

G-Code 详解

简介：玩过一段时间 3D 打印机的朋友，都会接触到 G-code 文件。所谓 G-code 文件，指的是 3D 模型在进入 3D 打印机实际打印之前，必须要经过切片器处理而成的一种中间格式文件。

玩过一段时间 3D 打印机的朋友，都会接触到 G-code 文件。所谓 G-code 文件，指的是 3D 模型在进入 3D 打印机实际打印之前，必须要经过切片器处理而成的一种中间格式文件。这种中间格式文件的内容，实际上每一行都是 3D 打印机固件所能理解的命令。而这些命令，也被称为 G-code 命令，是 3D 打印机和电脑之间最重要的命令交互界面。

既然所有 3D 打印机都使用 G-code 作为对外联系的唯一信息交互方式，那么这种“语言”的标准就很重要了。不幸的是，虽然所有的 3D 打印机，都使用 G-code 作为与计算机的交互语言，但实际上每种 3D 打印机“说出”的 G-code，都多多少少有些不同。这也可以说，每种 3D 打印机都有自己的“方言”。我们要研究 G-code，就要从一种最常见的“方言”，也可以说是“普通话”开始，先了解一种，然后再学习其他类似的语言，相互对比，就很容易了。

基础运动

既然 G-code 是计算机指挥 3D 打印机干活用的一套语言，那么其中最重要的就是运动类的指令。

G0/G1 直线移动

虽然从名字上看，G0 叫做“快速直线移动”，而 G1 叫做“直线移动”，但实际上在 Repetier-firmware 里面，G0 和 G1 指令是完全等价的，没有任何区别。移动是否快速，完全是靠参数 F 来决定的（下面会详细介绍）。这条指令的作用也很简单，就是将挤出头线性移动到一个特定的位置。这条指令带有不少参数，完整的形式是这样的：

```
G0 Xnnn Ynnn Znnn Ennn Fnnn Snnn
```

或者

```
G1 Xnnn Ynnn Znnn Ennn Fnnn Snnn
```

使用时，不需要所有的参数全部存在，但至少要有有一个参数。其中，

- Xnnn 表示 X 轴的移动位置；
- Ynnn 表示 Y 轴的移动位置；
- Znnn 表示 Z 轴的移动位置；

- Ennn 表示 E 轴（挤出头步进电机）的移动位置；
- Fnnn 表示速度，单位是毫米/每分钟；
- Snnn 表示是否检查限位开关，S0 不检查，S1 检查，缺省值是 S0；

举例来说，

```
G1 F1500  
G1 X50 Y25.3 E22.4
```

这样两行 G-code，表示了首先将速度设置为 1500mm/min，也就是 25mm/s，然后将挤出头移动至 x=50mm, y=25.3mm 的位置上，z 轴高度不变，并且将挤出头步进电机移动至 22.4mm 的位置上。这里，速度、xyz 位置都比较好理解，但挤出头步进电机的位置怎么理解呢？移动至 22.4mm 处，代表着挤出了多少耗材呢？实际上，这里挤出头的具体动作，要根据之前挤出头步进电机所在的状态（也就是位置）而定。比如在这两条语句之前，挤出头步进电机已经处于 20mm 的位置处，那么这里挤出头步进电机只要再前进 2.4mm 就可以了。

仔细想想，其实挤出头步进电机的工作方式，与 xyz 轴完全一样。xyz 轴之所以比较好理解，是因为我们清楚的知道原点(0, 0, 0)的位置在哪里。而对于 e 轴来说原点的位置也会在打印开始处被初始化到 0 的位置。知道了原点的位置，就可以正确理解挤出头步进电机的工作方式了。实际上，挤出头步进电机仍然是以原点为基础，只不过是在整个打印过程中持续增加的。（思考：切片器的挤出头回抽动作，对应了什么样的 G-code 代码？）

再举一例，

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

这个例子与上面的例子非常类似，唯一的区别，就是 F 参数了。而这两条语句的意义，除了对 XYZE 的移动之外，还会将打印速度，从语句执行开始时的 1500mm/min，提高到语句执行结束时的 3000mm/min。这里有两点需要注意。

第一点，F 参数与 XYZE 参数一样，在语句执行的过程中线性插值；

第二点，在预先知道第一点的前提下，F 参数使得计算机对 3D 打印机的控制更加深入和精准了。计算得当的情况下，切片器可以精确控制 3D 打印机的加速和减速过程，使得整个 3D 打印过程更加顺滑。

G2/G3 圆弧移动

这两条命令中，G2 是顺时针圆弧移动，G3 是逆时针圆弧移动。命令的完整形式是：

```
G2 Xnnn Ynnn Innn Jnnn Rnnn Ennn Fnnn
```

或者

```
G3 Xnnn Ynnn Innn Jnnn Rnnn Ennn Fnnn
```

其中，

- Xnnn 表示移动目标点的 X 坐标；
- Ynnn 表示移动目标点的 Y 坐标；
- Innn 表示圆心位置，值是圆心距离当前位置的 X 分量；
- Jnnn 表示圆心位置，值是圆心距离当前位置的 Y 分量；
- Rnnn 表示圆形的半径长度；
- Ennn 表示 E 轴（挤出头步进电机）的移动位置；
- Fnnn 表示速度，单位是毫米/每分钟；

根据勾股定理， $R^2 = I^2 + J^2$ 。因此，如果提供了圆心位置参数，就不需要提供半径参数了。反之，如果提供了半径参数，也可以根据当前点和目标点计算出圆心位置，就不需要提供 I/J 参数了。其他几个的参数用法，与 G0/G1 是完全一样的。

G2/G3 命令面临的最尴尬的问题，是常用的上位机切片器软件，包括 Slic3r 以及 Cura engine，并不会生成这两条指令。所有 3D 模型中的圆弧，在 STL 文件中已经被转化为使用大量小线段拟合而成的曲线。这样，切片器自然也不会把这些小线段当做圆弧处理。最终的 G-code 输出结果，也只会存在 G0/G1 指令，而不会存在 G2/G3 指令。当然，据打印虎所知，如果你使用的是比较小众的上位机软件，比如 artCAM 等，因为这些软件的输入并不是 STL 文件，因此它们的输出 G-code 是很有可能出现 G2/G3 命令的。

如果确定了你的 3D 打印机只会接收到 G0/G1 直线移动命令，那么我们完全可以在 3D 打印机固件配置中，定义

```
#define ARC_SUPPORT 0
```

