
```

However, in the above example, we set a feedrate of 1500 mm/minute on line 1 then do the move described above, accelerating to a feedrate of 3000 mm/minute (if possible). The extrusion will accelerate along with the X and Y movement, so everything stays synchronized.

---

# G2, G3 - Controlled Arc Move

[↔ 1.0.0-beta](#) [🔗 planner](#) [📌 Add an arc movement to the planner](#)

G2 adds a clockwise arc move to the planner; G3 adds a counter-clockwise arc. An arc move starts at the current position and ends at the given XYZ, pivoting around a center-point offset given by I and J or R.

This command has two forms:

## I J Form

- I specifies an X offset. J specifies a Y offset.
- At least one of the I J parameters is required.
- X and Y can be omitted to do a complete circle.
- The given X Y is not error-checked. The arc ends based on the angle of the destination.
- Mixing I or J with R will throw an error.

## R Form

- R specifies the radius. X or Y is required.
- Omitting both X and Y will throw an error.
- X or Y must differ from the current XY position.
- Mixing R with I or J will throw an error.

Arc moves actually generate several short straight-line moves, the length of which are determined by the configuration option MM\_PER\_ARC\_SEGMENT (default 1mm). Any change in the Z position is linearly interpolated over the whole arc.

---

## Usage

G2 [E<pos>] [F<rate>] I<offset> J<offset> R<radius> [X<pos>] [Y<pos>] [Z<pos>]

Argument	Description
[E<pos>]	The amount to extrude between the start point and end point
[F<rate>]	The maximum rate of the move between the start and end point
I<offset>	An offset from the current X position to use as the arc center
J<offset>	An offset from the current Y position to use as the arc center
R<radius>	A radius from the current XY position to use as the arc center
[X<pos>]	A coordinate on the X axis
[Y<pos>]	A coordinate on the Y axis
[Z<pos>]	A coordinate on the Z axis

## Examples

Move in a clockwise arc from the current position to (125, 32) with the center offset from the current position by (10.5, 10.5).

```
G2 X125 Y32 I10.5 J10.5
```

Move in a counter-clockwise arc from the current position to (125, 32) with the center offset from the current position by (10.5, 10.5).

```
G3 X125 Y32 I10.5 J10.5
```

Move in a complete clockwise circle with the center offset from the current position by 20, 20.

```
G2 I20 J20
```

---

## G4 - Dwell

[↔ 1.0.0-beta](#) [🔗 planner](#) [📌 Pause the planner](#)

Dwell pauses the command queue and waits for a period of time.

## Usage

G4 [P<time in ms>] [S<time in sec>]

Argument	Description
[P<time in ms>]	Amount of time to dwell
[S<time in sec>]	Amount of time to dwell

## Notes

If both S and P are included, S takes precedence.




M0 / M1 provides an interruptible "dwell" (Marlin 1.1.0 and up).

G4 with no arguments is effectively the same as M400.

## Example

```
G4 P500 ; Dwell for 1/2 second
```

## G5 - Bézier cubic spline

  1.1.0   Cubic B-spline with XYZ destination and IJPQ offsets

G5 creates a cubic B-spline in the XY plane with the X and Y axes only. P and Q parameters are required. I and J are required for the first G5 command in a series. For subsequent G5 commands, either both I and J must be specified, or neither. If I and J are unspecified, the starting direction of the cubic will automatically match the ending direction of the previous cubic (as if I and J are the negation of the previous P and Q).

See This interactive demo (<https://www.geogebra.org/m/WPHQ9rUt>) to understand how Bézier control points work.

## Usage

G5 [E<pos>] I<pos> J<pos> P<pos> Q<pos> X<pos> Y<pos>

Argument	Description
[E<pos>]	The length of filament to feed into the extruder between the start and end point
I<pos>	X incremental offset from start point to first control point
J<pos>	Y incremental offset from start point to first control point
P<pos>	X incremental offset from end point to second control point
Q<pos>	Y incremental offset from end point to second control point
X<pos>	A coordinate on the X axis
Y<pos>	A coordinate on the Y axis

## Notes

It is an error if an axis other than X or Y is specified.

# Examples

For example, to program a curvy “N” shape:

```
G0 X0 Y0  
G5 I0 J3 P0 Q-3 X1 Y1
```

A second curvy “N” that attaches smoothly to this one can now be made without specifying **I** and **J** :

```
G5 P0 Q-3 X2 Y2
```

## G10 - Retract

[↔ 1.0.0-beta](#) [📄 planner](#) [📄 Retract the filament](#) [FWRETRACT](#)

Retract the filament according to settings of `M207` (</docs/gcode/M207.html>).

Firmware retraction allows you to tune retraction at the machine level and can significantly reduce the size of G-code files.

## Usage

G10 [`S<flag>`]

Argument	Description
[ <code>S&lt;flag&gt;</code> ]	Use <code>G10 S1</code> to do a swap retraction, before changing extruders. The subsequent <code>G11</code> (after tool change) will do a swap recover. (Requires <code>EXTRUDERS &gt; 1</code> )

## Notes

Requires `FWRETRACT` .

See related codes `G11` (</docs/gcode/G011.html>), `M207` (</docs/gcode/M207.html>), `M208` (</docs/gcode/M208.html>), and `M209` (</docs/gcode/M209.html>).

## Example

```
G10 ; retract
```

## G11 - Recover

[↔ 1.0.0-beta](#) [📄 planner](#) [📄 Recover the filament with firmware-based retract.](#) [FWRETRACT](#)

Unretract (i.e., recover, prime) the filament according to settings of `M208` (</docs/gcode/M208.html>).

## Usage

G11

## Notes



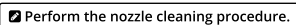
Requires `FWRETRACT` .

See related codes `G10` (</docs/gcode/G010.html>), `M207` (</docs/gcode/M207.html>), `M208` (</docs/gcode/M208.html>), and `M209` (</docs/gcode/M209.html>).

## Example

```
G11 ; recover
```

# G12 - Clean the Nozzle

   **NOZZLE\_CLEAN\_FEATURE**

{“Start the nozzle cleaning process. Two types of cleaning patterns are supported”=>“stroke-based and zigzag-based. This feature requires a dedicated cleaning area on or outside the bed, but within reach of the nozzle.”}

## Usage

G12 [P<0|1>] [S<count>] [T<count>]

Argument	Description
[P<0 1>]	Pattern style selection
[S<count>]	Number of strokes (i.e. back-and-forth movements)
[T<count>]	Number of repetitions

## Notes

Default behavior is defined by NOZZLE\_CLEAN\_STROKES , NOZZLE\_CLEAN\_START\_POINT , NOZZLE\_CLEAN\_END\_POINT and NOZZLE\_CLEAN\_PARK .

With NOZZLE\_CLEAN\_PARK enabled, the nozzle automatically returns to the XYZ position before G12 .

## Examples



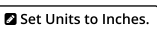
The most basic example is to use the command without any arguments, this will default to a stroke based pattern which will be stroked NOZZLE\_CLEAN\_STROKES times.

```
G12 ; stroke pattern (default)
```

To generate a three triangle zig-zag pattern which will be stroked one time use the following command.

```
G12 P1 S1 T3 ; zig-zag pattern with 3 triangles
```

# G20 - Inch Units

   **INCH\_MODE\_SUPPORT**

Set units to inches. In this mode, all positions, offsets, rates, accelerations, etc., specified in G-code parameters are interpreted as inches.



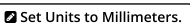
## Usage

G20

## Example

```
G20 ; set units to inches
```

# G21 - Millimeter Units

   **INCH\_MODE\_SUPPORT**

Set units to millimeters. In this mode, all positions, offsets, rates, accelerations, etc., specified in GCode parameters are interpreted as millimeters.







## Usage

G21

## Example

```
G21 ; set units to millimeters
```

## G26 - Mesh Validation Pattern

  1.1.0  calibration  Test the mesh and adjust.  HAS\_MESH  G26\_MESH\_VALIDATION

G26 Mesh Validation Pattern is designed to be used in conjunction with mesh-based leveling to test the accuracy of the probed mesh.

The G26 command prints a single-layer pattern over the entire print bed, giving a clear indication of how accurately every mesh point is defined. G26 can be used to determine which areas of the mesh are less-than-perfect and how much to adjust each mesh point.

## Usage




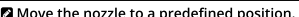

```
G26 [B<temp>] [C<bool>] [D<bool>] [F<linear>] [H<linear>] [K<bool>] [L<linear>] [O<linear>] [P<linear>] [Q<float>] [R<int>] [S<float>] [U<linear>] [X<linear>] [Y<linear>]
```

Argument	Description
[B<temp>]	Bed temperature (otherwise 60°C) to use for the test print.
[C<bool>]	Continue with the closest point (otherwise, don't)
[D<bool>]	Disable leveling compensation (otherwise, enable)
[F<linear>]	Filament diameter (otherwise 1.75mm)
[H<linear>]	Hot end temperature (otherwise 205°C) to use for the test print.
[K<bool>]	Keep heaters on when done
[L<linear>]	Layer height to use for the test
[O<linear>]	Ooze amount (otherwise 0.3mm). Emitted at the start of the test.
[P<linear>]	Prime Length
[Q<float>]	Retraction multiplier (otherwise 1.0)
[R<int>]	Number of G26 Repetitions (otherwise 999)
[S<float>]	Nozzle size (otherwise 0.4mm)
[U<linear>]	Random deviation. ( U with no value, 50).
[X<linear>]	X position (otherwise, current X position)
[Y<linear>]	Y position (otherwise, current Y position)

## Notes

- Before 1.1.6 requires UBL\_G26\_MESH\_VALIDATION and AUTO\_BED\_LEVELING\_UBL .
- Since 1.1.7 requires G26\_MESH\_VALIDATION and HAS\_MESH ( AUTO\_BED\_LEVELING\_UBL , MESH\_BED\_LEVELING , or AUTO\_BED\_LEVELING\_BILINEAR ).

## G27 - Park the nozzle

  1.1.0  nozzle  Move the nozzle to a predefined position.  NOZZLE\_PARK\_FEATURE

Park the nozzle at a predefined XYZ position.

# Usage

G27 [P<0|1|2>]

Argument	Description
[P<0 1 2>]	Z axis action <ul style="list-style-type: none"><li>P0 : If current Z-pos is lower than Z-park then the nozzle will be raised to reach Z-park height</li><li>P1 : No matter the current Z-pos, the nozzle will be raised/lowered to reach Z-park height</li><li>P2 : The nozzle height will be raised by Z-park amount but never going over the machine's limit of Z_MAX_POS</li></ul>

## Notes

Requires NOZZLE\_PARK\_FEATURE .

The park position is defined by NOZZLE\_PARK\_POINT .

## Examples

The most basic example is to use the command without any arguments, this will default to a move the the parking position and raising the Z-pos if lower than the default Z-park position.

```
G27 ; raise Z if lower
```

This one is useful to be used on the end-script of a print, it will raise the Z-pos by Z-park.

```
G27 P2 ; always raise Z
```

## G28 - Auto Home

[↔ 1.0.0-beta](#) [🔧 calibration](#) [📄 Auto home one or more axes.](#)

Auto-home one or more axes, moving them towards their endstops until triggered. Each axis is backed off and re-bumped according to the [XYZ]\_HOME\_BUMP\_MM and HOMING\_BUMP\_DIVISOR settings.

## Usage

G28 [X] [Y] [Z]

Argument	Description
[X]	Flag to go back to the X axis origin
[Y]	Flag to go back to the Y axis origin
[Z]	A coordinate on the Z axis

## Notes

Homing is required before G29 , M48 , and some other procedures.

If homing is needed the LCD will blink the X Y Z indicators.

## Examples

The most-used form of this command is to home all axes:

