This page tries to describe the flavour of G-codes that the RepRap firmwares use and how they work. The main target is additive fabrication using FFF/FDM processes (3D printing). Codes for print head movements follow the NIST RS274NGC G-code standard, so RepRap firmwares are quite usable for CNC milling and similar applications

As many different firmwares exist and their developers tend to implement new features without discussing strategies or looking what others did before them, a lot of different sub-flavours for the 3D-Printer specific codes developed over the years.

Contents

[hide]

Introduction

A typical piece of Gcode as sent to a RepRap machine might look like this:

```
N3 T0*57
N4 G92 E0*67
N5 G28*22
N6 G1 F1500.0*82
N7 G1 X2.0 Y2.0 F3000.0*85
N8 G1 X3.0 Y3.0*33
```

The meaning of all those symbols and numbers (and more) is explained below.

Below table clearly shows which specific Gcode(s) are implemented in any given firmware; some are still experimental.

	FiveD	**Teacup**	**Sprinter**	**Marlin**	**Repetier**
Support	yes	automatic	yes	yes	experimental

Here means:

Yes:

The Gcode is fully supported by the firmware.

Experimental:

There is some support for the Gcode. Often it is required to check out the source code branch for the firmware; may also require some modification.

Automatic:

The firmware handles this Gcode automatically, so there's no need to send the command. An example is power supply on/off Gcode (M80/M81) in the Teacup firmware.

No:

The firmware does not support this Gcode.

For the technically minded, Gcode line endings are Unix Line Endings (\n), but will accept Windows Line Endings (\r\n), so you should not need to worry about converting between the two.

RepRap G-code fields

This section explains the letter-preceded fields. The numbers in the fields are represented by nnn. Numbers can be integers, or can contain a decimal point, depending on context. For example an X coordinate can be integer (X175) or fractional (X17.62), whereas trying to select extruder number 2.76 would make no sense.）

Letter	Meaning
Gnnn	Standard GCode command, such as move to a point
Mnnn	RepRap-defined command, such as turn on a cooling fan
Tnnn	Select tool nnn. In RepRap, tools are extruders
Snnn	Command parameter, such as time in seconds; temperatures; voltage to send to a motor
Pnnn	Command parameter, such as time in milliseconds; proportional (Kp) in PID Tuning
Xnnn	A X coordinate, usually to move to. This can be an Integer or Fractional number.
Ynnn	A Y coordinate, usually to move to. This can be an Integer or Fractional number.
Znnn	A Z coordinate, usually to move to. This can be an Integer or Fractional number.
Innn	Parameter - X-offset in arc move; integral (Ki) in PID Tuning
Jnnn	Parameter - Y-offset in arc move
Fnnn	Feedrate in mm per minute. (Speed of print head movement)
Rnnn	Parameter - used for temperatures
Qnnn	Parameter - not currently used
Ennn	Length of extrudate. This is exactly like X, Y and Z, but for the length of filament to extrude. ~~It is common for newer stepper based systems to interpret ...~~ Better: Skeinforge 40 and up interprets this as the absolute length of input filament to consume, rather than the length of the extruded output.
Nnnn	Line number. Used to request repeat transmission in the case of communications errors.
*nnn	Checksum. Used to check for communications errors.

Comments

G Code comments:

```
N3 T0*57 ; This is a comment
N4 G92 E0*67
; So is this
N5 G28*22
```

Comments and white space will be ignored by your RepRap Printer. It's better to strip these out on the host computer before sending the Gcode to your printer, as this saves bandwidth.

Individual commands

Checking codes

N and *

Example: N123 [...G Code here...] *71

These are line number and tagged number. RepRap firmware will compare the tagged number with a locally-computed value and, if they differ, requests a repeat transmission of the line of the given number.

Although RepRap can still work without line number and checking number, it will not do the checking. You may use both or omitting both.

Checksum cs is to use Gcode cmd (including line number) to have bit exclusive-or operation.