这样，所有与 G2/G3 指令相关的代码，就都不会编译，也不会包含在最终的固件代码中了。可以节省一些固件的空间，同时并不会影响任何 3D 打印机的功能。

还有一个问题留给大家思考：在上位机切片软件输出 G2/G3 命令的情况下，相比于上位机切片软件输出 G0/G1 命令的情况，是否 3D 打印机打印圆形物体时会更圆呢？（答案是不会。）那么其中的原理是什么？

G4 暂停移动

这条命令让挤出机在当前位置停止一段时间。可能的参数包括：

- Pnnn 表示停止移动的时间，以毫秒为单位，1000 毫秒等于 1 秒。
- Snnn 也表示停止移动的时间，以秒为单位。

因此，G4 P2000 命令与 G4 S2 命令是完全等价的。

G10/G11 回抽/反回抽

这两条命令使挤出头执行一个回抽(G10)或者相反的动作(G11)。所谓回抽，就是让 E 轴步进电机反转一小段。而反回抽则让 E 轴步进电机正转一小段。参数只有一个：

- Snnn 表示回抽的距离。
- S1 表示长回抽，S0 表示短回抽。

实际上，目前的切片器并不太依赖于 G10/G11 指令执行回抽动作，而是利用 G1 Ennn 命令直接命令挤出头步进电机前进或倒退到某一个位置。因此，与 G2/G3 命令类似，G10/G11 命令基本上是个摆设，除非未来有专门的切片器可以生成这两条指令，否则完全可以将这两条指令关闭，节省内存空间。在固件配置中，定义

```
#define FEATURE_RETRACTION 0
```

可以关闭 G10/G11 功能，在编译期去除这段相关的代码。

G20/G21 设置距离单位

这两条命令非常简单，用于设置当前距离单位为英寸(G20)或者毫米(G21)。没有参数。

未设置时缺省值是毫米。

G28 归零

这条命令使 3D 打印机 XYZ 轴以及挤出头 E 轴归零。参数包括：

- X 表示使 X 轴归零
- Y 表示使 Y 轴归零
- Z 表示使 Z 轴归零

E 表示重置 E 轴的位置为 0，与 XYZ 轴不同的是，如果使用了 E 参数，E 轴步进电机并不运动，而是将当前的 E 轴位置直接设置为 0，这样下面对 E 轴的运动指令，都会解释为相对 0 点的运动。

如果使用时没有任何参数，直接使用 G28，等价于 G28 XYZ 命令。这时并不会对 E 轴进行重置为 0 的操作。

XYZ 轴归零的顺序，由固件配置 HOMEING_ORDER 决定，比如定义为

```
#define HOMING_ORDER HOME_ORDER_XYZ
```

就代表着先归零 X 轴，然后是 Y 轴，最后是 Z 轴。

T 设置当前挤出头

对于拥有多个挤出头的 3D 打印机来说，需要使用 T 命令选择当前工作的挤出头。这条命令有一个无名参数，参数值直接跟在 T 后面。例如：

- T0 表示选择第一个挤出头；
- T1 表示选择第二个挤出头；

参数是 T 命令最特殊的一点。这与其他所有的 G-code 命令都不相同。

Z 轴高度测试与自动调平

三角洲类型的 3D 打印机，由于其打印速度更快，受到很多 3D 打印用户的欢迎。与 XYZ 式 3D 打印机最大的一个不同，在于三角洲类型 3D 打印机的运动计算更加复杂，很难依赖人工调平达到较好的打印效果。因此，对 Z 轴的自动高度测试，以及自动调平相关的功能，就显得更加重要了。以下 G-code 命令，是 Repetier-firmware 对这方面进行支持的一组命令。当然，这些功能并不仅限于三角洲类型的 3D 打印机。如果是包含了 Z 轴高度测试微动开关的 XYZ 式 3D 打印机，也同样可以使用这些功能。

G29 Z 轴高度三点测试

这条命令测试打印平面上三个点的 Z 轴高度，并在串口上输出结果。参数包括：

Snnn 测试结果的处理方式。S1 表示更新内存中的 Z 轴高度值（重置系统会丢失），S2 表示更新内存以及 EEPROM 中的 Z 轴高度值（重置系统不会丢失）。

无参数时，G29 命令表示只从串口上输出结果，不更新内存或 EEPROM 中的 Z 轴高度值。

一般来说，只有使用高位限位开关（也就是说，Z 轴的限位开关位于 Z 轴坐标最大处），且在挤出头上附带有 Z 轴高度测试微动开关的机型，适合使用 G29 命令测试 Z 轴高度。其他机械配置的机型，不适合使用 G29 命令。G29 命令由固件配置

```
#define FEATURE_Z_PROBE 1
```

决定是否开启。如果这个配置项定义为 0，则编译时会去除对 G29 命令的支持，节省内存的使用。

命令执行时，打印平面上的三个点，其 XY 坐标由以下固件配置参数决定：

```
#define Z_PROBE_X1 -52
#define Z_PROBE_Y1 -30
#define Z_PROBE_X2 52
#define Z_PROBE_Y2 -30
#define Z_PROBE_X3 0
```

```
#define Z_PROBE_Y3 60
```

命令执行的开始和结束，分别会执行一段预定义的 G-code。缺省的固件配置定义为：

```
#define Z_PROBE_START_SCRIPT "G28"
```

```
#define Z_PROBE_FINISHED_SCRIPT ""
```

可以看出，在缺省状态下，开始执行 G29 时，系统会自动对挤出头进行复位（G28 命令）。结束执行 G29 时，没有特殊的动作。

G29 命令的 Z 轴高度测试，通常由一个微动开关控制触发。这个开关的端口号，由 Z_PROBE_PIN 单独指定。

G29 命令的输出，格式为：

```
X:0.00 Y:0.00 Z:200.00 E:0.00
```

```
Z-probe:5.01 X:-52.00 Y:-30.00
```

```
Z-probe:13.04 X:52.00 Y:-30.00
```

```
Z-probe:12.77 X:0.00 Y:60.00
```

```
X:0.00 Y:60.00 Z:-98.48 E:0.00
```

从以上例子的输出可以看出，G29 命令一共测试了三个坐标点，分别在(-52, -30), (52, -30) 以及(0, 60)的位置，形成一个正三角形。三个点的 Z 轴高度相差比较悬殊，在第一个点正好是 5mm 的情况下，后两个点分别是 13.04mm 以及 12.77mm。第一行和最后一行，是测试开始时以及测试结束时的挤出头坐标位置。

G30 Z 轴高度单点测试（单步）

这条命令作为一个完整 Z 轴高度测试过程的一步，测试打印平面上一个点的 Z 轴高度，并在串口上输出结果。这个完整的 Z 轴高度测试过程，通常是由 3D 打印机控制软件连续发出的，通过参数控制 G30 的执行状态。因此在手动工作模式下，G30 命令只适合不带参数运行（等价于 G30 P3，见下面的参数说明）。