```
G28 ; Go to origin on all axes
```

With no arguments to G28 , Marlin homes according to the Z\_SAFE\_HOMING , QUICK\_HOME and HOME\_Y\_BEFORE\_X settings.

# G29 - Automatic Bed Leveling

[<> 1.0.0-beta](#)
[🔧 calibration](#)
[🔍 Probe the bed and enable leveling compensation.](#)
[AUTO\\_BED\\_LEVELING\\_\(3POINT|LINEAR|BILINEAR\)](#)

ABL probes the bed at 3 or more points and enables bed leveling compensation. The exact procedure and method depends on the type of bed leveling chosen in `Configuration.h`:

- `AUTO_BED_LEVELING_3POINT` probes 3 points and uses a matrix to compensate for bed tilt.
- `AUTO_BED_LEVELING_LINEAR` also uses a tilt matrix but probes a grid and applies “least-squares.”
- `AUTO_BED_LEVELING_BILINEAR` probes a grid and produces a mesh to adjust Z across the bed.

The printer must be homed with `G28` before `G29`.

- For `AUTO_BED_LEVELING_UBL` see [G29 UBL \(/docs/gcode/G029-ubl.html\)](/docs/gcode/G029-ubl.html) and [G26 Mesh Editing \(/docs/gcode/G026.html\)](/docs/gcode/G026.html).
- For `MESH_BED_LEVELING` see the [G29 MBL \(/docs/gcode/G029-mbl.html\)](/docs/gcode/G029-mbl.html) page.

## Automatic Probing

With an electronic probe Marlin can probe the entire bed in a matter of seconds—even before every print—with a single `G29` command. Available parameters depend on the style of leveling. See parameter descriptions and examples below for details.

## Manual Probing

Auto Bed Leveling now includes a `PROBE_MANUALLY` option for systems lacking a probe.

**`PROBE_MANUALLY`** adds these parameters to **`G29`**:

- `Q` : Query leveling and `G29` state
- `A` : Abort current leveling procedure

To do manual probing simply repeat `G29` until the procedure is complete.

The first `G29` accepts the same parameters, shown in the Usage section below. The exact parameters available will depend on which style of bed leveling is enabled. (**Note:** *UBL parameters are not covered on this page unless they coincide. See the `G29` for UBL (/docs/gcode/G029-ubl.html) page for a full list of its options.*)

**Before probing the bed:**

1. Use `G29 Q` to get the current status. If `G29` isn't idle, abort with `G29 A`.
2. Use `M420 V` to view leveling data. You can send `M420 S1` to use the existing data.

**To probe the bed using GCode:**

1. Use `G29` to move to the first point for Z adjustment.
2. Adjust Z so a piece of paper can just pass under the nozzle.
3. Use `G29` to save the Z value and move to the next point.
4. Repeat steps 3-4 until completed.
5. Use `M500` to save the leveling data to EEPROM, if desired.

**To probe the bed using your LCD controller:** (Requires `LCD_BED_LEVELING`)

1. Select the `Level Bed` sub-menu, then choose `Level Bed (not Cancel)`.
2. Wait for `Homing XYZ` to complete.
3. When `Click to Begin` appears, press the controller button to move to the first point.
4. Use the controller wheel to adjust Z so that a piece of paper can just pass under the nozzle.
5. Press the controller button to save the Z value and move to the next point.
6. Repeat steps 4-5 until completed.
7. Use `Control > Store memory` to save the mesh to EEPROM, if desired.

## Mesh Editing

`AUTO_BED_LEVELING_BILINEAR` adds these parameters to `G29` for editing mesh points:

- `W` Write a mesh point. (Ignored during leveling.)
- `I` X index for mesh point
- `J` Y index for mesh point
- `X` logical X of mesh point
- `Y` logical Y of mesh point
- `Z` Z to store in mesh. If omitted, current raw Z.

## Usage



G29 [A<bool>] [B<linear>] [C<bool>] [D<bool>] [E<bool>] [F<linear>] [J<bool>] [L<linear>] [P<int>] [Q<bool>] [R<linear>] [S<rate>] [T<bool>] [V<0|1|2|3|4>] [W<bool>] [X<int/float>] [Y<int/float>] [Z<float>]

Argument	Description
[A<bool>]	Abort leveling procedure in-progress ( PROBE_MANUALLY )
[B<linear>]	Set the back limit of the probing grid ( AUTO_BED_LEVELING_LINEAR , AUTO_BED_LEVELING_BILINEAR )
[C<bool>]	Create a fake grid for testing. ( DEBUG_LEVELING_FEATURE )
[D<bool>]	Dry-Run mode. Just probe the grid but don't update the bed leveling data
[E<bool>]	<ul style="list-style-type: none"> <li>By default G29 will engage the Z probe, test the bed, then disengage.</li> <li>Include "E" to engage/disengage the Z probe for each sample.</li> <li>There's no extra effect if you have a fixed Z probe. (without PROBE_MANUALLY )</li> </ul>
[F<linear>]	Set the front limit of the probing grid ( AUTO_BED_LEVELING_LINEAR , AUTO_BED_LEVELING_BILINEAR )
[J<bool>]	Jettison the leveling data stored in SRAM and turn off leveling compensation. Data in EEPROM is not affected.
[L<linear>]	Set the left limit of the probing grid ( AUTO_BED_LEVELING_LINEAR , AUTO_BED_LEVELING_BILINEAR )
[P<int>]	Set the size of the square grid that will be probed - P x P points ( AUTO_BED_LEVELING_LINEAR )
[Q<bool>]	Query the current leveling state ( PROBE_MANUALLY , DEBUG_LEVELING_FEATURE )
[R<linear>]	Set the right limit of the probing grid ( AUTO_BED_LEVELING_LINEAR , AUTO_BED_LEVELING_BILINEAR )
[S<rate>]	Set the XY travel speed between probe points ( AUTO_BED_LEVELING_LINEAR , AUTO_BED_LEVELING_BILINEAR )
[T<bool>]	Generate a Bed Topology Report ( AUTO_BED_LEVELING_LINEAR )
[V<0 1 2 3 4>]	Set the verbose level <ul style="list-style-type: none"> <li>V2 : Level 2 and up act like 'T'</li> </ul>
[W<bool>]	Write a mesh Z offset ( PROBE_MANUALLY ). W requires X , Y , and Z .
[X<int/float>]	<ul style="list-style-type: none"> <li>Override the X-size of the grid that will be probed ( AUTO_BED_LEVELING_LINEAR ).</li> <li>Specify X when setting a mesh value ( PROBE_MANUALLY ).</li> </ul>
[Y<int/float>]	<ul style="list-style-type: none"> <li>Override the Y-size of the grid that will be probed ( AUTO_BED_LEVELING_LINEAR ).</li> <li>Specify Y when setting a mesh value ( PROBE_MANUALLY ).</li> </ul>
[Z<float>]	Specify the Z offset when setting a mesh value ( PROBE_MANUALLY ).

## Notes

Any arguments left out of G29 will use the default values set in Configuration.h .

## Examples

Automatic Probing examples

```
G29 ; execute ABL
```

G29 is most commonly used without any arguments. This uses the defaults set in Configuration.h .

```
G29 P5 ; 5x5 matrix
```

Probe a 5x5 matrix. ( AUTO\_BED\_LEVELING\_LINEAR )

```
G29 X4 Y8 L50 R150 F50 B150 T V4
```

Probe a 4x8 matrix from X50 Y50 to X150 Y150 , printing a full report. ( AUTO\_BED\_LEVELING\_LINEAR , AUTO\_BED\_LEVELING\_BILINEAR )

### Manual Probing example

```
G29 V1 ; Ready! Go to Point 1, wait...
G29 V1 ; Store Z, go to Point 2, wait...
G29 V1 ; Store Z, go to Point 3, wait...
G29 V1 ; Store Z, go to Point 4, wait...
G29 V1 ; Store Z, go to Point 5, wait...
G29 V1 ; Store Z, go to Point 6, wait...
G29 V1 ; Store Z, go to Point 7, wait...
G29 V1 ; Store Z, go to Point 8, wait...
G29 V1 ; Store Z, go to Point 9, wait...
G29 V1 ; Store Z. Calculate matrix. Activate.
```

Each G29 command goes to the next step until the whole procedure is done. The V1 parameter provides a progress report.

## G29 - Mesh Bed Leveling

↗ 1.0.0-beta

🔧 calibration

📏 Measure Z heights in a grid, enable leveling compensation

MESH\_BED\_LEVELING

With Mesh Bed Leveling (MBL) you can interactively measure a grid of Z heights without a bed probe. The only tool required is a piece of paper or a feeler gauge. MBL uses the mesh to compensate for variations in height across the bed. Comparable to using AUTO\_BED\_LEVELING\_BILINEAR with PROBE\_MANUALLY .

This feature is enabled with the MESH\_BED\_LEVELING option in Configuration.h . Users with a probe should enable one of the AUTO\_BED\_LEVELING\_\* (/docs/gcode/G029-abl.html) options instead.

### To do Mesh Bed Leveling from your host software:

1. Use G29 S0 to get the current status and mesh. If there's an existing mesh, you can send M420 S1 to use it.
2. Use G29 S1 to move to the first point for Z adjustment.
3. Adjust Z so a piece of paper can just pass under the nozzle.
4. Use G29 S2 to save the Z value and move to the next point.
5. Repeat steps 3-4 until completed.
6. Use M500 to save the mesh to EEPROM, if desired.

### To do LCD Bed Leveling with your controller: (Requires LCD\_BED\_LEVELING )

1. Select Level Bed then choose Level Bed (not Cancel) in the sub-menu.
2. Wait for Homing XYZ to complete.
3. When Click to Begin appears, press the controller button to move to the first point.
4. Use the controller wheel to adjust Z so that a piece of paper can just pass under the nozzle.
5. Press the controller button to save the Z value and move to the next point.
6. Repeat steps 4-5 until completed.
7. Use Control > Store memory to save the mesh to EEPROM, if desired.

## Usage

```
G29 S<0|1|2|3|4|5>
```


Argument	Description
S<0 1 2 3 4 5>	<ul style="list-style-type: none"><li>• S0 : Produce a mesh report (see example 1)</li><li>• S1 : Start probing mesh points</li><li>• S2 : Probe the next mesh point</li><li>• S3 : Manually modify a single point</li><li>• S4 : Set Z-Offset, positive away from bed, negative closer to bed.</li><li>• S5 : Reset and disable mesh</li></ul>

## Example

S0 produces a mesh report as follows:

```
+----> X-axis 1-n
|
|
v Y-axis 1-n
```

# G29 - Unified Bed Leveling

 calibration  Probe the bed and enable leveling compensation. `AUTO_BED_LEVELING_UBL`

The Unified Bed Leveling System (UBL) provides a comprehensive set of resources to produce the best bed leveling results possible.

See the full Unified Bed Leveling ([/docs/features/unified\\_bed\\_leveling.html](/docs/features/unified_bed_leveling.html)) documentation for more details. (Examples below.)

## Usage

G29 [A<bool>] [B<bool>] [C<bool/float>] [D<bool>] [E<bool>] [F<linear>] [H<linear>] [I<int>] [J<int>] [K<index>] [L<index>] [P<0|1|2|3|4|5|6>] [Q<index>] [R<int>] [S<slot>] [T<0|1>] [U<bool>] [V<0|1|2|3|4>] [W<bool>] [X<linear>] [Y<linear>]

Argument	Description
[A<bool>]	Activate Unified Bed Leveling (i.e., M420 S1)
[B<bool>]	Business Card mode ( P2 only)
[C<bool/float>]	Continue ( P1 ), Current ( P2 ), Constant ( P3 , Q2 )
[D<bool>]	Disable Unified Bed Leveling (i.e., M420 S0).
[E<bool>]	Stow probe after probing E ach point ( P1 only).
[F<linear>]	Fade height. (UBL only! For others use M420 Z)
[H<linear>]	Height for Manual Probe raise ( P2 only).
[I<int>]	Invalidate this number of mesh points. (No value = 1)
[J<int>]	<ul style="list-style-type: none"><li>With a value (v), do <i>Square Grid</i> probing of v x v points.</li><li>With no value, do <i>Three Point</i> probing - e.g. to adjust a loaded mesh to match slight bed misalignment.</li></ul>
[K<index>]	<b>Kompare:</b> Subtract the stored mesh with the given index from the current mesh. This operates on the mesh in-memory, so it will probably invalidate the active mesh for purposes of printing.
[L<index>]	Load a mesh. If no index is given, load the previously-activated mesh.
[P<0 1 2 3 4 5 6>]	Phase <ul style="list-style-type: none"><li>P0 : <b>Zero Mesh Data</b> and turn off the Mesh Compensation System.</li><li>P1 : Invalidate Mesh and do <b>Automatic Z Probing</b>.</li><li>P2 : <b>Probe Areas</b> of the Mesh that can't be automatically handled.</li><li>P3 : <b>Fill Unpopulated</b> regions of the Mesh with a fixed value ( C ) or use 'smart fill' to extrapolate from already probed points ( no argument ).</li><li>P4 : <b>Fine Tune</b> the Mesh. Generally used in the form G29 P4 Rnn Xxxx Yyyy .</li><li>P5 : <b>Find Mean</b> Mesh Height and Standard Deviation.</li><li>P6 : <b>Shift Mesh</b> height by the C value.</li></ul>
[Q<index>]	Test Pattern
[R<int>]	Repeat count. (Default GRID_MAX_POINTS_X * GRID_MAX_POINTS_Y)
[S<slot>]	Store mesh to EEPROM in the given slot. If no slot given, use last-activated. Use S-1 for GCode output.
[T<0 1>]	<b>Topology:</b> Include a Topology Map in the output. Can be used alone or with several other commands. A map type can also be specified: <ul style="list-style-type: none"><li>T0 : Human-readable</li><li>T1 : Spreadsheet-readable</li></ul>
[U<bool>]	<b>Unlevel:</b> Probe the outer perimeter to assist physical leveling. (Use with G29 P1 0)
[V<0 1 2 3 4>]	Verbosity Level (0-4)

Argument	Description
[W<bool>]	<i>What?</i> : Displays current Unified Bed Leveling info
[X<linear>]	X position (otherwise, current X position)
[Y<linear>]	Y position (otherwise, current Y position)

## Notes

Requires AUTO\_BED\_LEVELING\_UBL .

## Examples

This is a minimal 'quick-start' sequence for set-up and initial probing of a UBL mesh on a machine that includes a display and z-probe

```
M502      ; Reset settings to configuration defaults...
M500      ; ...and Save to EEPROM. Use this on a new install.
M501      ; Read back in the saved EEPROM.
M190 S65  ; Not required, but having the printer at temperature helps accuracy
M104 S210 ; Not required, but having the printer at temperature helps accuracy
G28       ; Home XYZ.
G29 P1    ; Do automated probing of the bed.
G29 P2 B T ; Manual probing of locations USUALLY NOT NEEDED!!!!
G29 P3 T   ; Repeat until all mesh points are filled in.
G29 T     ; View the Z compensation values.
G29 S1    ; Save UBL mesh points to EEPROM.
G29 F 10.0 ; Set Fade Height for correction at 10.0 mm.
G29 A     ; Activate the UBL System.
M500      ; Save current setup. WARNING - UBL will be active at power up, before any `G28`.
```

Use G26 and G29 commands to fine-tune a measured mesh

```
G26 C P T3.0 ; Produce mesh validation pattern with primed nozzle.
; NOTE - PLA temperatures are assumed unless you specify - e.g. - B 105 H 225 for ABS Plastic
G29 P4 T     ; Move nozzle to 'bad' areas and fine tune the values if needed.
; Repeat G26 and G29 P4 T commands as needed.
G29 S1      ; Save UBL mesh values to EEPROM.
M500       ; Resave UBL's state information.
```

Use 3-point probe to 'tilt' a stored mesh; e.g. in your startup script

```
G29 L1      ; Load the mesh stored in slot 1 (from G29 S1)
G29 J       ; No size specified on the J option tells G29 to probe the specified 3 points and tilt the mesh according to what it find
s.
```

## G30 - Single Z-Probe

[↔ 1.0.0-beta](#) [🔧 calibration](#) [📄 Probe bed at current XY location](#) [HAS\\_BED\\_PROBE](#)

Do a single Z probe at a specified position. By default probe in the current position.

## Usage

G30 [E<|>] [X<pos>] [Y<pos>]

Argument	Description
[E< >]	Engage the probe for each point.
[X<pos>]	X probe position
[Y<pos>]	Y probe position

# G31 - Dock Sled

`</> 1.0.0-beta` `planner` `Dock the Z probe sled.` `Z_PROBE_SLED`

Dock the Z probe sled.

## Usage

G31

## Notes

Requires `Z_PROBE_SLED`.

## Example

Dock the sled

```
G31
```

# G32 - Undock Sled

`</> 1.0.0-beta` `planner` `Undock the Z probe sled.` `Z_PROBE_SLED`

Undock the Z probe sled.

## Usage

G32

## Example

Undock the sled