G30 命令的参数包括：

Pnnn 表示测试的状态，P1 表示当前这一步是整个 Z 轴高度测试过程的第一步；P2 表示当前这一步是整个 Z 轴高度测试过程的最后一步；P3 表示当前这一步是 Z 轴高度测试过程的唯一一步，也就是说既是第一步也是最后一步；P0 表示当前这一步是 Z 轴高度测试过程中的中间一步。无参数情况下，P 的缺省值是 3。

与 G29 命令类似，G30 命令同样由固件配置

```
#define FEATURE_Z_PROBE 1
```

决定是否开启。

G30 命令的输出，与上面的 G29 输出格式一致，但只有其中的一行，需要上位机软件多次发出 G30 命令，再综合处理所有的输出结果。

G31 输出 Z 轴高度测试微动开关状态

这条命令非常简单，没有参数。执行后会输出当前 Z 轴高度测试微动开关的当前状态：

```
Z-probe state:L
```

其中 L 表示微动开关没有触发。如果是处于触发状态，这里会输出 H。

G29 命令、G30 命令、G31 命令只进行 Z 轴的高度测试，并不进行自动调平。有些上位机 3D 打印机控制软件，会通过这一组命令配合自动跳屏算法实现（上位机）热床自动调平功能。如果希望不通过上位机，只由 3D 打印机自身完成自动调平功能，需要使用 G32 命令。

G32 热床自动调平

这条命令在 G29 命令的基础上，不仅测试打印平面上三个点的 Z 轴高度，而且还会根据测试的结果，对 3D 打印机的机械参数进行调整，实现热床自动调平。G32 命令使用的参数与 G29 命令是一致的：

Snnn 测试结果的处理方式。S1 表示更新内存中的相关参数值（重置系统会丢失），S2 表示更新内存以及 EEPROM 中的相关参数值（重置系统不会丢失）。

G32 命令执行完成时，不仅 Z 轴高度参数发生了改变，而且还会根据 3D 打印机的硬件配置，对热床进行相应的调平处理。

如果热床本身是使用步进电机进行高度控制的，那么程序会自动调整步进电机的位置，使热床自动调整为平整的状态；如果热床本身不能移动（这个应该是更常见的情况），那么 G32 命令会在 3D 打印机内存中构建一个转换矩阵（Transformation matrix），让未来 3D 打印机所处理的所有三维空间位置，都先经过这个矩阵的变换，保证在 Z=0 的情况下，正好与热床平面完全吻合。由于这里涉及到高深的计算机图形学知识，我们就不详细介绍了。有修改这方面代码需求的朋友，可以直接与打印虎进行联系。

G32 命令，由固件配置

```
#define FEATURE_AUTOLEVEL 1
```

决定是否开启。

G32 命令的输出，格式与 G29 命令类似：

```
X:0.00 Y:0.00 Z:200.00 E:0.00

Z-probe:5.00 X:-52.00 Y:-30.00

Z-probe:12.97 X:52.00 Y:-30.00

Z-probe:12.76 X:0.00 Y:60.00

Info: 0.99709 -0.00319 -0.07628 0.00000 0.99912 -0.4191 0.07634 0.004179
0.99621

Info: Autoleveling enabled

X:7.32 Y:64.08 Z:-95.66 E:0.00

X:0.00 Y:0.00 Z:200.00 E:0.00
```

除了与 G29 命令相似的测量信息之外，G32 命令还输出了计算得到的自动调平矩阵，并且打开了自动调平功能。需要注意的一点是，G32 命令虽然生成了自动调平矩阵，但并没将其保存在 EEPROM 中，因此下次开机这个信息将会丢失。可以配合 M320 S1 命令，将自动调平矩阵保存在 EEPROM 中。

M251 将当前 Z 轴位置保存为 Z 轴高度值

这条命令可以将当前的 Z 轴位置保存为 Z 轴高度值，以使前面的 Z 轴高度手动/自动测量的结果起作用。通常，M251 命令只工作在三角洲机型上，并且应该与 G29 命令联合使用（自动测量 Z 轴高度）。这条命令没有相关的参数。

当 3D 打印机打开 EEPROM 支持时，这条命令还会将 Z 轴高度值同时保存在 EEPROM 中。

只有当固件配置定义

```
#define Z_HOME_DIR -1
```

也就是 Z 轴向正方向归位，并且定义

```
#define MAX_HARDWARE_ENDSTOP_Z true
```

也就是存在硬件的 Z 轴高位限位开关时，M251 命令才会在编译中包含相关的代码。

通常，只有三角洲类型的 3D 打印机才能满足这两个条件限制。

M320/M321 开启/关闭自动调平

开启(M320)或者关闭(M321)自动调平功能，使自动调平转换矩阵起作用或不起作用。命令参数为

Snnn 表示是否保存于 EEPROM，没有 S 参数或者 S0 表示不保存于 EEPROM，S1 表示保存于 EEPROM，在关闭自动调平(M321)命令中 S3 表示将自动调平矩阵清零且保存于 EEPROM 中；

M320 的输出结果为：

```
Info:Autoleveling enabled
```

表示自动调平已经打开。

M321 的输出结果为：

```
Info:Autoleveling disabled
```

表示自动调平已经关闭。

M322 清零自动调平转换矩阵

清零(M322)自动调平转换矩阵。显然，清零这个动作的同时自动调平功能也关闭了。命令参数为

Snnn 表示是否保存于 EEPROM，S0 表示不保存于 EEPROM，S1 表示保存于 EEPROM；

也就是说，

```
M321 S3
```

命令等价于

```
M322 S1
```

命令，两者都是清零自动调平矩阵，关闭自动调平功能，并且将这个设置保存于 EEPROM 之中。

以上三条命令，与 G32 命令相同，由固件配置

```
#define FEATURE_AUTOLEVEL 1
```

决定是否开启。

M322 的输出结果为：

Info:Autolevel `matrix reset`

表示自动调平转换矩阵已经被清零。

第三节，坐标模式与坐标位置

G90/G91 设置坐标模式

这两条命令用于设置当前坐标模式为绝对坐标模式(G90)或者相对坐标模式(G91)。没有参数。

未设置时缺省值是绝对坐标模式。我们在这篇教程中，所有的例子也都是以绝对坐标模式给出的。

在相对坐标模式下，每次步进电机 XYZE 移动之后，当前位置都会重置为 0。对于以下两条 G-code 命令

```
G0 X1
```

```
G0 X-1
```

如果 3D 打印机当前处于相对坐标模式下，那么 X 轴步进电机先向正方向移动一个单位，再向反方向移动一个单位。第二条语句，实际移动距离是 1 个单位（向 X 轴反方向）。

而如果 3D 打印机当前处于绝对坐标模式下，那么 X 轴步进电机先移动到 X=1 的位置处，再移动到 X=-1 的位置处。第二条语句，实际移动距离是 2 个单位（向 X 轴反方向）。

G92 设置位置

设置 3D 打印机内存中 XYZE 的位置值。不移动对应的步进电机。参数包括：

- Xnnn 表示 X 轴的位置值；
- Ynnn 表示 Y 轴的位置值；
- Znnn 表示 Z 轴的位置值；
- Ennn 表示 E 轴（挤出机步进电机）的位置值；

辅助步进电机

一些 3D 打印机的机械设计，会在 XYZE 四个步进电机轴之外，使用更多的辅助步进电机。Repetier-firmware 提供了一套辅助步进电机指令，让用户（以及上位机软件）可以操作这

些辅助步进电机。由于辅助步进电机的用途、参数各异，为了让这套指令更加通用，这些指令被设计为非常简单的形式。

G201 移动步进电机位置

将步进电机 P 的位置移动到 X 位置处。参数包括：

- Pnnn 表示第 P 个辅助步进电机；
- Xnnn 表示这个步进电机的目标位置；

这条命令与 G1 命令非常类似。

G202 设置当前位置

将 X 位置设置为步进电机 P 的当前位置。不实际移动步进电机。参数包括：

- Pnnn 表示第 P 个辅助步进电机；
- Xnnn 表示这个步进电机的当前位置；

这条命令与 G92 命令非常类似。

G203 报告当前位置

报告步进电机 P 的当前位置。参数包括：

- Pnnn 表示第 P 个辅助步进电机；

这条命令与 M114 命令非常类似。

G203 开启/关闭步进电机

用于开启/关闭步进电机 P。参数包括：

- Pnnn 表示第 P 个辅助步进电机；
- Snnn 表示开闭标志，S0 表示关闭步进电机，S1 表示开启步进电机；

步进电机开启后，有两种可能的状态。一种是“运动”状态，也就是正在进行正向或反向的旋转。另一种是“保持位置”状态，也就是保持当前的位置不变。虽然步进电机关闭也不会主动移动位置，但“保持位置”状态与步进电机关闭状态仍有显著的区别。“保持位置”状态下，当步进电机受力时，会产生一个反向的力矩，使步进电机位置保持不变。

这条命令与“节能管理”一节中的 M84 命令有关。M84 命令用于关闭 XYZE 步进电机，但不能打开这些步进电机。

SD 卡管理

M20 列目录

显示 SD 卡所有目录内容。没有相关的参数。

M20 命令的输出，格式为：

```
Begin file list  
  
a.gcode  
  
TEST/  
  
TEST/c.gcode  
  
b.gcode  
  
End file list
```

这个目录内容清单，说明了这张 SD 卡上目前有 3 个文件，分别是 a.gcode，b.gcode 以及 c.gcode，其中，a.gcode 和 b.gcode 都保存于根目录下，而 c.gcode 保存于一个名称为 TEST 的文件夹里面。

M21 加载 SD 卡

尝试加载 SD 卡，也就是执行 Mount 动作。没有相关的参数。

M22 卸载 SD 卡

卸载 SD 卡，也就是执行 Unmount 动作。没有相关的参数。

M23 选择文件

选择一个 SD 卡上的文件。参数为

- filename 表示被选择的文件名（包含目录名，以/分隔）；文件选择之后，可以执行打印、删除等动作。例如命令

M23 TEST/c.gcode

选定了 SD 卡 TEST 文件夹里面的 c.gcode 文件作为当前文件。

同时输出格式为：

```
File opened:c.gcode Size:1127565
```

```
File selected
```

表示文件已经顺利打开。

M24 开始 SD 卡打印

打印当前选定的 SD 卡文件。逐行读入 SD 卡文件内容 G-code 代码，并执行。没有相关参数。

M25 暂停 SD 卡打印

暂停当前的 SD 卡打印。没有相关参数。

M26 设置当前文件当前位置

设置当前文件的当前位置。参数为

- Snnn 表示当前位置的字节数。

M27 获取 SD 卡打印进度

获取 SD 卡打印进度。没有相关参数。

M27 命令的输出，格式为：

```
SD printing byte 11518/1127578
```

这条命令供上位机获取当前的 3D 打印进度信息，用于显示在电脑界面上。

M28 写 SD 卡文件

写一个 SD 卡文件。参数为：

- filename 表示待写入的文件名（包含目录名，以/分隔）；

从执行 M28 命令开始，所有 3D 打印机接收到的 G-code，除了 M29 命令以外，都会保存至指定的 SD 卡文件中，而不会被实际执行。这条命令可以将一个 G-code 文件从上位机 3D 打印控制软件复制到 3D 打印机的 SD 卡上，以供未来执行。

M29 结束写 SD 卡文件

结束以 M28 开始的“保存至 SD 卡文件”状态，将 3D 打印机恢复到正常状态。从此，所有接收到的 G-code 命令，都会被直接解释执行。没有相关参数。

M30 删除文件

删除一个 SD 卡中的文件。参数为：

- filename 表示待删除的文件名（包含目录名，以/分隔）；

M32 创建子目录

在 SD 卡上创建一个子目录。参数为：

filename 表示待创建的子目录（包含目录名，以/分隔）；

以上所有 SD 卡相关指令，都由固件配置

define SDSUPPORT 1

决定是否开启。如果固件不需要支持 SD 卡，关闭这项固件配置，可以节省不少内存空间。

节能管理

M84 设置步进电机自动关闭时间

当 3D 打印机一段时间没有接收到步进电机运动指令之后，3D 打印机（为了节能）会自动关闭步进电机。使用 M84 指令，可以设置这个自动关闭步进电机的时间。参数包括：

- Snnn 表示步进电机关闭的时间，以秒为单位。

如果使用 M84 时没有指定 S 参数，则步进电机立即关闭。

M84 命令的缺省值是 360 秒。在固件配置中，缺省值由

```
#define STEPPER_INACTIVE_TIME 360L
```

控制。

M85 设置 3D 打印机自动关闭时间

当 3D 打印机一段时间没有接收到指令之后，3D 打印机（为了节能）会自动关闭步进电机以及挤出头、热床等设备。使用 M85 指令，可以设置这个自动关闭 3D 打印机的时间。参数包括：