```
G32
```

# G33 - Delta Auto Calibration

`</> 1.1.0` `calibration` `Calibrate various Delta parameters` `DELTA_AUTO_CALIBRATION`

With the `G33` command you can:

- Probe a circular grid of points,
- calibrate Delta Height,
- calibrate endstops,
- calibrate Delta Radius, and
- calibrate Tower Angles.

## Usage

G33 [A<|>] [C<>] [E<|>] [F<>] [P<0|1|2|3|4-10|>] [T<|>] [V<0|1|2|>]

Argument	Description
[A< >]	Auto tune calibration factors.
[C<>]	If omitted iterations stop at best achievable precision; if set force iterations to stop at the set precision.
[E< >]	Engage the probe for each point.
[F<>]	Force to run at least n iterations and take the best result.

Argument	Description
[P<0 1 2 3 4-10 >]	<ul style="list-style-type: none"> <li>• P0 : Normalize end-stops and tower angle corrections only (no probing).</li> <li>• P1 : Probe center and set height only.</li> <li>• P2 : Probe center and towers. Set height, endstops, and delta radius.</li> <li>• P3 : Probe all positions - center, towers and opposite towers. Set all.</li> <li>• P4-10 : Probe all positions + at different intermediate locations and average them.</li> </ul>
[T< >]	Disable tower angle corrections calibration ( P3 - P7 )
[V<0 1 2 >]	Set the verbose level <ul style="list-style-type: none"> <li>• V0 : Dry run, no calibration</li> <li>• V1 : Report settings</li> <li>• V2 : Report settings and probe results</li> </ul>

## Examples

Default (Verbose 1)

```
G33

> G33 Auto Calibrate
> Checking... AC
> .Height:295.00  Ex:+0.00  Ey:+0.00  Ez:+0.00  Radius:100.00
> .Tower angle :  Tx:+0.00  Ty:+0.00  Tz:+0.00
> Iteration : 01  std dev:2.665
> .Height:297.85  Ex:-0.18  Ey:-0.13  Ez:+0.00  Radius:100.68
> .Tower angle :  Tx:-0.05  Ty:+0.08  Tz:+0.00
> Iteration : 02  std dev:0.128
> .Height:297.77  Ex:-0.19  Ey:-0.09  Ez:+0.00  Radius:100.80
> .Tower angle :  Tx:-0.07  Ty:+0.15  Tz:+0.00
> Iteration : 03  std dev:0.025
> .Height:297.78  Ex:-0.17  Ey:-0.09  Ez:+0.00  Radius:100.78
> .Tower angle :  Tx:-0.09  Ty:+0.20  Tz:+0.00
> Iteration : 04  std dev:0.022
> .Height:297.80  Ex:-0.14  Ey:-0.07  Ez:+0.00  Radius:100.79
> .Tower angle :  Tx:-0.10  Ty:+0.22  Tz:+0.00
> Iteration : 05  std dev:0.019
> .Height:297.81  Ex:-0.13  Ey:-0.06  Ez:+0.00  Radius:100.80
> .Tower angle :  Tx:-0.10  Ty:+0.25  Tz:+0.00
> Calibration OK  rolling back.
> .Height:297.80  Ex:-0.14  Ey:-0.07  Ez:+0.00  Radius:100.79
> .Tower angle :  Tx:-0.10  Ty:+0.22  Tz:+0.00
> Save with M500 and/or copy to Configuration.h
```

Verbose 2

```
G33 V2

> G33 Auto Calibrate
> Checking... AC
> .Height:297.80  Ex:-0.14  Ey:-0.07  Ez:+0.00  Radius:100.79
> .Tower angle :  Tx:-0.10  Ty:+0.22  Tz:+0.00
> .      c:+0.01  x:+0.06  y:+0.04  z:+0.01
> .      yz:-0.02  zx:-0.01  xy:+0.01
> Iteration : 01  std dev:0.028
> .Height:297.81  Ex:-0.10  Ey:-0.04  Ez:+0.00  Radius:100.81
> .Tower angle :  Tx:-0.10  Ty:+0.24  Tz:+0.00
> .      c:-0.03  x:-0.01  y:-0.02  z:-0.03
> .      yz:-0.05  zx:-0.05  xy:-0.06
> Calibration OK  rolling back.
> .Height:297.80  Ex:-0.14  Ey:-0.07  Ez:+0.00  Radius:100.79
> .Tower angle :  Tx:-0.10  Ty:+0.22  Tz:+0.00
> Save with M500 and/or copy to Configuration.h
```

Using V0 for Dry Run with no calibration.

G33 V0

```
> G33 Auto Calibrate
> Checking... AC (DRY-RUN)
> .Height:295.00 Ex:+0.00 Ey:+0.00 Ez:+0.00 Radius:100.00
> .Tower angle : Tx:+0.00 Ty:+0.00 Tz:+0.00
> . c:-2.86 x:-2.68 y:-2.62 z:-2.56
> . yz:-2.55 zx:-2.61 xy:-2.78
> End DRY-RUN std dev:2.668
```

Using the T flag for no tower angles.

G33 T

```
> G33 Auto Calibrate
> Checking... AC
> .Height:297.79 Ex:-0.13 Ey:-0.06 Ez:+0.00 Radius:100.83
> Iteration : 01 std dev:0.024
> .Height:297.82 Ex:-0.09 Ey:-0.05 Ez:+0.00 Radius:100.82
> Calibration OK rolling back.
> .Height:297.79 Ex:-0.13 Ey:-0.06 Ez:+0.00 Radius:100.83
> Save with M500 and/or copy to Configuration.h
```

Use a single point ( P1 ) to check the height.

G33 P1

```
> G33 Auto Calibrate
> Checking... AC
> .Height:297.79
> Calibration OK
> .Height:297.80
> Save with M500 and/or copy to Configuration.h
```

---

## G38.2, G38.3 - Probe target

[↔ 1.1.0](#) [🔧 calibration](#)  Probe towards a workpiece and stop on contact. [G38\\_PROBE\\_TARGET](#)

The Probe Target commands are used to probe towards a workpiece and determine its precise position. The Z endstop doubles as the probe for these commands. You might, for example, use a grounded metal workpiece, with a metal probe spliced into the Z endstop circuit.

- G38.2 probes towards a target and stops on contact, signaling an error if it reaches the target position without triggering the Z endstop.
- G38.3 probes towards a target and stops on contact. No error is given if it fails to trigger the Z endstop.

These commands use the current homing feedrate, by default.

---

## Usage

G38.2 [F<rate>] [X<pos>] [Y<pos>] [Z<pos>]

Argument	Description
[F<rate>]	Feedrate for the move
[X<pos>]	Target X
[Y<pos>]	Target Y
[Z<pos>]	Target Z

## Notes

These commands require G38\_PROBE\_TARGET .

---

## G90 - Absolute Positioning

[↔ 1.0.0-beta](#) [🔧 units](#)  Use absolute positions.

In absolute mode all coordinates given in G-code are interpreted as positions in the logical coordinate space. This includes the extruder position unless overridden by M83 (/docs/gcode/M083.html).

## Usage

G90

## Notes

Absolute positioning is the default.

## G91 - Relative Positioning

[↗ 1.0.0-beta](#) [🔗 units](#) [🔗 Use relative positions.](#)

Set relative position mode. In this mode all coordinates are interpreted as relative to the last position. This includes the extruder position unless overridden by M82 (/docs/gcode/M082.html).

## Usage

G91

## G92 - Set Position

[↗ 1.0.0-beta](#) [🔗 planner](#) [🔗 Set the current position of one or more axes.](#)

Set the current position to the values specified. In Marlin 1.1.0 and up, the software endstops are adjusted to preserve the physical movement limits. Thus you could use G92 to set the middle of the bed to 0,0 and then run .gcode that was sliced for a Deltabot.

## Usage

G92 [E<pos>] [X<pos>] [Y<pos>] [Z<pos>]

Argument	Description
[E<pos>]	New extruder position
[X<pos>]	New X axis position
[Y<pos>]	New Y axis position
[Z<pos>]	New Z axis position

## Notes

In earlier versions of Marlin G92 doesn't update the software endstops, so it was unsupported to set coordinates outside these boundaries. In Marlin 1.1.0 and up, the physical boundaries are maintained. This means you can no longer use G92 to move below the bed, for example.

## Examples

Specify that the nozzle's current X position is 10 and the current extruder position is 90.

```
G92 X10 E90
```

Specify that the nozzle's current XYZ position is 0, 0, 0.

```
G92 X0 Y0 Z0
```

## M0, M1 - Unconditional stop



[↔ 1.0.0-beta](#) [🔗 planner](#) [🔗 Stop and wait for user.](#)

The M0 and M1 commands pause after the last movement and wait for the user to continue.

## Usage

M0 [P<ms>] [S<sec>] [string]

Argument	Description
[P<ms>]	Expire time, in milliseconds
[S<sec>]	Expire time, in seconds
[string]	An optional message to display on the LCD

## Notes

If both S and P are included, S takes precedence.

Without an LCD controller or EMERGENCY\_PARSER this command is ignored.

With EMERGENCY\_PARSER enabled the M108 command can be used to continue.

## Examples

Stop and wait

```
M0
```

Display a message, stop, and wait

```
M0 Click to continue
```

---

## M3 - Spindle CW / Laser On

[↔ 1.1.2](#) [🔗 control](#) [🔗 Set the spindle CW speed or laser power](#) [SPINDLE\\_LASER\\_ENABLE](#)

Wait for moves to complete, then set the spindle speed (clockwise) or laser power.

## Usage

M3 [S<power>]

Argument	Description
[S<power>]	Spindle speed or laser power

## Examples

Set spindle rotation clockwise at 50%

```
M3 S128
```

Turn on the laser at full power

```
M3
```

---

## M4 - Spindle CCW / Laser On

`<> 1.1.2` `control` `Set the spindle CCW speed or laser power` `SPINDLE_LASER_ENABLE`

Wait for moves to complete, then set the spindle speed (counter-clockwise) or laser power.

## Usage

M4 [S<power>]

Argument	Description
[S<power>]	Spindle speed or laser power

## Examples

Set spindle rotation counter-clockwise at 50%

```
M4 S128
```

Turn on the laser at full power

```
M4
```

## M5 - Spindle / Laser Off

`<> 1.1.2` `control` `Turn off spindle or laser` `SPINDLE_LASER_ENABLE`

Wait for moves to complete, then turn off the spindle / laser power and PWM.

## Usage

M5

## Examples

Turn off the spindle or laser

```
M5
```

## M17 - Enable Steppers

`control` `Power on all steppers`

Enable power on all stepper motors.

## Usage

M17

## Examples

Enable power on all stepper motors

```
M17
```

## M18, M84 - Disable steppers

`control` `Disable steppers (same as M84).`

This command can be used to set the stepper inactivity timeout ( S ) or to disable one or more steppers ( X , Y , Z , E ).

If a timeout is given with `S`, this command just sets the stepper inactivity timeout.

If no steppers are specified, this command disables all steppers immediately.

If one or more axes are specified, this command disables the specified steppers immediately.

## Usage

M18 [`E<flag>`] [`S<seconds>`] [`X<flag>`] [`Y<flag>`] [`Z<flag>`]

Argument	Description
[ <code>E&lt;flag&gt;</code> ]	E Disable
[ <code>S&lt;seconds&gt;</code> ]	Inactivity Timeout. If none specified, disable now.
[ <code>X&lt;flag&gt;</code> ]	X Disable
[ <code>Y&lt;flag&gt;</code> ]	Y Disable
[ <code>Z&lt;flag&gt;</code> ]	Z Disable

## Examples

Set the stepper inactivity timeout to 1 minute

```
M18 S60
```

Disable all steppers immediately

```
M18
```

Disable Z and E steppers immediately

```
M18 Z E
```

## M20 - List SD Card

`sdcard`  List the contents of the SD Card.  `SDSUPPORT`

List the entire contents of the SD card to serial output in the more compact DOS 8.3 format.

Marlin 1.1.0 includes file sizes in the output.

## Usage

M20

## Notes

Requires `SDSUPPORT` (</docs/configuration/configuration.html#sd-card>)

## M21 - Init SD card

`sdcard`  Attempt to detect an SD card in the slot.  `SDSUPPORT`

Use this command if the SD card isn't detected automatically.

## Usage




M21

## Notes

Requires SDSUPPORT (</docs/configuration/configuration.html#sd-card>)

---

## M22 - Release SD card

If Marlin gets confused about the state of the SD card, this command can be used to simulate an ejection of the SD card.

Re-insert the SD card or use M21 to enable the SD card following M22 .

---

## Usage




M22

## Notes

Requires SDSUPPORT (</docs/configuration/configuration.html#sd-card>)

---

## M23 - Select SD file

## Usage




M23

## Notes

Requires SDSUPPORT (</docs/configuration/configuration.html#sd-card>)

---

## M24 - Start or Resume SD print

Start an SD print or resume the paused SD print. If PARK\_HEAD\_ON\_PAUSE is enabled, unpark the nozzle.

---

## Usage


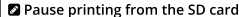

M24

## Notes

Requires SDSUPPORT (</docs/configuration/configuration.html#sd-card>)

---

## M25 - Pause SD print

Pause the SD print in progress. If PARK\_HEAD\_ON\_PAUSE is enabled, park the nozzle.

---

## Usage


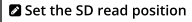

M25

## Notes

Requires SDSUPPORT (</docs/configuration/configuration.html#sd-card>)

---

## M26 - Set SD position

Set the next read position in the open SD file.

---

### Usage


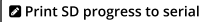

M26

### Notes

Requires `SDSUPPORT` (</docs/configuration/configuration.html#sd-card>)

---

## M27 - Report SD print status

Report the current SD read position in the form "SD printing byte 123/12345".

---

### Usage


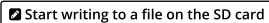

M27

### Notes

Requires `SDSUPPORT` (</docs/configuration/configuration.html#sd-card>)

---

## M28 - Start SD write

This command starts a file write. All commands received by Marlin are written to the file and are not executed until `M29` closes the file.

---

### Usage

M28


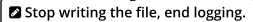

### Notes

Requires `SDSUPPORT` (</docs/configuration/configuration.html#sd-card>)

To write commands to a file while also printing, use `M928` (</docs/gcode/M928.html>)

---

## M29 - Stop SD write

Stop writing to a file that was begun with `M28` or `M928`. Logging is disabled.

---

### Usage



M29

### Notes

Requires `SDSUPPORT` (</docs/configuration/configuration.html#sd-card>)

---

## M30 - Delete SD file

# Usage

M30

## Notes



Requires SDSUPPORT (/docs/configuration/configuration.html#sd-card)

## Example

Delete the file "/path/to/file.gco"

```
M30 /path/to/file.gco
```

## M31 - Print time

This command reports the time elapsed since the start of the current print job to the host. When printing from SD card, the print job timer starts as soon as SD printing starts.


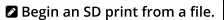

If PRINTJOB\_TIMER\_AUTOSTART is enabled then the first M109 or M190 command received from the host will also start the print job timer.

For manual control from the host, use M75, M76, and M77 to start, pause, and stop the print job timer.

## Usage

M31

## M32 - Select and Start

The M32 command exists to allow G-code to load other G-code files and run them as sub-programs. This can be useful to change the start / end gcode for a batch of files without having to edit them all.

For legacy reasons M32 uses '!' (and '#' ) to delimit the filepath parameter. The filepath must be the last parameter.

## Usage

M32 [P<flag>] [S<filepos>]

Argument	Description
[P<flag>]	Sub-Program flag
[S<filepos>]	Starting file offset

## Notes

Requires SDSUPPORT (/docs/configuration/configuration.html#sd-card)

This is a seldom-used beta feature that needs more testing and use-cases.

## Examples

Select and start a file at offset 5022.

```
M32 S5022 !/boats/sailboat.gco
```

Select and start a file from within G-code.

```
M32 P !/models/1gbust.gco#
```

The # suffix is needed when using P to "stop buffer pre-reading" so no commands after M32 will go into the buffer until after it returns.

## M33 - Get Long Path

`</> 1.0.2` `sdcard` `Convert a short pathname to a long pathname.` `SDSUPPORT` `LONG_FILENAME_HOST_SUPPORT`

### Usage

M33 path

Argument	Description
path	DOS 8.3 path to a file or folder

### Notes

Requires `SDSUPPORT` (</docs/configuration/configuration.html#sd-card>) and `LONG_FILENAME_HOST_SUPPORT`

### Examples

Get the long path for a file

```
M33 funstuff/mask.gco
```

Output

```
FunStuff/Mask.gcode
```

## M34 - SDCard Sorting

`</> 1.1.0` `sdcard` `Set SDCard file sorting options.` `SDSUPPORT` `SDCARD_SORT_ALPHA`

Marlin now contains support for SDCard alphabetical file sorting in the LCD menus. This feature uses free SRAM to create a sorting index for up to the first 256 files in the current folder, and (if you have *lots* of SRAM) can optionally cache file listings for a more responsive UI. Buffering only occurs during file browsing. Otherwise the SRAM is freed.

### Usage

M34 [F<-1|0|1>] [S<bool>]

Argument	Description
[F<-1 0 1>]	Folder Sorting <ul style="list-style-type: none"><li>F-1 : Folders before files</li><li>F0 : No folder sorting</li><li>F1 : Folders after files</li></ul>
[S<bool>]	Sorting on/off

### Notes

Requires `SDSUPPORT` (</docs/configuration/configuration.html#sd-card>) and `SDCARD_SORT_ALPHA`.

## M42 - Set Pin State

`control` `Set an analog or digital pin to a specified state.`



For custom hardware not officially supported in Marlin, you can often just connect up an unused pin and use `M42` to control it.

## Usage

`M42 [P<int>] S<int>`

Argument	Description
[P<int>]	A digital pin number (even for analog pins) to write to. ( <code>LED_PIN</code> if omitted)
S<int>	The state to set. PWM-able pins may be set from 0-255.

## M43 - Debug Pins

   Get information about pins. `PINS_DEBUGGING`

When setting up or debugging a machine it's useful to know how pins are assigned to functions by the firmware, and to be able to find pins for use with new functions. `M43` provides these tools. `M43` by itself reports all pin assignments. Use `P` to specify a single pin. Use `W` to watch the specified pin, or all pins. Use the `E` option to monitor endstops.

The `W` watch mode option continues looping, blocking all further commands, until the board is reset. If `EMERGENCY_PARSER` is enabled, `M108` may also be used to exit the watch loop without needing to reset the board.

## Usage

`M43 [E<bool>] [P<pin>] [W<bool>]`

Argument	Description
[E<bool>]	Watch endstops
[P<pin>]	Digital Pin Number
[W<bool>]	Watch pins

## Notes

Requires `PINS_DEBUGGING`. This feature should be disabled for production use.

## Examples

Get a report on all pins

```
M43
```

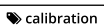
Watch pin 56 for changes

```
M43 P56 W
```

Start watching endstops

```
M43 E
```

## M48 - Probe Accuracy Test

  Measure Z Probe repeatability. `Z_MIN_PROBE_REPEATABILITY_TEST`

Probes come in many flavors and as such have varying levels of accuracy, reliability, and repeatability, depending on several factors. This command tests the probe for accuracy and produces a standard deviation based on two or more probes of the same XY position.



## Usage


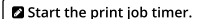
M48 [E<engage>] [L<legs>] [P<count>] [V<level>] [X<pos>] [Y<pos>]

Argument	Description
[E<engage>]	Engage for each probe
[L<legs>]	Number of legs to probe
[P<count>]	Number of probes to do
[V<level>]	Verbose Level
[X<pos>]	X Position
[Y<pos>]	Y Position

## Notes

Requires Z\_MIN\_PROBE\_REPEATABILITY\_TEST .

## M75 - Start Print Job

Start the print job timer.

## Usage


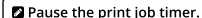
M75

## Example

Start the print job timer

```
M75
```

## M76 - Pause Print Job

Pause the print job timer.

## Usage


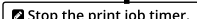
M76

## Example

Pause the print job timer

```
M76
```

## M77 - Stop Print Job

Stop the print job timer.

## Usage

M77

## Example

Stop the print job timer

```
M77
```

## M78 - Print Job Stats

printjob  Print statistics about print jobs.  PRINTCOUNTER

## Usage

M78

## M80 - Power On

control  Turn on the power supply.  POWER\_SUPPLY>0

Turn on the high-voltage power supply. Requires a board that's powered from USB or another 5V source.

## Usage

M80 [S]

Argument	Description
[S]	Report Power Supply state

## Notes

Requires `POWER_SUPPLY` and a digital pin connected to the PSU's enable pin.

## M81 - Power Off

control  Turn off the power supply.  POWER\_SUPPLY>0

Turn off the high-voltage power supply. If the board is not powered from another source, this may also shut down the electronics.

## Usage

M81

## Notes

Requires `POWER_SUPPLY` and a digital pin connected to the PSU's enable pin.

## M82 - E Absolute

units  Set E to absolute positioning.

This command is used to override `G91` and put the E axis into absolute mode independent of the other axes.

## Usage

M82

# M83 - E Relative

units  Set E to relative positioning.

This command is used to override G90 and put the E axis into relative mode independent of the other axes.

## Usage

M83

# M85 - Inactivity Shutdown

control  Set the inactivity timeout.

Use this command to set a maximum period of time for the machine to be inactive (with no moves). If the machine is idle for longer than the set period, the firmware will shut everything down and halt the machine.

## Usage

M85 S<seconds>

Argument	Description
S<seconds>	Max inactive seconds

# M92 - Set Axis Steps-per-unit

planner  Set the number of steps-per-mm or steps-per-inch.

Use M92 to set the steps-per-unit for one or more axes. This setting affects how many steps will be done for each unit of movement. Units will be in steps/mm unless *inch* mode is set with G20 (</docs/gcode/G020.html>) (which requires INCH\_MODE\_SUPPORT).

## Usage

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

Argument	Description
[E<steps>]	E steps per unit
[T<index>]	Target extruder (Requires DISTINCT_E_FACTORS)
[X<steps>]	X steps per unit
[Y<steps>]	Y steps per unit
[Z<steps>]	Z steps per unit

## Notes

Get the current steps-per-unit settings with M503.

With EEPROM\_SETTINGS enabled:

- This setting for all axes is saved with M500 and loaded with M501.
- M502 resets steps-per-unit for all axes to the values from DEFAULT\_AXIS\_STEPS\_PER\_UNIT.

## Example

Set E steps for a new extruder

```
M92 E688.4
```

---

# M100 - Free Memory

Use `M100` for development purposes to observe how much memory (particularly stack) is being used by code. Proper AVR code should avoid use of `new`, `malloc`, etc., and instead use either pre-allocated static variables or stack.

## Usage

`M100` [`C<n>`] [`D`] [`F`] [`I`]


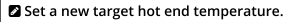
Argument	Description
[ <code>C&lt;n&gt;</code> ]	Corrupt ' <code>n</code> ' locations in the free memory pool and report the locations of the corruption. This is useful to check the correctness of the <code>M100 D</code> and <code>M100 F</code> commands.
[ <code>D</code> ]	Dump the free memory block from <code>__brkva1</code> to the stack pointer.
[ <code>F</code> ]	Return the number of free bytes in the memory pool along with other vital statistics that define the memory pool.
[ <code>I</code> ]	Initialize the free memory pool so it can be watched and print vital statistics that define the free memory pool.

## Notes

Requires `M100_FREE_MEMORY_WATCHER`.

---

# M104 - Set Hotend Temperature

Set a new target hot end temperature and continue without waiting. The firmware will continue to try to reach and hold the temperature in the background.

Use `M109` (</docs/gcode/M109.html>) to wait for the hot end to reach the target temperature.

## Usage

`M104` [`B<temp>`] [`F<flag>`] [`S<temp>`]

Argument	Description
[ <code>B&lt;temp&gt;</code> ]	AUTOTEMP : The max auto-temperature.
[ <code>F&lt;flag&gt;</code> ]	AUTOTEMP : Autotemp flag. Omit to disable autotemp.
[ <code>S&lt;temp&gt;</code> ]	Target temperature. AUTOTEMP : the min auto-temperature.

## Notes

- With `PRINTJOB_TIMER_AUTOSTART` this command will stop the print job timer if the temperature is set at or below half of `EXTRUDE_MINTEMP`.

## Examples

Simple set target temperature

```
M104 S180
```

AUTOTEMP : Set autotemp range



```
M104 F S180 B190
```

AUTOTEMP : Disable autotemp

```
M104
```

---

## M105 - Report Temperatures

Request a temperature report to be sent to the host at some point in the future.

### Usage

M105 [T<index>]

Argument	Description
[T<index>]	Hotend index

### Notes

Some hosts may hide the reply from M105 .

A better way for hosts to get regular temperature updates is to use M155 (requires AUTO\_REPORT\_TEMPERATURES and EXTENDED\_CAPABILITIES\_REPORT ). Hosts then no longer need to run an extra process or use up slots in the command buffer to receive temperatures.


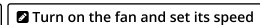
### Examples

Get a temperature report

```
M105
```

---

## M106 - Set Fan Speed

Turn on one of the fans and set its speed. If no fan index is given, the print cooling fan.

### Usage

M106 [P<index>] [S<speed>]

Argument	Description
[P<index>]	Fan index
[S<speed>]	Speed

### Notes

M106 with no speed sets the fan to full speed.

Turn off fans with M107 (</docs/gcode/M107.html>).

---

## M107 - Fan Off

Turn off one of the fans. If no fan index is given, the print cooling fan.

## Usage


M107 [P<index>]

Argument	Description
[P<index>]	Fan index

## Notes

Turn on fans with M106 ([docs/gcode/M106.html](/docs/gcode/M106.html)).

## M108 - Break and Continue

  Break out of the current waiting loop

The M108 command requires EMERGENCY\_PARSER for full effectiveness. (Otherwise a full queue blocks the parser.)

Some G-code commands cause Marlin to go into a closed loop, waiting indefinitely for a certain state or event. For example, M109 waits for the target temperature to be reached, and M0 waits for an LCD click.

- In the case of M109, the M108 command stops waiting for the target temperature and continues processing G-code. This may result in “cold extrude” messages. For a full stop use M112.
- In the case of M0 the M108 command acts like the LCD button, breaking out of M0 and continuing to process the G-code queue.

## Usage

M108

## Notes


With both EMERGENCY\_PARSER and HOST\_KEEPALIVE\_FEATURE enabled, hosts will be able to prompt for continuation or cancellation, confirming with M108 and cancelling with M112.

## Example

Use M108 as a “Continue” button in your host software.

```
M0 You're up, mate ; in your G-code file
M108 ; as your "Continue" button
```

## M109 - Wait for Hotend Temperature

  Wait for the hot end to reach its target.

This command optionally sets a new target hot end temperature and waits for the target temperature to be reached before proceeding. If the temperature is set with S then M109 waits *only when heating*. If the temperature is set with R then M109 will also wait for the temperature to go down.

## Usage

M109 [B<temp>] [F<flag>] [R<temp>] [S<temp>]

Argument	Description
[B<temp>]	With AUTOTEMP, the max auto-temperature.
[F<flag>]	Autotemp flag. Omit to disable autotemp.
[R<temp>]	Target temperature (wait for cooling or heating).

Argument	Description
[S<temp>]	Target temperature (wait only when heating). Also AUTOTEMP : The min auto-temperature.

## Notes

With PRINTJOB\_TIMER\_AUTOSTART this command will start the print job if heating, and stop the print job timer if the temperature is set at or below half of EXTRUDE\_MINTEMP .

This command (and M190 (/docs/gcode/M190.html)) can block new commands from the host, preventing remote shutdown. However, if EMERGENCY\_PARSER is enabled, a host can send M108 to break out of the wait loop.

To set the hot end temperature and proceed without waiting, use M104 (/docs/gcode/M104.html).

## Examples

Set target temperature and wait (if heating)

```
M109 S180
```

Set target temperature, wait even if cooling

```
M109 R120
```


AUTOTEMP : Set autotemp range, wait for temp

```
M109 F S180 B190
```

AUTOTEMP : Disable autotemp, wait for temp

```
M109
```

## M110 - Set Line Number

  Set the current line number.

Hosts can use M110 to set the current line number in a print job. Each line number sent by a host must be one higher than the previous line number, or the firmware will ignore the line and send an error requesting a resend of the missing line. This is one technique Marlin uses to keep in sync with hosts.

## Usage


M110 N<line>

Argument	Description
N<line>	Line number

## Notes

All these are valid: N100 M110 , M110 N100 , N101 M110 N100 .

## M111 - Debug Level

  Report and optionally set the debug flags.

Marlin has several debug bits that can be set, in combination, to help configure, troubleshoot, and debug the firmware. Add up the debug bits you need:

Mask	Name	Description
1	ECHO	Echo all commands sent to the parser.
2	INFO	Print extra informational messages.
4	ERRORS	Print extra error messages.
8	DRYRUN	Don't extrude, don't save leveling data, etc.

16	COMMUNICATION	Not currently used.
32	LEVELING	Detailed messages for homing, probing, and leveling. (Requires <code>DEBUG_LEVELING_FEATURE</code> .)
64	Reserved	Reserved for future usage
128	Reserved	Reserved for future usage

## Usage

M111 [`S<flags>`]

Argument	Description
[ <code>S&lt;flags&gt;</code> ]	Debug flag bits

## Examples

Enable extra messages

```
M111 S38 ; LEVELING, ERRORS, INFO
```

Enable dry-run mode

```
M111 S8
```



Enable everything except dry-run mode

```
M111 S247 ; 255 - 8
```

Disable previously set extra debugging output.