- Snnn 表示在关闭步进电机之前步进电机没有活动的时间，以秒为单位。

如果使用 M85 时没有指定 S 参数，或者使用了 S0 参数，则代表取消 3D 打印机自动关闭功能，挤出头、热床等在工作完成之后，一直会处于当前状态，而不会被自动关闭。

M85 命令的缺省值是 0（不自动关闭）。在固件配置中，缺省值由

```
#define MAX_INACTIVE_TIME 0L
```

控制。

温度管理

M104 设置挤出头目标温度

设置挤出头的目标温度。执行这条命令后，不需要等待达到这个温度，立即开始执行下一条 G-code 语句。相关参数包括：

- Snnn 表示目标温度；
- Tnnn 表示对应的挤出头；
- P 表示要等待前面的指令完成之后，再开始设置挤出头温度；
- Fnnn 表示到达目标温度之后，是否触发蜂鸣器。F1 表示要触发；

如果执行命令时没有带 T 参数，则针对当前挤出头设置目标温度。

M140 设置热床目标温度

设置热床的目标温度。执行这条命令后，不需要等待达到这个温度，立即开始执行下一条 G-code 语句。相关参数包括：

- Snnn 表示目标温度；
- Fnnn 表示到达目标温度之后，是否触发蜂鸣器。F1 表示要触发；

M105 获取当前的温度

获取当前温度值，包括挤出头和热床的温度。相关参数包括：

X 表示输出 AD 转换输入的原始值；

M105 命令的输出，格式为：

```
T:18.97 /0 B:18.75 /0 B@:0 @:0
```

可以看到，T:之后的部分，代表挤出头的当前温度/目标温度；B:之后的部分代表热床的当前温度/目标温度。

在 PID 温度控制模式下，B@:后面的数字代表热床当前的输出强度，是一个 0²⁵⁵ 的值，@:后面的数字，代表挤出头当前的输出强度，也是一个 0²⁵⁵ 的值。例子中，挤出头、热床都处于关闭状态，所以这个位置的值都是 0。

M109 等待挤出头加热达到目标温度

设置挤出头的目标温度，并等待达到这个温度。相关参数包括：

- Snnn 表示目标温度；
- Tnnn 表示对应的挤出头；
- Fnnn 表示到达目标温度之后，是否触发蜂鸣器。F1 表示要触发；

如果执行命令时没有带 T 参数，则针对当前挤出头设置目标温度。

M190 等待热床加热达到目标温度

设置热床的目标温度，并等待达到这个温度。相关参数包括：

- Snnn 表示目标温度；
- Fnnn 表示到达目标温度之后，是否触发蜂鸣器。F1 表示要触发；

M116 等待温度达到目标温度

等待所有挤出头/热床到达由之前的 M104/M140 指令所指定的目标温度。没有相关参数。

其他常用指令

M92 设置分辨率

设置 3D 打印机内存中 XYZE 步进电机的分辨率。参数包括：

- Xnnn 表示 X 轴的分辨率；
- Ynnn 表示 Y 轴的分辨率；
- Znnn 表示 Z 轴的分辨率；
- Ennn 表示 E 轴（挤出机步进电机）的分辨率；

M106/M107 打开/关闭风扇

这两条命令用于打开(M106)或关闭(M107)风扇。相关的参数包括：

- Snnn 表示打开风扇时风扇的转速，取值范围在 0~255 之间；
- P 表示要等待前面的指令完成之后，再开始调整风扇转速；

在固件配置中，定义

```
#define FEATURE_FAN_CONTROL 1
```

表示支持风扇控制功能，在编译中会包含相关的代码。

M114 输出当前位置

输出挤出头当前位置。没有相关的参数。

M114 命令的输出，格式为：

```
X:20.00 Y:30.00 Z:10.000 E:0.0000
```

M115 输出 3D 打印机信息

输出 3D 打印机信息。没有相关的参数。

M115 命令的输出，格式为：

```
FIRMWARE_NAME:Repetier_0.92.3 FIRMWARE_URL:...  
  
Printed filament:0.00m Printing time:0 days 0 hours 0 min  
  
SpeedMultiply:100  
  
FlowMultiply:100
```

第一行是固件的版本信息，很长，我没有列完整。第二行是已经打印了多少米耗材，打印时间是几天几小时几分钟。第三行是速度系数，参考 M220 命令。第四行是流率系数，参考 M221 命令。

M119 输出限位开关状态

将当前限位开关状态输出。没有相关的参数。

M119 命令的输出，格式为：

```
endstops hit: x_min:L y_min:L z_min:L
```

列出了 XYZ 三个轴的低位限位开关的当前状态。L 代表限位开关没有触发。H 代表限位开关被触发了。

M201/M202 设置最大加速度

这两条命令设置打印加速度。包括挤出头工作时（打印中）的运动加速度（M201），以及挤出头不工作时（移动中）的运动加速度（M202）。参数为

- Xnnn 表示 X 轴的加速度；
- Ynnn 表示 Y 轴的加速度；
- Znnn 表示 Z 轴的加速度；

- Ennn 表示 E 轴的加速度；

在固件配置中，定义

```
#define RAMP_ACCELERATION 1
```

表示支持加速度功能，在编译中会包含相关的代码。

M203 监控温度

使用串口输出监控 3D 打印机的温度。参数为

- Snnn 表示是否监控，S0 关闭监控，S1 打开监控；

当监控处于打开状态，可以从串口定时获取当前的温度信息。

监控输出格式与 M105 命令的输出结果完全一致。

M204 设置 PID 参数

设置挤出头温度控制的 PID 参数，命令参数为

- Snnn 表示对应的挤出头，无 S 参数表示使用当前挤出头；
- Xnnn 表示 P 参数；
- Ynnn 表示 I 参数；
- Znnn 表示 D 参数；

M207 修改抖动(Jerk)值

修改当前的最大抖动值。命令参数为

- Xnnn 表示 XY 轴的最大抖动值；
- Znnn 表示 Z 轴的最大抖动值；
- Ennn 表示 E 轴的最大抖动值；

XY 轴抖动指的是 3D 打印机同时在 X 轴和 Y 轴上移动时，产生的和速度最大值。比如，3D 打印机加热头正在向 X 轴正方向全速移动，下一条指令变为向 Y 轴正方向移动。如果同时在 X 轴和 Y 轴上改变速度，那么实际产生的速度是 X 方向的速度和 Y 方向的速度的向量和，这个比较大的速度变化值，会对 3D 打印机的机械部件产生不利的影响，而且会造成比较大的噪音。这里的设置，就限制了这个 XY 轴上和速度的最大值。当然这个值也不能设置的太小，太小的话，首先打印速度会变得很慢，而且打印会产生更多的瑕疵。