```
M111 S0
```

## M112 - Emergency Stop

 safety  Shut everything down and halt the machine.

Used for emergency stopping, `M112` shuts down the machine, turns off all the steppers and heaters, and if possible, turns off the power supply. A reset is required to return to operational mode.

## Usage

M112

## Notes


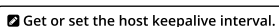
`M112` is the fastest way to shut down the machine using a host, but it may need to wait for a space to open up in the command queue. Enable `EMERGENCY_PARSER` for an instantaneous `M112` command.

## Examples

Shut down now!

```
M112
```

## M113 - Host Keepalive

 hosts  Get or set the host keepalive interval. `HOST_KEEPALIVE_FEATURE`

During some lengthy processes, such as `G29`, Marlin may appear to the host to have “gone away.” The “host keepalive” feature will send messages to the host when Marlin is busy or waiting for user response so the host won’t try to reconnect.



# Usage

M113 [S<seconds>]

Argument	Description
[S<seconds>]	Keepalive interval (0-60)

## Notes

Requires HOST\_KEEPALIVE\_FEATURE .

---

## M114 - Get Current Position

🔗 hosts  Report the current tool position to the host.

Get the current position of the active nozzle. Includes stepper values.

---

## Usage

M114

## Notes

Hosts should respond to the output of M114 by updating their current position.

## Examples

Get the current position

```
M114
```

---

## M115 - Firmware Info

🔗 hosts  Print the firmware info and capabilities.

This command causes Marlin to output a string like this:

```
FIRMWARE_NAME:Marlin 1.1.0 (Github) SOURCE_CODE_URL:https://github.com/MarlinFirmware/Marlin PROTOCOL_VERSION:1.0 MACHINE_TYPE:RepRap E
XTRUDER_COUNT:1 UUID:cede2a2f-41a2-4748-9b12-c55c62f367ff
```

When EXTENDED\_CAPABILITIES\_REPORT is enabled, Marlin also reports its capabilities:

```
Cap:EEPROM:1
Cap:AUTOREPORT_TEMP:1
Cap:PROGRESS:0
Cap:AUTOLEVEL:1
Cap:Z_PROBE:1
Cap:SOFTWARE_POWER:0
Cap:TOGGLE_LIGHTS:0
Cap:EMERGENCY_PARSER:1
```

Hosts use this information to improve interoperability, so it's a good feature to enable.

---

## Usage

M115

---

## M117 - Set LCD Message

🔗 lcd  Set the message line on the LCD.  ULTRA\_LCD

Set the status line message on the LCD.

## Usage

M117 [string]

Argument	Description
[string]	LCD status message

## Notes

Requires an LCD controller.

The message should appear immediately, but it will depend on LCD settings.

## Examples

Set the message to "Yello World!"

```
M117 Yello World!
```

---

## M119 - Endstop States

debug  Report endstop and probe states to the host.

Use this command to get the current state of all endstops, useful for setup and troubleshooting. Endstops are reported as either " open " or " TRIGGERED ".

The state of the Z probe is also reported.

## Usage

M119

## Examples

Get all endstop states

```
M119
```

---

## M120 - Enable Endstops

control  Enable endstops and keep them enabled when not homing.

Enable endstops.

## Usage

M120

## Notes

After this command endstops will be kept enabled when not homing. This may have side-effects if using `ABORT_ON_ENDSTOP_HIT_FEATURE_ENABLED` .



## Examples

Enable endstops

```
M120
```

---

# M121 - Disable Endstops

  Disable endstops and keep them enabled when not homing.

Disable endstops.

---

## Usage

M121

## Notes

After this command endstops will be kept disabled when not homing. This may have side-effects if using `ABORT_ON_ENDSTOP_HIT_FEATURE_ENABLED`.




## Examples

Disable endstops

```
M121
```

---

# M122 - TMC Debugging

 1.1.7   Get TMC Debug Info `HAVE_TMC2130|HAVE_TMC2208` `TMC_DEBUG`

When sent without a parameter, returns a table of current register settings for any Trinamic TMC2130 or TMC2208 stepper motor drivers. Sending the command with the `s` parameter and a following boolean will respectively enable or disable reporting the debugging information on a continuous basis.

---

## Usage

M122 [S]

Argument	Description
[S]	Flag to enable/disable continuous reporting of debugging information.

## Notes

Need to have `TMC_DEBUG` enabled in `Configuration_adv.h`.

## Examples

Enabling debugging output:

```
M122 S1
```

Example Output:

```

SENDING:M122
X      Y
Enabled      false  false
Set current  850    850
RMS current  826    826
MAX current  1165   1165
Run current  26/31  26/31
Hold current 13/31  13/31
CS actual    13/31  13/31
PWM scale    41     41
vsense       1=.18  1=.18
stealthChop  true    true
msteps       16     16
tstep        1048575 1048575
pwm
threshold    0     0
[mm/s]       -     -
OT prewarn   false  false
OT prewarn has
been triggered false  false
off time     5     5
blank time   24    24
hysteresis
-end         2     2
-start      3     3
Stallguard thrs 0    0
DRVSTATUS    X     Y
stallguard
sg_result    0     0
fsactive
stst         X     X
o1b
o1a
s2gb
s2ga
otpw
ot
'Driver registers:'
  X = 0x80:0D:00:00
  Y = 0x80:0D:00:00

```

## M125 - Park Head

nozzle
  Save current position and move to filament change position.
  PARK\_HEAD\_ON\_PAUSE

Save the current nozzle position and move to the configured park position.

## Usage

M125 [L<linear>] [X<linear>] [Y<linear>] [Z<linear>]

Argument	Description
[L<linear>]	Retract length (otherwise FILAMENT_CHANGE_RETRACT_LENGTH)
[X<linear>]	X position to park at (otherwise FILAMENT_CHANGE_X_POS)
[Y<linear>]	Y position to park at (otherwise FILAMENT_CHANGE_Y_POS)
[Z<linear>]	Z raise before park (otherwise FILAMENT_CHANGE_Z_ADD)

## Notes

Requires PARK\_HEAD\_ON\_PAUSE.

## Examples

Retract 2cm of filament and park the nozzle

M125 L20 ; park and retract

---

## M126 - Baricuda 1 Open

baricuda  Open the valve for Baricuda 1.  BARICUDA

Open the valve for Baricuda paste extruder 1.

---

### Usage

M126

### Notes

Requires BARICUDA .

### Examples

M126 ; open valve 1

---

## M127 - Baricuda 1 Close

baricuda  Close the valve for Baricuda 1.  BARICUDA

Close the valve for Baricuda paste extruder 1.

---

### Usage

M127

### Notes

Requires BARICUDA .

### Examples

M127 ; close valve 1

---

## M128 - Baricuda 2 Open

baricuda  Open the valve for Baricuda 2.  BARICUDA

Open the valve for Baricuda paste extruder 2.

---

### Usage

M128

### Notes


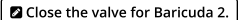

Requires BARICUDA .

### Examples

M128 ; open valve 2

---

# M129 - Baricuda 2 Close


Close the valve for Baricuda paste extruder 2.

---

## Usage

M129

## Notes


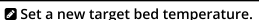
Requires .

## Examples

```
M129 ; cclose valve 2
```

---

# M140 - Set Bed Temperature

Set a new target heated bed temperature and continue without waiting. The firmware will continue to try to reach and hold the temperature in the background.

---

Use `M190` (</docs/gcode/M190.html>) to wait for the bed to reach the target temperature.

---

## Usage

M140 [S<temp>]

Argument	Description
[S<temp>]	Target temperature. AUTOTEMP : the min auto-temperature.




## Examples

Simple set target temperature

```
M140 S80
```

---

# M145 - Set Material Preset

Set the preheating presets for materials in the LCD menu.

---

## Usage

M145 [B<temp>] [F<speed>] [H<temp>] [S<index>]

Argument	Description
[B<temp>]	Bed temperature
[F<speed>]	Fan speed

Argument	Description
[H<temp>]	Hotend temperature
[S<index>]	Material index

## Notes

Requires an LCD controller.

View the current settings with `M503`.


If `EEPROM_SETTINGS` is enabled, these settings are saved with `M500`, loaded with `M501`, and reset with `M502`.

## Examples

Set heatup presets for material 1

```
M145 S0 H190 B70 F50
```

## M149 - Set Temperature Units

  Set temperature units to Celsius, Fahrenheit, or Kelvin.


Set temperature units to Celsius, Fahrenheit, or Kelvin. Celsius is the default.

## Usage

M149 [C] [F] [K]

Argument	Description
[C]	Celsius
[F]	Fahrenheit
[K]	Kelvin

## M150 - Set RGB Color

  Set the color of the RGB LED, backlight, or LED strip. **BLINKM|RGB\_LED**

If you have an RGB light, either as part of a controller or installed separately, the `M150` command can be used to set its color.

## Usage

M150 [B<intensity>] [R<intensity>] [U<intensity>]

Argument	Description
[B<intensity>]	Blue component from 0 to 255
[R<intensity>]	Red component from 0 to 255
[U<intensity>]	Green component from 0 to 255

## Notes

Requires BLINKM or RGB\_LED .

---

## M155 - Temperature Auto-Report

hosts  Auto-report temperatures to host periodically. `AUTO_REPORT_TEMPERATURES` `EXTENDED_CAPABILITIES_REPORT`

It can be useful for host software to track temperatures, display and graph them over time, but polling with M105 is less than optimal. With M155 hosts simply set an interval and Marlin will keep sending data automatically. This method is preferred over polling with M105 .

---

### Usage

M155 [S<seconds>]

Argument	Description
[S<seconds>]	Interval in seconds between auto-reports. S0 to disable.

### Notes

Requires `AUTO_REPORT_TEMPERATURES` in `Configuration_adv.h` .

Also enable `EXTENDED_CAPABILITIES_REPORT` to notify hosts about this capability.

### Examples

Report temperatures every 4 seconds

```
M155 S4
```

Stop reporting temperatures

```
M155 S0
```

---

## M163 - Set Mix Factor

mixing  Set a single mix factor for a mixing extruder. `MIXING_EXTRUDER`

Set a single mix factor from 0 to 1. (The mix will be "normalized" by M164 before saving.)

---

### Usage

M163 [P<factor>] [S<index>]

Argument	Description
[P<factor>]	Mix factor
[S<index>]	Component index

### Notes

Requires `MIXING_EXTRUDER` .

### Examples

Save a 60/40 mix as tool index 5:



```
M163 S0 P0.6
M163 S1 P0.4
M164 S5
```

Save a 3/5 mix as tool index 4:

```
M163 S0 P3
M163 S1 P5
M164 S4
```

---

## M164 - Save Mix

mixing  Save the current mix as a virtual tool.  MIXING\_EXTRUDER  MIXING\_VIRTUAL\_TOOLS

### Usage

M164 S<index>

Argument	Description
S<index>	Tool index (0 if none)

### Notes

Requires MIXING\_EXTRUDER and MIXING\_VIRTUAL\_TOOLS.

### Examples

Save a 60/40 mix as tool index 5:

```
M163 S0 P0.6
M163 S1 P0.4
M164 S5
```

Save a 3/5 mix as tool index 4:

```
M163 S0 P3
M163 S1 P5
M164 S4
```

---

## M165 - Set Mix

mixing  Set all mix factors for the mixing extruder.  MIXING\_EXTRUDER  DIRECT\_MIXING\_IN\_G1

Set the mix all at once. Any factors left out are set to 0.0. This is based on a reference implementation by Pia Taubert.

### Usage

M165 [A<factor>] [B<factor>] [C<factor>] [D<factor>] [H<factor>] [I<factor>]

Argument	Description
[A<factor>]	Mix factor 1
[B<factor>]	Mix factor 2
[C<factor>]	Mix factor 3
[D<factor>]	Mix factor 4
[H<factor>]	Mix factor 5

Argument	Description
[I<factor>]	Mix factor 6

## Notes



Requires MIXING\_EXTRUDER and DIRECT\_MIXING\_IN\_G1 .

## Examples

Set a colorful mix

```
M165 A0.2 B0.4 C0.3 D0.1
```

## M190 - Wait for Bed Temperature

  Wait for the bed to reach target temperature.

This command optionally sets a new target bed temperature and waits for the target temperature to be reached before proceeding. If the temperature is set with `S` then it waits *only when heating*.

## Usage

M190 [R<temp>] [S<temp>]

Argument	Description
[R<temp>]	Target temperature (wait for cooling or heating).
[S<temp>]	Target temperature (wait only when heating). Also AUTOTEMP : The min auto-temperature.

## Notes

This command (and M109 (</docs/gcode/M109.html>)) can block new commands from the host. To break out of wait for temperature using M108 from the host, enable EMERGENCY\_PARSER .

Use M140 (</docs/gcode/M140.html>) to set the bed temperature and proceed without waiting.

## Examples



Set target bed temperature and wait (if heating)

```
M190 S80
```

Set target bed temperature, wait even if cooling

```
M190 R40
```

## M200 - Set Filament Diameter

  Set the diameter for volumetric extrusion.

Set the filament's current diameter and enable volumetric extrusion.

In volumetric extrusion mode the E axis specifies cubic mm instead of linear mm, and the firmware calculates how much length to extrude for the given volume based on the filament diameter.

# Usage

M200 [D<diameter>]

Argument	Description
[D<diameter>]	Filament diameter

## Examples

A common diameter close to 3mm:


```
M200 D2.85
```

Turn off volumetric extrusion

```
M200 D0  
M200 D ; also works
```

---

## M201 - Set Print Max Acceleration

  Set maximum acceleration for print moves one or more axes.

Set the max acceleration for one or more axes (in current units-per-second squared).

## Usage

M201 [E<accel>] [T<index>] [X<accel>] [Y<accel>] [Z<accel>]

Argument	Description
[E<accel>]	E axis max acceleration
[T<index>]	Target extruder (Requires DISTINCT_E_FACTORS )
[X<accel>]	X axis max acceleration
[Y<accel>]	Y axis max acceleration
[Z<accel>]	Z axis max acceleration

## Notes

View the current setting with M503 .

If EEPROM\_SETTINGS is enabled, these are saved with M500 , loaded with M501 , and reset with M502 .


## Examples

Set max acceleration lower so it sounds like a robot:

```
M201 X50 Y50
```

---

## M203 - Set Max Feedrate

  Set maximum feedrate for one or more axes.

Set the max feedrate for one or more axes (in current units-per-second).

# Usage

M203 [E<accel>] [T<index>] [X<accel>] [Y<accel>] [Z<accel>]

Argument	Description
[E<accel>]	E axis max feedrate
[T<index>]	Target extruder (Requires DISTINCT_E_FACTORS )
[X<accel>]	X axis max feedrate
[Y<accel>]	Y axis max feedrate
[Z<accel>]	Z axis max feedrate

## Notes

View the current setting with M503 .



If EEPROM\_SETTINGS is enabled, these are saved with M500 , loaded with M501 , and reset with M502 .

## Examples

Set max feedrate for XY to 100mm/s:

```
M203 X100 Y100
```

## M204 - Set Starting Acceleration

  Set the starting acceleration for moves by type.

Set the preferred starting acceleration for moves of different types.

## Usage

M204 [P<accel>] [R<accel>] [T<accel>]



Argument	Description
[P<accel>]	Printing acceleration
[R<accel>]	Retract acceleration
[T<accel>]	Travel acceleration

## Notes

View the current setting with M503 .

If EEPROM\_SETTINGS is enabled, these are saved with M500 , loaded with M501 , and reset with M502 .

## M205 - Set Advanced Settings

  Set some advanced settings related to movement.

Set various motion settings. See parameters for details.

## Usage

M205 [B<μs>] [E<jerk>] [S<feedrate>] [T<feedrate>] [X<jerk>] [Y<jerk>] [Z<jerk>]

Argument	Description
[B<μs>]	Minimum segment time (μs)
[E<jerk>]	E max jerk (units/s)
[S<feedrate>]	Minimum feedrate for print moves (units/min)
[T<feedrate>]	Minimum feedrate for travel moves (units/min)
[X<jerk>]	X max jerk (units/s)
[Y<jerk>]	Y max jerk (units/s)
[Z<jerk>]	Z max jerk (units/s)

## Notes

View the current setting with M503 .

If EEPROM\_SETTINGS is enabled, these are saved with M500 , loaded with M501 , and reset with M502 .

## Examples

Set some advanced settings.

```
M205 T2400
```

## M206 - Set Home Offsets

[planner](#) [Description Here](#) `HAS_HOME_OFFSET`

Use M206 to apply a persistent offset to the native home position and coordinate space. This effectively shifts the coordinate space in the negative direction. See examples below.

- The current position is adjusted to align to the new home offset values.
- The home offset is persistent — added to the current position until changed.
- Some uses include fine adjustment of Z position (without moving endstops) and shifting the coordinate space to print on a different part of the bed.

## Usage

M206 [P<offset>] [T<offset>] [X<offset>] [Y<offset>] [Z<offset>]

Argument	Description
[P<offset>]	SCARA Psi offset (Requires MORGAN_SCARA )
[T<offset>]	SCARA Theta offset (Requires MORGAN_SCARA )
[X<offset>]	X home offset
[Y<offset>]	Y home offset
[Z<offset>]	Z home offset

## Notes

- This command isn't available on DELTA . (For delta use M665 H .)

- This GCode can be disabled with `NO_WORKSPACE_OFFSETS` to optimize movement.
- Changing the home offsets will not invalidate bed leveling or other saved data.
- View the current offsets with `M503`.
- If `EEPROM_SETTINGS` is enabled, the home offsets are saved with `M500`, loaded with `M501`, and reset with `M502`.
- `M428` (</docs/gcode/M428.html>) sets home offsets so the current position aligns to the native home position.

## Examples

Raise Z up a little bit on the first layer:

```
M206 Z-0.2
```

Shift the print area 10mm to the left:

```
M206 X10
```

## M207 - Set Firmware Retraction

[planner](#)

[Set options for firmware-based retraction.](#)

**FWRETRACT**

Set lengths, feedrate, and Z lift for firmware-based retraction. See parameters below.

### Usage

M207 [F<feedrate>] [S<length>] [W<length>] [Z<length>]

Argument	Description
[F<feedrate>]	Retract feedrate (units/min)
[S<length>]	Retract length
[W<length>]	Retract swap length (multi-extruder)
[Z<length>]	Z lift on retraction

### Notes

Requires `FWRETRACT`.

See related codes `G10` (</docs/gcode/G010.html>), `G11` (</docs/gcode/G010.html>), `M208` (</docs/gcode/M208.html>), and `M209` (</docs/gcode/M209.html>).

## M208 - Set Firmware Recovery

[planner](#)

[Set values for firmware-based retract recovery.](#)

**FWRETRACT**

Set lengths and feedrate for firmware-based retract recovery. The new values will apply to all subsequent G11

### Usage

M208 [F<feedrate>] [S<length>] [W<length>]




Argument	Description
[F<feedrate>]	Recover feedrate (units/min)
[S<length>]	Recover length
[W<length>]	Recover swap length (multi-extruder)

# Notes

Requires FWRETRACT .

See related codes G10 (/docs/gcode/G010.html), G11 (/docs/gcode/G010.html), M207 (/docs/gcode/M207.html), and M209 (/docs/gcode/M209.html).

## M209 - Set Auto Retract

  Enable / disable auto-retraction. 

Enable or disable automatic retraction. This option is meant to help slicers that don't support G10 / G11 . But it can be used to override retraction in any GCode.

When auto-retract is enabled, all reversed E-only moves are treated as retraction. (Recover moves are also automatically overridden.) When disabled, E retraction derives from G-code.

## Usage

M209 S<flag>

Argument	Description
S<flag>	Set Auto-Retract on/off

## Notes




Requires FWRETRACT .

Most slicers today can generate G10 / G11 . But this option is useful for older G-code.

Though not currently very popular, both volumetric extrusion and firmware-based retraction (and/or M209 ) make G-code more immune to changes from one machine to another, and permit changing the hardware without needing to re-slice.

See related codes G10 (/docs/gcode/G010.html), G11 (/docs/gcode/G010.html), M207 (/docs/gcode/M207.html), and M208 (/docs/gcode/M208.html).

## M211 - Software Endstops

  Set and/or get the software endstops state 

Optionally enable/disable software endstops, then report the current state.

With software endstops enabled, moves will be clipped to the physical boundaries from [XYZ]\_MIN\_POS to [XYZ]\_MAX\_POS .

## Usage




M211 [S<flag>]

Argument	Description
[S<flag>]	Software endstops state

## Notes

Requires either MIN\_SOFTWARE\_ENDSTOPS or MAX\_SOFTWARE\_ENDSTOPS for the enable option.

## M218 - Set Hotend Offset

  Set the offset of a hotend (from hotend 0). 

To keep nozzles aligned to the work area between tool-changes, the firmware needs to know how they relate to each other.

# Usage



M218 [T<index>] [X<offset>] [Y<offset>] [Z<offset>]

Argument	Description
[T<index>]	Hotend index. Active extruder by default.
[X<offset>]	Hotend X offset
[Y<offset>]	Hotend Y offset
[Z<offset>]	Hotend Z offset. Requires DUAL_X_CARRIAGE or SWITCHING_NOZZLE .

## Notes

- Requires 2 or more nozzles.
- The default hotend offsets are set with HOTEND\_OFFSET\_[XYZ] .
- Z hotend offset only available with DUAL\_X\_CARRIAGE or SWITCHING\_NOZZLE .
- View current hotend offsets with M503 .
- If EEPROM\_SETTINGS is enabled, these offsets are saved with M500 , loaded with M501 , and reset with M502 .

## M220 - Set Feedrate Percentage

  Set the global feedrate percentage.



Set the feedrate percentage, which applies to all G-code-based moves in all (X, Y, Z, and E) axes.

## Usage

M220 S<percent>

Argument	Description
S<percent>	Feedrate percentage

## M221 - Set Flow Percentage

  Set the flow percentage, which applies to all E moves.

Set the flow percentage, which applies to all E moves added to the planner.

## Usage

M221 S<percent> [T<index>]

Argument	Description
S<percent>	Feedrate percentage
[T<index>]	Target extruder (requires multi-extruder). Default is the active extruder.

## Examples

Set the flow rate to 150%.

```
M221 S150
```



# M226 - Wait for Pin State

control  Wait for a pin to have a given state.

Wait for a pin to have a certain value or state.

## Usage

M226 P<pin> [S<state>]

Argument	Description
P<pin>	Pin number
[S<state>]	State 0 or 1. Default -1 for inverted.

# M240 - Trigger Camera

extras  Trigger camera snapshot or recording.  CHDK|PHOTOGRAPH\_PIN

Trigger the photograph pin to take a snapshot or toggle recording.

## Usage

M240

## Notes

Requires CHDK or PHOTOGRAPH\_PIN . See this article ([http://www.doc-diy.net/photo/rc-1\\_hacked/](http://www.doc-diy.net/photo/rc-1_hacked/)) for more info.

# M250 - LCD Contrast

lcd  Set and/or get the LCD contrast.  HAS\_LCD\_CONTRAST

Set and/or get the LCD contrast. The value is constrained based on the LCD.

## Usage

M250 [C<contrast>]

Argument	Description
[C<contrast>]	Contrast value

## Notes

Requires an LCD controller with software-controlled contrast.

# M260 - I2C Send

i2c  Send data to the I2C bus.  EXPERIMENTAL\_I2CBUS

Utility to send data over the I2C bus.

## Usage

M260 [A<addr>] [B<byte>] [R<flag>] [S<flag>]

Argument	Description
----------	-------------

Argument	Description
[A<addr>]	The bus address to send to
[B<byte>]	The byte to add to the buffer
[R<flag>]	Reset and rewind the I2C buffer
[S<flag>]	Send flag. Flush the buffer to the bus.

## Notes

Requires EXPERIMENTAL\_I2CBUS .

## Examples

Send "Marlin" to the slave device with address 0x63 (99)


```
M260 A99 ; Target slave address
M260 B77 ; M
M260 B97 ; a
M260 B114 ; r
M260 B108 ; l
M260 B105 ; i
M260 B110 ; n
M260 S1 ; Send the current buffer
```

Request 6 bytes from slave device with address 0x63 (99)

```
M261 A99 B5
```

```
i2c-reply: from:99 bytes:5 data:hello
```

## M261 - I2C Request

  Request and echo bytes from the I2C bus. EXPERIMENTAL\_I2CBUS

Request bytes from the I2C bus and echo them to the host. To find out how to do more useful things with I2C see the I2C master / slave article.

## Usage


M261 A<addr> B<count>

Argument	Description
A<addr>	The bus address to request bytes from
B<count>	The number of bytes to request

## Notes

Requires EXPERIMENTAL\_I2CBUS .

## M280 - Servo Position

  Set or get a servo position. NUM\_SERVOS>0

Set or get the position of a servo.

# Usage

M280 P<index> S<pos>

Argument	Description
P<index>	Servo index to set or get
S<pos>	Servo position to set. Omit to read the current position.

## Notes

Requires NUM\_SERVOS of 1 or more.

## M290 - Babystep

[↔ 1.1.7](#) [🔧 calibration](#) [📌 Babystep one or more axes](#) **BABYSTEPPING**

Apply babysteps to one or more axes using current units. Offsets applied with M290 aren't added to the current coordinates, but are intended for making small adjustments, especially in the Z axis, at the start of a print.

Note that when BABYSTEP\_ZPROBE\_OFFSET is enabled, M290 also modifies the Probe Z Offset (with no immediate effects). The new Z offset applies to successive probing operations, and can be saved with M500. This behavior is means to coincide with the LCD Menu replacing "Z Babystepping" with "Babystep Z Probe Offset." To avoid this side-effect, use M290 P0 or leave BABYSTEP\_ZPROBE\_OFFSET disabled.

## Usage

M290 [P<bool>] [S<pos>] [X<pos>] [Y<pos>] [Z<pos>]

Argument	Description
[P<bool>]	Use P0 to leave the Probe Z Offset unaffected. (Requires BABYSTEP_ZPROBE_OFFSET)
[S<pos>]	Alias for Z
[X<pos>]	A distance on the X axis
[Y<pos>]	A distance on the Y axis
[Z<pos>]	A distance on the Z axis

## Notes

Requires BABYSTEP\_XY for babystepping on the XY axes.

## Example

Babystep the Z axis by 0.25mm (in mm units mode)

```
M290 Z0.25 ; move up 0.25mm on the Z axis
```

## M300 - Play Tone

[🔊 lcd](#) [📌 Play a single tone, buzz, or beep.](#) **SPEAKER**

Add a tone to the tone queue.

# Usage

M300 [P<ms>] [S<Hz>]

Argument	Description
[P<ms>]	Duration (1s)
[S<Hz>]	Frequency (260Hz)

## Notes

Requires `SPEAKER` to play tones (not just beeps).

In Marlin 1.0.2, playing tones block the command queue. Marlin 1.1.0 uses a tone queue and background tone player to keep the command buffer from being blocked by playing tones.



## Examples

Play a tune.