Z 轴抖动与 XY 轴抖动意义类似，不同点是 Z-Jerk 是 Z 轴方向不为 0 的抖动速度值。因为这项涉及到 Z 轴的运动，因此最大速度就低多了。

M207 命令的输出，格式为：

```
Jerk:20.00 ZJerk:0.30
```

这个输出意义很简单，表示 XY 轴抖动速度为 20mm/s，Z 轴抖动速度为 0.3mm/s。

M220 设置速度

设置 3D 打印机运行速度系数。命令参数为

- Snnn 表示系数，是一个百分数，如果 S 参数不存在，则使用缺省值 100；

3D 打印机运行速度系数，是一个在 25%到 500%范围内变化的值。这个系数值在 3D 打印机运行过程中，与切片器给出的 3D 打印机运动速度基础值相乘，得到最终的 3D 打印机实际运动速度值。

M220 命令的输出，格式为：

```
SpeedMultiply:100
```

M221 设置流率

设置 3D 打印机的流率系数（Flow rate）。命令参数为

- Snnn 表示系数，是一个百分数，如果 S 参数不存在，则使用缺省值 100；

3D 打印机流率系数，是在上位机切片软件通过耗材直径、喷头直径、层高以及 3D 打印速度等因素综合计算得到的 E 轴运动速度的基础上，叠加的一个 E 轴运动速度系数。简单地说，就是控制挤出头耗材挤出量的多少。这个系数可以在 25%到 500%范围内变化。

M221 命令的输出，格式为：

```
FlowMultiply:100
```

M302 设置是否允许冷挤出

为了保护 3D 打印机的挤出头，通常设置下，E 轴的运动必须在挤出头加热到一定温度之后才被允许。在挤出头冷却的情况下，所有的 E 轴运动命令是被 3D 打印机固件忽略的。但有些情况下我们需要在挤出头冷却的情况下运动 E 轴，这时可以通过 M302 命令进行设置。命令参数为

- Snnn 表示是否允许冷挤出，S0 表示不允许，S1 表示允许，没有 S 参数缺省表示允许；

M302 命令的输出，为当前是否允许冷挤出。允许时会输出：

```
Cold extrusion allowed
```

不允许时会输出：

```
Code extrusion disallowed
```

较不常用指令

这些指令实在无法归类了，只能以“较不常用指令”为名字，放在了一起。

M42 直接读写端口

此命令直接读/写一个 Arduino 端口，为 3D 打印控制软件上位机扩展程序功能提供基础。

参数包括：

- Pnnn 表示 Arduino 的输入/输出端口；输出时固件程序会同时输出到数字端口和模拟端口；输入时固件程序会从数字端口输入；
- Snnn 表示写入输出端口的值，0 到 255 之间是合法的数字；当 S 参数不存在的时候，M42 指令起输入作用；

Repetier-firmware 固件中预先定义了一个表格，称为“敏感端口表格”，所有位于这个表格内的端口，也就是当前已经被步进电机、限位开关以及热敏电阻占用的端口，都不能被 M42 命令影响。其他当前未占用的端口，可以由这条命令进行 IO 操作。

M82/M83 设置挤出头步进电机坐标模式

与 G90/G91 命令类似，这两条命令用于设置挤出头当前坐标模式为绝对坐标模式(M82)或者相对坐标模式(M83)。没有参数。

未设置时缺省值是绝对坐标模式。

需要注意的是，G90/G91 设置的坐标模式，同时对 XYZE 四个轴起作用，但 M82/M83 设置的坐标模式，只对 E 轴（挤出头步进电机）起作用。

M99 暂时关闭步进电机

M99 命令可以暂时关闭 XYZ 轴步进电机一段时间。命令参数包括：

- Snnn 表示所需暂时关闭步进电机的时间，以秒为单位；
- X 表示暂时关闭 X 轴步进电机；
- Y 表示暂时关闭 Y 轴步进电机；
- Z 表示暂时关闭 Z 轴步进电机；

如果 S 参数没有指定，则暂时关闭 10 秒钟时间。暂时关闭时间到达之后，重新打开相应轴的步进电机。

M111 允许/禁止运行时调试标志

运行时调试标志是一组布尔值，一共有 6 个不同的标志，使用位域（Bit Field）的表示方式。用户可以利用 M111 指令修改这些标志的值。相关参数包括：

- Snnn 表示直接将调试标志设置为 S 值；
- Pnnn 表示以位操作的方式，将 P 值与当前调试标志做某种操作。如果 P 值是正数，则进行按位或操作（增加 P 参数所带的标志位）；如果 P 值是负数，则忽略 P 的符号，进行取反后按位与操作（去除 P 参数所带的标志位）；

调试标志的位域，由以下 6 个布尔值组成：

- 第 1 位，值为 1，表示是否回显（Echo）由上位机发送至下位机的命令；
- 第 2 位，值为 2，表示是否输出信息（Info），实际在固件代码中并未使用；
- 第 3 位，值为 4，表示是否输出错误（Error），在固件出错时会将出错信息发送回上位机；
- 第 4 位，值为 8，表示是否进入模拟执行模式（Dry run），在模拟执行模式下，3D 打印机不实际执行上位机发送的命令，只修改 3D 打印机的内存状态；
- 第 5 位，值为 16，表示是否进入调试通讯模式（Communication），实际在固件代码中似乎并未使用；
- 第 6 位，值为 32，表示是否进入禁止移动模式（No Move），在这个模式下，所有对步进电机的移动命令，都会被忽略；

M117 发送消息至 LCD 屏幕

将一条详细发送至 LCD 屏幕，显示为当前状态信息。参数为

message 表示待显示在 LCD 屏幕上的文本；

M120 测试蜂鸣器

使蜂鸣器发出蜂鸣声。参数为

- Snnn 表示发出声音/不发出声音的时间，以毫秒为单位；
- Pnnn 表示重复的次数；

如果 3D 打印机有蜂鸣器，而且是无源蜂鸣器，那么通过 S 参数和 P 参数的组合，可以得到不同频率的声音。比如

M120 S24 P8

可以得到一个较长的蜂鸣声。如果 3D 打印机的蜂鸣器是有源蜂鸣器，那么 M120 指令只能控制蜂鸣时间，不能控制蜂鸣器的声音频率。

M200 设置体积挤出模式

将 3D 打印机设置为“体积挤出模式”，同时设定挤出头直径参数。相关参数包括

- Tnnn 表示对应的挤出头，无 T 参数表示使用当前挤出头；
- Dnnn 表示挤出头的实际直径，无 D 参数表示关闭体积挤出模式；

体积挤出模式，是相对于缺省的“长度挤出模式”而言的另一种挤出模式。在常见的“长度挤出模式”下，G-code 中的使 E 轴运动的 G0/G1 命令，其参数都是以长度单位 mm 作为单位的。这样确实比较简单，但问题是我们在切片的时候，就必须知道要使用的喷头直径，否则无法计算出耗材前进的实际长度。

为了使 G-code 在生成之后适用于多种不同喷头直径的 3D 打印机机型，我们可以在上位机切片时，将 E 轴参数变为以体积单位 mm³ 作为单位，然后在下位机固件中，再设定正在使用的喷头直径，以达到最终正确输出的目的。为了以体积单位 mm³ 作为 E 轴的参数单位，上位机需要将喷头直径设定为 1.128mm（这样，耗材每前进 1mm，会喷出 $1\text{mm}\pi(1.128\text{mm}/2)^2$ 约等于 1mm³ 的耗材。）同时，下位机要使用下面的语句：

```
M200 T0 D0.4
```

将实际的挤出头喷头直径设置为 0.4mm。同时在上位机和下位机进行这样的操作之后，3D 打印机可以在 E 轴参数单位为 mm³ 的情况下，正确完成打印操作。

M209 开启/关闭自动回抽

开启/关闭自动回抽功能。命令参数为

- Snnn 表示是否开启自动回抽功能，1 表示开启，0 表示关闭；

通常上位机切片器负责在合适的位置处加入回抽指令。如果你的切片器功能比较弱，不能加入合适的回抽指令，那么可以打开这个特性，由固件自动回抽。

在固件配置中，定义

```
#define FEATURE_RETRACTION 1
```

表示支持自动回抽功能，在编译中会包含相关的代码。

M280 多头重复打印模式设置

有些特殊配置的 3D 打印机，允许 2~4 个挤出头同时工作，并且这些挤出头动作完全一致，同时打印出多件完全一样的打印件，这种工作模式叫做多头重复打印模式（Ditto mode）。M280 命令对这个模式进行设置。命令参数为

- Snnn 表示这个模式下的挤出头个数；S0 表示关闭多头重复打印模式；S1^{S3} 表示工作在多头重复打印模式下，并且 3D 打印机拥有额外的 13 个挤出头。

在固件配置中，定义

```
#define FEATURE_DITTO_PRINTING 1
```

表示支持多头重复打印模式功能，在编译中会包含相关的代码。

M281 测试硬件看门狗功能

这条命令用于测试 CPU 硬件中的看门狗功能。实际上，就是造成一个死循环，不再执行“喂狗”动作，从而触发 CPU 硬件看门狗，最终（故意地）造成 3D 打印机重启。这条命令只是用于 3D 打印机固件开发测试。

M303 自动测试 PID 参数

自动测试 PID 参数值。命令参数为

- Pnnn 表示待测试的挤出头编号，从 0 开始，P<挤出头个数>代表待测试的是热床；
- Snnn 表示打印温度；
- Rnnn 代表重复测试次数；

X 代表是否保存于 EEPROM 中；

由于加热、散热需要较多时间，这条命令执行时间很长。

M330 测试蜂鸣器

测试（无源）蜂鸣器，产生一个特定频率的声音。命令参数为

- Snnn 表示声音的频率；
- Pnnn 表示声音持续的时间，以毫秒为单位；

如果命令没有包含 S 参数或者 P 参数，则会使用缺省值 S1 以及 P1000。

保存与恢复当前位置

M400 等待当前所有移动指令完成

等待在 3D 打印机内存中待处理的移动命令执行完成。没有相关的参数。

执行这条语句之后，可以保证在下一条 G-code 命令执行时，所有步进电机都不处于运动状态中。

M401 保存当前的位置

将当前位置，包括 XYZE 步进电机，保存于内存的一组专用变量中。未来可以用 M402 命令恢复这组位置。没有相关的参数。

M402 恢复之前保存的位置

恢复之前由 M401 命令保存的位置值。命令参数为

- X 表示恢复 X 位置；
- Y 表示恢复 Y 位置；
- Z 表示恢复 Z 位置；
- E 表示恢复 E 位置；
- Fnnn 表示使用参数给定的速度，无 F 参数时使用当前速度值；

暂停与更换耗材

M600 更换耗材

在拥有显示屏的 3D 打印机上，启动更换耗材向导界面。没有相关的参数。

通常，这个向导界面是从显示屏界面上触发的。M600 命令提供一个接口，使更换耗材向导界面可以从上位机软件触发。

M601 暂停/恢复挤出头

暂停或者恢复挤出头。命令参数为

- Snnn 表示暂停或者恢复，S1 表示暂停挤出头，S0 表示恢复挤出头工作；

暂停挤出头包括停止挤出头加温以及停止挤出头步进电机工作。恢复则相反，加热挤出头到原来的温度。

设置与 EEPROM 管理

固件的设置，是一个比较有趣的话题，很多玩 3D 打印机的朋友，在遇到设置相关的问题时都会犯迷糊。实际上，对于某一项特定的设置，比如说 X 轴的步进电机分辨率，在 3D

打印机主板上，有三个不同的位置（也是三种不同的存储器）保存了这项内容，而它们的值还有可能不同。让我们先来了解一下这些保存设置内容的位置，以方便大家的理解。

首先，是固件配置文件（`configuration.h`）中的设置值。配置文件中的值，会跟随固件一起编译，之后在刷机过程中，保存在了 3D 打印机的静态存储区（Flash ROM）中。除了刷机之外，静态存储区的内容不会发生变动，可以认为是只读的。每次开机的时候，都是一样的值在等待着我们。

第二份设置值，保存在电可擦写静态存储区（EEPROM）。EEPROM 的读写代价，比静态存储区要小。因此，3D 打印机允许在刷机之后，修改设置值，而这些修改之后的设置值，就存储在 EEPROM 之中。每次开机，程序会先检查 EEPROM，如果 EEPROM 中是空白的，则将静态存储区的第一份设置值复制到 EEPROM 之中。而如果 EEPROM 中已经有保存好的设置值，则程序会直接使用 EEPROM 中的值。有些朋友在玩 3D 打印机过程中可能会有这样的经验，就是明明修改了固件配置文件中的设置值，但刷机之后竟然没有发生变化。这种情况，往往就是 EEPROM 在捣鬼了。我们完全可以使用 G-code M502 M500 两条指令（指令的具体含义可以参考下面），重写 EEPROM，解决这样的问题。