```
M300 S440 P200
M300 S660 P250
M300 S880 P300
```

---

## M301 - Set Hotend PID

  **PIDTEMP**

Set the values that control the PID loop for a hotend.

## Usage

M301 [C<value>] [D<value>] [E<index>] [I<value>] [L<value>] [P<value>]

Argument	Description
[C<value>]	C term (requires <code>PID_EXTRUSION_SCALING</code> )
[D<value>]	Derivative value
[E<index>]	Extruder index to set. Default 0.
[I<value>]	Integral value
[L<value>]	Extrusion scaling queue length (requires <code>PID_EXTRUSION_SCALING</code> )
[P<value>]	Proportional value

## Notes



Requires `PIDTEMP`.

View current PID values with `M503`.

If `EEPROM_SETTINGS` is enabled, all hotend PID values are saved with `M500`, loaded with `M501`, and reset with `M502`.

---

## M302 - Cold Extrude

  **PREVENT\_COLD\_EXTRUSION**

Set the minimum extrusion temperature, potentially allowing E movement at temperatures below the melting point of the material.

## Usage

M302 [P<flag>] [S<temp>]

Argument	Description
[P<flag>]	Flag to allow extrusion at any temperature
[S<temp>]	Minimum temperature for safe extrusion

## Notes


Requires PREVENT\_COLD\_EXTRUSION .

## Examples

Some common uses...

```
M302          ; report current cold extrusion state
M302 P0       ; enable cold extrusion checking
M302 P1       ; disable cold extrusion checking
M302 S0       ; always allow extrusion (disable checking)
M302 S170     ; only allow extrusion above 170
M302 S170 P1  ; set min extrude temp to 170 but leave disabled
```

## M303 - PID autotune

  Auto-tune the PID system to find stable values.  PIDTEMP  PIDTEMPBED

This command initiates a process of heating and cooling to determine the proper PID values for the specified hotend or the heated bed.

## Usage

M303 C<count> [E<index>] S<temp> U<flag>

Argument	Description
C<count>	Cycles. Default 5.
[E<index>]	Hotend index (-1 for heated bed). Default 0.
S<temp>	Target temperature
U<flag>	Use PID result. (Otherwise just print it out.)


## Notes

Requires PIDTEMP or PIDTEMPBED .

View current PID values with M503 .

If EEPROM\_SETTINGS is enabled, all PID values are saved with M500 , loaded with M501 , and reset with M502 .

## M304 - Set Bed PID

  Set PID values for the heated bed.  PIDTEMPBED

Set the values that control the PID loop for the heated bed.

# Usage

M304 [D<value>] [I<value>] [P<value>]

Argument	Description
[D<value>]	Derivative value
[I<value>]	Integral value
[P<value>]	Proportional value

## Notes



Requires PIDTEMPBED .

View current bed PID values with M503 .

If EEPROM\_SETTINGS is enabled, bed PID values are saved with M500 , loaded with M501 , and reset with M502 .

---

## M350 - Set micro-stepping

  Set micro-stepping for drivers that support it 

If your board has digital micro-stepping pins ( X\_MS1 , Y\_MS1 , etc.), use this command to set the micro-steps.



## Usage

M350 [B<1|2|4|8|16>] [E<1|2|4|8|16>] [S<1|2|4|8|16>] [X<1|2|4|8|16>] [Y<1|2|4|8|16>] [Z<1|2|4|8|16>]

Argument	Description
[B<1 2 4 8 16>]	Set micro-stepping for the 5th stepper driver.
[E<1 2 4 8 16>]	Set micro-stepping for the E0 stepper driver.
[S<1 2 4 8 16>]	Set micro-stepping for all 5 stepper drivers.
[X<1 2 4 8 16>]	Set micro-stepping for the X stepper driver.
[Y<1 2 4 8 16>]	Set micro-stepping for the Y stepper driver.
[Z<1 2 4 8 16>]	Set micro-stepping for the Z stepper driver.

---

## M351 - Set Microstep Pins

  Directly set the micro-stepping pins 

If your board has digital micro-stepping pins ( X\_MS1 , Y\_MS1 , etc.), use this command to set the micro-steps.

## Usage

M351 [B<0|1>] [E<0|1>] S<1|2> [X<0|1>] [Y<0|1>] [Z<0|1>]

Argument	Description
[B<0 1>]	Set the MS1/2 pin for the 5th stepper driver.
[E<0 1>]	Set the MS1/2 pin for the E stepper driver.

Argument	Description
S<1 2>	Select the pin to set for all specified axes. <ul style="list-style-type: none"> <li>S1 : Select pin MS1 for all axes being set.</li> <li>S2 : Select pin MS2 for all axes being set.</li> </ul>
[X<0 1>]	Set the MS1/2 pin for the X stepper driver.
[Y<0 1>]	Set the MS1/2 pin for the Y stepper driver.
[Z<0 1>]	Set the MS1/2 pin for the Z stepper driver.

## Examples

Set the X\_MS1 pin and clear the E0\_MS1 pin

```
M351 S1 X1 E0
```

## M355 - Case Light Control

 control

Turn the case light on or off, set brightness

CASE\_LIGHT\_PIN

Set the case light power state and/or brightness.

## Usage

M355 [P<byte>] [S<bool>]

Argument	Description
[P<byte>]	Set the brightness factor from 0 to 255.
[S<bool>]	Turn the case light on or off.

## Notes

Requires a CASE\_LIGHT\_PIN to control the on/off or PWM state of the case light. The pin will need to be a 12V MOSFET pin with PWM control, or a signal pin connected to a MOSFET or relay to control the higher current required for the light.

## Example

Turn the case light on at half brightness

```
M355 S1 P128
```

## M360 - SCARA Theta A

 control

Move to Theta A

MORGAN\_SCARA

Move the nozzle to SCARA calibration position Theta 0 (A0 B120) for calibration of "zero degrees."

## Usage

M360

## M361 - SCARA Theta-B

Move to Theta-B

Move the nozzle to SCARA Theta-B calibration position Theta 90 (A90 B130) for calibration of “90 degrees steps-per-degree.”

## Usage

M361

## M362 - SCARA Psi-A

Move to Psi-A

Move the nozzle to SCARA Psi-A calibration position Psi 0 (A60 B180) for calibration of “zero degrees.”

## Usage

M362

## M363 - SCARA Psi-B

Move to Psi-B

Move the nozzle to SCARA Psi-B calibration position Psi 90 (A50 B90) for calibration of “90 degrees steps-per-degree.”

## Usage

M363

## M364 - SCARA Psi-C

Move to Psi-C

Move the nozzle to SCARA Psi-C calibration position Theta-Psi 90 (A45 B135) for calibration of “90 degrees to Theta.”

## Usage

M364

## M380 - Activate Solenoid

Activate

Activate the solenoid on the active extruder.

## Usage

M380

## Example

```
M381 ; Deactivate all solenoids
T0 ; Extruder 0
M380 ; Activate the E0 solenoid
```

## M381 - Deactivate Solenoids

Deactivate all extruder solenoids

Deactivate all solenoids on all extruders.

## Usage


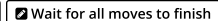


## Example

```
M381 ; Deactivate all solenoids
```

---

## M400 - Finish Moves

 planner  Wait for all moves to finish

This command causes all GCode processing to pause and wait in a loop until all moves in the planner are completed.

---

## Usage

M400

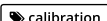
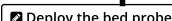

## Example

Wait for moves to finish before turning off the spindle

```
M400  
M5 ; Without M400 this happens too soon
```

---

## M401 - Deploy Probe

 calibration  Deploy the bed probe  HAS\_BED\_PROBE

Deploy the bed probe. The Z axis may raise up to make room for the probe to deploy.

---

## Usage

M401

## Notes

Requires some kind of bed probe. This command has no effect for probes that don't deploy.

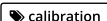
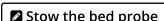

## Examples

Deploy the probe

```
M401
```

---

## M402 - Stow Probe

 calibration  Stow the bed probe  HAS\_BED\_PROBE

Stow the bed probe. The Z axis may raise up to make room for the probe to stow.

---

## Usage

M402

## Notes

Requires some kind of bed probe. This command has no effect for probes that don't deploy.

## Examples

Stow the probe

```
M402
```

## M404 - Set Filament Diameter

control  Set the nominal diameter for filament width sensor auto-flow `FILAMENT_WIDTH_SENSOR`

Report or set the nominal filament width, such as 1.75 or 3.00.

This value is used to determine the percentage difference when auto-adjusting flow in response to the measured filament width, and should match the value used for filament width in your slicer settings.

### Usage

M404 [W<linear>]

Argument	Description
[W<linear>]	The new nominal width value

### Notes

Requires `FILAMENT_WIDTH_SENSOR`.

### Examples

Set the diameter to 1.75mm

```
M404 W1.75
```

Get the current width

```
M404  
> Filament dia (nominal mm): 1.75
```

## M405 - Filament Width Sensor On

control  Enable filament width sensor flow control `FILAMENT_WIDTH_SENSOR`

Turn on the filament width sensor and start using it to do flow control. Initially, the filament between the sensor and the hot-end will be treated as the nominal width.

### Usage

M405 [D<cm>]

Argument	Description
[D<cm>]	Distance from measurement point to hot end. If not given, the previous value will be used. The default startup value is set by <code>MEASUREMENT_DELAY_CM</code> .

### Notes

Requires `FILAMENT_WIDTH_SENSOR`.

### Example

Start measuring filament width, adjusting flow

M405

---

## M406 - Filament Width Sensor Off

  Disable filament width sensor flow control  FILAMENT\_WIDTH\_SENSOR

Turn off the filament width sensor and stop using it to do flow control.

---

### Usage

M406

### Notes

Requires FILAMENT\_WIDTH\_SENSOR .


### Example

Stop measuring filament width

```
M406
```

---

## M407 - Filament Width

  Report the measured filament width  FILAMENT\_WIDTH\_SENSOR

Report the current measured filament width to the host.

---

### Usage

M407

### Notes

Requires FILAMENT\_WIDTH\_SENSOR .


### Example

Get the current measured filament width

```
M407
> Filament dia (measured mm): 1.768
```

---

## M410 - Quickstop

  Stop all steppers instantly

Stop all steppers instantly. Since there will be no deceleration, steppers are expected to be out of position after this command.

---

### Usage

M410

### Notes

This command is intended only for emergency situations.

If EMERGENCY\_PARSER is not enabled, this will be delayed.

## Example

Stop all steppers now.

```
M410
```

## M420 - Bed Leveling State

  `AUTO_BED_LEVELING_(3POINT|LINEAR|BILINEAR|UBL)|MESH_BED_LEVELING`

Get and/or set the enabled state of bed leveling compensation, plus Z fade height.

This command has been extended with `L` to load a mesh for Unified Bed Leveling.

## Usage


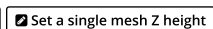
```
M420 [L<int>] [S<bool>] [V<bool>] [Z<linear>]
```

Argument	Description
[L<int>]	Load mesh from EEPROM index (Requires <code>AUTO_BED_LEVELING_UBL</code> and <code>EEPROM_SETTINGS</code> )
[S<bool>]	Enabled state
[V<bool>]	Verbose: Print the stored mesh / matrix data
[Z<linear>]	Leveling fade Z height (Requires <code>ENABLE_LEVELING_FADE_HEIGHT</code> )

## Notes

The "current position" may change in response to `M420 Sn`.

## M421 - Set Mesh Value

  `AUTO_BED_LEVELING_(BILINEAR|UBL)|MESH_BED_LEVELING`

This command is used to set a single Z value for a mesh point in the stored bed leveling data.

Allowed forms are `M421 In Jn Zn` or `M421 Xn Yn Zn`. (UBL can use `Q` in place of `Z`.)

## Usage

```
M421 [I<int>] [J<int>] [Q<linear>] [X<linear>] [Y<linear>] [Z<linear>]
```

Argument	Description
[I<int>]	X index into the mesh array
[J<int>]	Y index into the mesh array
[Q<linear>]	A value to add to the existing Z value (Requires <code>AUTO_BED_LEVELING_UBL</code> )
[X<linear>]	X position (which should be very close to a grid line)
[Y<linear>]	Y position (which should be very close to a grid line)
[Z<linear>]	The new Z value to set

## Examples

Set the Z height in the middle of a 5x5 grid




```
M421 I2 J2 Z-0.05
```

Set the same Z height using XY

```
M421 X100 Y100 Z-0.05
```

---

## M428 - Home Offsets Here

  Set home offsets based on current position 

Use `M428` to set a persistent offset to the native home position and coordinate space by assigning the current position as the native home position. See the example below.

- The current position is set to the native home position.
- Any previous position shift from `G92` is cleared.
- The home offset is persistent — added to the current position until changed.
- Some uses include fine adjustment of Z position (without moving endstops) and shifting the coordinate space to print on a different part of the bed.

---

## Usage

M428

## Notes

- Only the Z offset can be altered on `DELTA`.
- This GCode can be disabled with `NO_WORKSPACE_OFFSETS` to optimize movement.
- Changing the home offsets will not invalidate bed leveling or other saved data.
- If `EEPROM_SETTINGS` is enabled, the home offsets are saved with `M500`, loaded with `M501`, and reset with `M502`.
- Use `M206` (</docs/gcode/M206.html>) to set the home offsets directly.

---



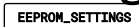
## Examples

What was `X=10` becomes `X=0`. So the X home offset becomes `-10`.

```
G1 X10  
M428
```

---

## M500 - Save Settings

  Save settings to EEPROM. 

Save all configurable settings to EEPROM.

---

## Usage

M500

## Notes

Requires `EEPROM_SETTINGS`.

Since Marlin 1.1.0 only changed bytes are written to prolong EEPROM life.


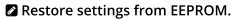
---

## Examples

Save settings

```
M500
```

# M501 - Restore Settings

  `EEPROM_SETTINGS`

Load all saved settings from EEPROM.

---

## Usage

M501

## Notes

Requires `EEPROM_SETTINGS`.



## Examples

Restore all settings.

```
M501
```

---

# M502 - Factory Reset

Reset all configurable settings to their factory defaults.

---

To also reset settings in EEPROM, follow with `M500` (</docs/gcode/M500.html>).

---

## Usage

M502

## Notes

This command can be used even if `EEPROM_SETTINGS` is disabled.


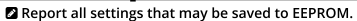
## Examples

Reset settings and save them to EEPROM

```
M502 ; reset!  
M500 ; saved!!
```

---

# M503 - Report Settings

Print a concise report of all current settings (in SRAM) to the host console.

---

## Usage

M503 [`S<f1ag>`]

Argument	Description
[ <code>S&lt;f1ag&gt;</code> ]	Detailed output flag. ( <code>true</code> if omitted.)

## Notes

Does not require `EEPROM_SETTINGS` .

## M540 - Endstops Abort SD

`sdcard`  Abort SD printing when an endstop is triggered. `SDSUPPORT` `ABORT_ON_ENDSTOP_HIT_FEATURE_ENABLED`

Set whether SD printing should abort in the event of any endstop being triggered. This provides a fast way to abort a print in the event of mechanical failure such as loose couplings, lost steps, diverted axes, binding, etc., which lead to axes being very far out of position.

### Usage

M540 S<flag>

Argument	Description
S<flag>	Whether (1) or not (0) to abort SD printing on endstop hit.

### Notes

Requires `SDSUPPORT` and `ABORT_ON_ENDSTOP_HIT_FEATURE_ENABLED` .

Use `ENDSTOPS_ALWAYS_ON_DEFAULT` or `M120` to ensure that endstops are enabled.

## M600 - Filament Change

`</> 1.1.0` `filament`  Automatically change filament `FILAMENT_CHANGE_FEATURE`

The `M600` command initiates the filament change procedure. The basic procedure will move the print head away from the print, eject the filament, wait for new filament to be inserted and the user to confirm, load and prime the filament, and continue with the print. `M600` may be initiated automatically if a filament runout sensor is installed.

### Usage

M600 [E<pos>] [L<pos>] [X<pos>] [Y<pos>] [Z<pos>]

Argument	Description
[E<pos>]	Retract before moving to change position (negative)
[L<pos>]	Load/unload length, longer for bowden (negative)
[X<pos>]	X position for filament change
[Y<pos>]	Y position for filament change
[Z<pos>]	Z relative lift for filament change position

### Notes

Requires `FILAMENT_CHANGE_FEATURE` .

The settings for this command can be found in `Configuration.h` . At this time `M600` requires an LCD controller.

### Examples

With no parameters `M600` uses the settings in `Configuration.h` .

```
M600 ; execute filament change
```

To set the change position:

M600 X10 Y15 Z5 ; Do filament change at X:10, Y:15 and Z:+5 from current

## M605 - Dual Nozzle Mode

control  Set the behavior mode for dual nozzles  DUAL\_NOZZLE\_DUPLICATION\_MODE | DUAL\_X\_CARRIAGE

This command behaves differently for DUAL\_X\_CARRIAGE vs. DUAL\_NOZZLE\_DUPLICATION\_MODE

For DUAL\_NOZZLE\_DUPLICATION\_MODE the S2 parameter enables duplication mode. Any other value disables it.

For DUAL\_X\_CARRIAGE, this command sets the Dual X mode. See the description of S below.

## Usage

M605 [R<temp>] S<0|1|2> [X<linear>]

Argument	Description
[R<temp>]	Temperature difference to apply to E1. (Requires DUAL_X_CARRIAGE )
S<0 1 2>	Select the pin to set for all specified axes. <ul style="list-style-type: none"><li>S0 : Full control mode. Both carriages are free to move, constrained by safe distance. (Requires DUAL_X_CARRIAGE )</li><li>S1 : Auto-park mode. One carriage parks while the other moves. (Requires DUAL_X_CARRIAGE )</li><li>S2 : Duplication mode. Carriages and extruders move in unison.</li></ul>
[X<linear>]	X distance between dual X carriages. (Requires DUAL_X_CARRIAGE )

## M665 - Delta Configuration

none  Set delta geometry values  DELTA

Delta machines are very fast and accurate when tuned. The first key is to make sure all your dimensions are set correctly. Even small errors in these values can lead to curved movements and failed prints. To ensure the best delta experience, use this command in conjunction with G33 to get these dimensions set perfectly before beginning to print.

## Usage

M665 [B<linear>] [H<linear>] [L<linear>] [R<linear>] [S<float>] [X<float>] [Y<float>] [Z<float>]

Argument	Description
[B<linear>]	Delta calibration radius
[H<linear>]	Delta height
[L<linear>]	Diagonal rod
[R<linear>]	Delta radius
[S<float>]	Segments per second
[X<float>]	Alpha (Tower 1) angle trim
[Y<float>]	Beta (Tower 2) angle trim
[Z<float>]	Gamma (Tower 3) angle trim






# M666 - Coming Soon

## Usage

M666

# M851 - Z Probe Offset

Set the vertical distance from the nozzle to the Z probe trigger-point.

The easiest way to get the value is to:

- Home the Z axis.
- Raise Z and deploy the probe.
- Move Z down slowly until the probe triggers.
- Take the current Z value and negate it. ( 5.2 => -5.2 )
- Set with M851 Z-5.2 and #define Z\_PROBE\_OFFSET\_FROM\_EXTRUDER -5.2 .

## Usage

M851 Z<linear>

Argument	Description
Z<linear>	Z probe Z offset

## Notes

The default (factory) value is set by Z\_PROBE\_OFFSET\_FROM\_EXTRUDER .

## Examples




Set a reasonable offset for an inductive probe

```
M851 Z-2.0
```

A probe that is triggered by the nozzle itself

```
M851 Z1.2
```

# M852 - Bed Skew Compensation

Bed Skew Compensation corrects for misalignment in the XY, XZ, and ZY axes through the use of correction factors.

## Usage

M852 [I] [J] [K] [S]

Argument	Description
[I]	Skew correction factor for XY axis.
[J]	Skew correction factor for XZ axis
[K]	Skew correction factor for YZ axis

Argument	Description
[S]	Alias for I when only XY skew correction is enabled

## Notes

Correction factors have a range of -1 to 1.

See Bed Skew Compensation section in the configuration file for more information on calculating the correction factors.

## M900 - Linear Advance Factors

control  Get and set Linear Advance K and E/D values `LIN_ADVANCE`

Get and/or set the Linear Advance K and E/D factors.

## Usage

M900 [D<linear>] [H<linear>] [K<float>] [R<float>] [W<linear>]

Argument	Description
[D<linear>]	Diameter factor for $W * H / D$ .
[H<linear>]	Height factor for $W * H / D$ .
[K<float>]	Advance K factor. Unchanged if omitted. Set this value higher for more flexible filament or a longer filament path.
[R<float>]	Set the E/D ratio directly (overriding $W * H / D$ ).
[W<linear>]	Width factor for $W * H / D$ .

## Example

Set the E/D ratio

```
M900 R1.1
```

Set the E/D ratio with W\*H/D

```
M900 W0.5 H0.2 D3.0
```

## M906 - TMC Motor Current

none  Set the motor current (in milliamps) `HAVE_TMC2130`

Set the current for a stepper motor in milliamps units.

## Usage

M906 [E<mA>] [S<bool>] [X<mA>] [Y<mA>] [Z<mA>]

Argument	Description
[E<mA>]	Current for the E0 stepper
[S<bool>]	Auto current control on/off (Requires <code>AUTOMATIC_CURRENT_CONTROL</code> )
[X<mA>]	Current for the X stepper

Argument	Description
[Y<mA>]	Current for the Y stepper
[Z<mA>]	Current for the Z stepper

## Example

Set the XYZ motor currents to 5mA

```
M906 X5 Y5 Z5
```

## M907 - Set Motor Current

control  Set motor current via digital trimpot `DIGIPOTSS_PIN|HAS_MOTOR_CURRENT_PWM|DIGIPOT_I2C|DAC_STEPPER_CURRENT`

Set digital trimpot motor current using axis codes X, Y, Z, E, plus B and S. The unit used for current depends on the type of

## Usage

M907 [B<current>] [C<current>] [D<current>] [E<current>] [S<current>] [X<current>] [Y<current>] [Z<current>]

Argument	Description
[B<current>]	Current for the E1 stepper (Requires DIGIPOTSS_PIN or DIGIPOT_I2C)
[C<current>]	Current for the E2 stepper (Requires DIGIPOT_I2C)
[D<current>]	Current for the E3 stepper (Requires DIGIPOT_I2C)
[E<current>]	Current for the E0 stepper
[S<current>]	Set this current on all steppers (Requires DIGIPOTSS_PIN or DAC_STEPPER_CURRENT)
[X<current>]	Current for the X stepper (and the Y stepper with MOTOR_CURRENT_PWM_XY)
[Y<current>]	Current for the Y stepper (Use X with MOTOR_CURRENT_PWM_XY)
[Z<current>]	Current for the Z stepper

## M908 - Set Trimpot Pins

control  Set a digital trimpot directly `DAC_STEPPER_CURRENT|DIGIPOTSS_PIN`

Set the digital trimpot current directly by address/channel/pin index. DAC\_STEPPER\_CURRENT pertains to the MCP4728.

## Usage

M908 P<address> S<current>

Argument	Description
P<address>	Pin (i.e., Address, Channel)
S<current>	Current value

## Notes

Requires DAC\_STEPPER\_CURRENT or DIGIPOTSS\_PIN .

---

## M909 - DAC Print Values

control  Report DAC current values to host  DAC\_STEPPER\_CURRENT

Print the DAC stepper current values in the format “ % (Amps) ”.

---

## Usage

M909

## Notes

Requires DAC\_STEPPER\_CURRENT .

## Examples

```
M909
```

---

## M910 - Commit DAC to EEPROM

control  Commit digipot/DAC value to external EEPROM  DAC\_STEPPER\_CURRENT

Commit digipot/DAC value to external EEPROM via I2C.

---

## Usage

M910

## Notes

Requires DAC\_STEPPER\_CURRENT .

---

## M911 - TMC OT Pre-Warn Condition

control  Driver overtemperature pre-warn condition  HAVE\_TMC2130

Report the TMC stepper driver overtemperature pre-warn condition to the host.

---

## Usage

M911

---

## M912 - Clear TMC OT Pre-Warn

control  Clear overtemperature pre-warn condition flag  HAVE\_TMC2130

Clear the stepper driver overtemperature pre-warn condition flag.

---

## Usage

M912

## Notes

Requires one or more TMC stepper drivers.

---

# M913 - Set Hybrid Threshold Speed

control  TMC driver switching to spreadCycle  HAVE\_TMC2130 or HAVE\_TMC2208  HYBRID\_THRESHOLD

When `HYBRID_THRESHOLD` is enabled, the TMC driver is switched from the quieter StealthChop to spreadCycle when the feed rate for a given stepper motor is over its `_HYBRID_THRESHOLD`.

## Individual Threshold Values

- `X_HYBRID_THRESHOLD`
- `X2_HYBRID_THRESHOLD`
- `Y_HYBRID_THRESHOLD`
- `Y2_HYBRID_THRESHOLD`
- `Z_HYBRID_THRESHOLD`
- `Z2_HYBRID_THRESHOLD`
- `E0_HYBRID_THRESHOLD`
- `E1_HYBRID_THRESHOLD`
- `E2_HYBRID_THRESHOLD`
- `E3_HYBRID_THRESHOLD`
- `E4_HYBRID_THRESHOLD`

## Usage

M913 [E] [X] [Y] [Z]

Argument	Description
[E]	Set <code>E0_HYBRID_THRESHOLD</code> and <code>E1_HYBRID_THRESHOLD</code> and <code>E2_HYBRID_THRESHOLD</code> and <code>E3_HYBRID_THRESHOLD</code> to the provided value.
[X]	Set <code>X_HYBRID_THRESHOLD</code> and <code>X2_HYBRID_THRESHOLD</code> to the provided value.
[Y]	Set <code>Y_HYBRID_THRESHOLD</code> and <code>Y2_HYBRID_THRESHOLD</code> to the provided value.
[Z]	Set <code>Z_HYBRID_THRESHOLD</code> and <code>Z2_HYBRID_THRESHOLD</code> to the provided value.

## Notes

At least one parameter must be used.

Setting `X`, `Y`, `Z`, or `E` will set the hybrid threshold for all motors that fall under that category that have the `_IS_TRINAMIC` flag set. See examples below.

## Examples

In this case, assume that `X_IS_TRINAMIC` and `X2_IS_TRINAMIC` are both enabled in `Configuration_adv.h`.

```
M913 X100
```

Both `X_HYBRID_THRESHOLD` and `X2_HYBRID_THRESHOLD` are set to 100.

In this case, assume that `X_IS_TRINAMIC`, `X2_IS_TRINAMIC`, `Y_IS_TRINAMIC`, `E0_IS_TRINAMIC`, and `E2_IS_TRINAMIC` are enabled in `Configuration_adv.h`.

```
M913 X100 Y120 E30
```

`X_HYBRID_THRESHOLD` and `X2_HYBRID_THRESHOLD` are set to 100. `Y_HYBRID_THRESHOLD` is set to 120. `E0_HYBRID_THRESHOLD` and `E2_HYBRID_THRESHOLD` are both set to 30. `E1_HYBRID_THRESHOLD` is not changed since the `E0_IS_TRINAMIC` flag is not set.

# M914 - TMC Bump Sensitivity

control  Set sensorless homing sensitivity  HAVE\_TMC2130  SENSORLESS\_HOMING

Some TMC stepper drivers have the ability to detect when they bump into something that causes them to stop moving. This feature is so sensitive that it can actually take the place of traditional endstops. Set the bump sensitivity for the X and Y stepper drivers here.

# Usage

M914 [X<int>] [Y<int>]

Argument	Description
[X<int>]	Sensitivity of the X stepper driver
[Y<int>]	Sensitivity of the Y stepper driver

# Notes

Requires `SENSORLESS_HOMING` in `Configuration_adv.h`.

This feature is currently only compatible with the TMC2130.

It is advised to set both `X_HOME_BUMP_MM` and `Y_HOME_BUMP_MM` to 0.

Higher values = Lower sensitivity.

---

# M928 - Start SD Logging

`sdcard`  Log serial input to an SD file

Use this command to start logging all console and host input to an SD file while still operating the machine.

# Usage

M928

# Notes

Stop logging with `M29`.

# Examples

Start logging to `log.txt`

```
M928 log.txt
```

---

# M999 - STOP Restart

`control`  Return the machine to Running state

If a `STOP` occurs you can use `M999` to restart the “stopped” machine after resolving the issue.

Marlin will call `STOP` if any error occurs that would make continuing the current process problematic. For example, if the probe fails to deploy, it will abort probing and `STOP`. Note that this disables all heaters.

# Usage

M999 S<bool>

Argument	Description
S<bool>	Resume without flushing the command buffer. The default behaviour is to flush the serial buffer and request a resend to the host starting on the last N line received.

# Examples

Restart the machine

Brought to you with ❤️ lack of 🛠️ and lots of ☹️.

The contents of this website are © 2018 under the terms of the GPLv3 (<http://www.gnu.org/licenses/gpl-3.0.txt>) License.

[Back to top ↕](#)