第三份设置，保存在内存（RAM）中。实际用户使用的值，就是内存中的值。由于内存只在加电情况下能够保持其中的内容，因此每次开机时，3D 打印机会根据上面描述的逻辑，重建内存中的设置值。如果某条指令修改的是内存中的设置值，那么这也代表着这次修改是一个临时修改，下次开机这个值就会消失了。

总的来说，三份固件设置，使用的优先级是

内存 > EEPROM > 配置文件

但设置的持久性，就要反过来了。明确了解了这些，特别有助于我们解决一些与设置相关的问题，自然，看下面这些命令描述的时候，也就不会迷糊了。

M205 输出 EEPROM 设置

输出 EEPROM 的当前设置值表格。没有相关的参数。

M205 命令的输出，格式为：

```
EPR:2 75 115200 Baudrate

EPR:3 129 0.000 Filament printed [m]

EPR:2 125 0 Printer active [s]
...
```

这是一个很长的输出，我们这里只截取了前三行。每行中，EPR:后面的第一个数字，是这个设置项值的类型。0 代表 8bit 整数类型，1 代表 16bit 整数类型，2 代表 32bit 整数类型，3 代表 32bit 浮点类型。第二个数字，是设置项值的位置（即 EEPROM 中的地址）。第三个数字，是设置项的值。最后，是设置项的意义。

以第一行为例，第一行设置的是通讯波特率（Baudrate），当前值是 115200。波特率设置项，在 EEPROM 中的位置（地址）是 75，这个值是一个 32bit 整数类型，因此占据了从位置 75 开始的连续 4 个字节（也就是位置 75, 76, 77, 78）。

M206 修改 EEPROM 设置

修改 EEPROM 中的某个值。命令参数为

- Pnnn 表示待修改的值的地址（即 EEPROM 中的地址）；
- Tnnn 表示值的类型，0 代表 8bit 整数类型，1 代表 16bit 整数类型，2 代表 32bit 整数类型，3 代表 32bit 浮点类型；
- Snnn 表示值，只能带整数，用于 T 为 0, 1, 2 的情况；
- Xnnn 表示值，只能带浮点数，用于 T 为 3 的情况；

可以看出，M206 指令的使用是很复杂的，需要了解 EEPROM 中数值的存储位置以及数值类型，才能进行有效的修改。因此打印虎建议除非你完全理解 M206 指令的含义，否则不要使用这个指令。

M360 输出固件配置信息

输出固件配置信息。没有相关参数。

M360 命令的输出，格式为：

```
Config:Baudrate:115200
```

```
Config:InputBuffer:127
```

```
Config:NumExtruder:1
```

```
...
```

这是一个很长的输出，我们这里只截取了前三行。每行中，都有一项配置信息的名称，以及对应的值。

M500 保存内存中的设置值到 EEPROM

将 3D 打印机内存中的设置值保存到 EEPROM 中。没有相关的参数。

M501 读取 EEPROM 的设置值到内存

将 EEPROM 中的设置值读取到 3D 打印机内存中。没有相关的参数。

M502 将内存中的设置值重置

将内存中的设置值重置为固件配置（configuration.h）中的值。没有相关的参数。

由于每次系统掉电后，内存中的值都会消失，重新启动时从 EEPROM 中读取，因此单独使用 M502 命令将只对 3D 打印机掉电重启之前起作用。如果想起长期作用，需要配合 M500，将设置值保存到 EEPROM 中。

步进电机参考电压调节

目前市面上支持软件设置步进电机参考电压的 3D 打印机主板很少。大部分 3D 打印机主板只能通过调整微调电位器来控制步进电机参考电压。在这些 3D 打印机上，这一组命令是无效的。

M907 设置步进电机参考电压（百分比值）

设置步进电机参考电压。命令参数为

- Snnn 表示对所有步进电机进行统一设置；
- Xnnn 表示对 X 轴步进电机进行设置；
- Ynnn 表示对 Y 轴步进电机进行设置；
- Znnn 表示对 Z 轴步进电机进行设置；
- Ennn 表示对 E 轴步进电机进行设置；

所有的参数值，都是一个 0~100 之间的百分比数值。

M908 设置步进电机参考电压

与 M907 命令类似，设置步进电机参考电压。命令参数为

- Pnnn 表示步进电机编号；
- Snnn 表示步进电机参考电压设置值，要求为 0~255 之间的一个数值；

这个命令与 M907 命令类似，同样要求 3D 打印机主板支持。在不支持软件调整参考电压的 3D 打印机主板上，M908 命令无效。

M909 输出步进电机参考电压值

输出当前的步进电机参考电压值。没有相关的参数。

M910 将步进电机参考电压值保存至 EEPROM

将 M907/M908 命令设置的步进电机参考电压值保存至 EEPROM。没有相关的参数。

需要辅助硬件支持的指令

M80/M81 打开/关闭 ATX 电源

在配置了 ATX 电源的 3D 打印机上，打开(M80)或者关闭(M81)ATX 电源。没有相关的参数。

M340 伺服电机控制

伺服电机控制功能。命令参数为

- Pnnn 表示伺服电机编号，从 0 开始，最大为 3，可以控制 4 个伺服电机；
- Snnn 为控制时间，单位为毫秒，应该是一个 500 到 2500 之间的数值；
- Rnnn 为自动关闭时间，单位为毫秒；

M350 设置步进电机细分数

在支持细分数设置的 3D 打印机主板上（这类主板很少见），设置步进电机细分数。命令参数为

- Snnn 表示将细分数的每一位（bit）都设置为相同的值，S0 表示所有都设置为 0，S1 表示所有都设置为 1；
- Xnnn 表示设置细分数第 0 位；
- Ynnn 表示设置细分数第 1 位；
- Znnn 表示设置细分数第 2 位；
- Ennn 表示设置细分数第 3 位；
- Pnnn 表示设置细分数第 4 位；

需要注意的是，在大多数 3D 打印机主板上，细分数设置是主板硬件设计时就固定的，不能通过软件调整。这种情况下，M350 命令无效。

M355 设置照明灯开关

设置照明灯的开关。命令参数为

- Snnn 表示照明灯的开关状态，S0 表示关闭照明灯，S1 表示打开照明灯；
无参数时输出当前照明灯的状态。

在固件配置中，定义

```
#define CASE_LIGHTS_PIN -1
```

表示照明灯的电路硬件连接 pin 值，-1 代表照明灯未连接。

M355 命令的输出，为当前是否打开了照明灯。打开时会输出：

```
Case lights on
```

关闭时会输出：

```
Case lights off
```