Below is example code:

```
int cs = 0;
for(i = 0; cmd[i] != '*' && cmd[i] != NULL; i++)
   cs = cs ^ cmd[i];
cs &= 0xff;    // Defensive programming...
```

The cs value is the decimal digit on the right of *.

The RepRap firmware expects line numbers to increase by 1 each line, and if that doesn't happen it is flagged as an error. But you can reset the count using M110 (see below).

Note: the purpose for line number and checksum is to reduce signal interference.

Delay G-commands

If buffering is supported, the RepRap firmware stores some commands in a ring buffer internally for execution. This means that there is no (appreciable) delay while a command is acknowledged and the next transmitted. In turn, this means that sequences of line segments can be plotted without a dwell between one and the next. As soon as one of these buffered commands is received it is acknowledged and stored locally. If the local buffer is full, then the acknowledgment is delayed until space for storage in the buffer is available. This is how flow control is achieved.

G0: Rapid move

Example: G0 X12

Support	FiveD	Teacup	Sprinter	Marlin	Repetier
	???	yes	yes	yes	yes

It will move X = 12 mm for this command. In fact, for RepRap, the efficiency of this command is the same with G1 X12. (this command is for old machines that prefer moving in folding line to go linear. That's why it's called rapid move)

G1: Controlled move

Example: G1 X90.6 Y13.8 E22.4

Support	FiveD	Teacup	Sprinter	Marlin	Repetier
	yes	yes	yes	yes	yes

From the current position (X, Y) moves to the designated position (90.6, 13.8) while extruding 22.4mm of filament between the two points. (note: the extrusion is controlled by E22.4command)

RepRap is very sensitive to the feedrate, so we first set the feedrate to 1500mm/minute, then move to (X90.6 Y13.8) and do the the extrusion.

```
G1 F1500
G1 X90.6 Y13.8 E22.4
```

But for the code below, we set a feedrate of 1500 mm/minute first, then do the move described above accelerating to a feedrate of 3000 mm/minute as it does so.

```
G1 F1500
G1 X90.6 Y13.8 E22.4 F3000
```

To reverse the extruder by a given amount (for example to reduce its internal pressure while it does an in-air movement so that it doesn't dribble) simply use G0 or G1 to send an E value that is less than the currently extruded length.

Note: not all firmware supports this function (change feedrate during movement). For example, Marlin will use the new feedrate at the very beginning and will not change during the movement.

For the first example, it keeps fixed feedrate during movement; for the second one, it changes the feedrate during the movement, so for below code, the feedrate will first be accelerated to 3000 mm/minute and then reduce to1500 mm/minute.

```
G1 F1500
G1 X90.6 Y13.8 E22.4 F3000
G1 X80 Y20 E36 F1500
```

For the filament retraction, we can reverse the extruder motor to reduce its internal pressure while it does an in-air movement so that it doesn't dribble, which can greatly reduce burr defect during the printing. Specific operation: send G1 Ennn command to RepRap, in which the nnn should be lower than the current value.

G28: Move to Origin

Example: G28

Support	FiveD	Teacup	Sprinter	Marlin	Repetier
	yes	yes	yes	yes	yes

When the RepRap firmware receives this command, it moves all (or the supplied) axis's back to the zero endstops as quickly as it can, then backs off by a millimeter and slowly moves back to the zero endstop activation points to increase position accuracy.

If you add coordinates, only coordinates added will be homed. For example “G28 X0 Y72.3”, only X and Y axis will be homed, no homing for the Z axis (numbers after X and Y will be ignored).

G29-G32: Bed probing

- G29 Detailed Z-Probe

probes the bed at 3 points.

- G30 Single Z Probe

probes bed at current XY location.

- G31 Report Current Probe status

reports whether Z probe is triggered. G32 Probe Z and calibrate with FPU probes the bed at 3 points and updates transformation matrix for bed leveling compensation.

Immediate G command

The below commands will be buffered and be responded only after all the previous buffered commands are executed. So the main machine will wait until all commands are executed (latest respond). These temporary stops will not affect the normal function of the machine.

Teacup Firmware buffers G20, G21, G90 and G91.

G4: Dwell

Example: G4 P200

In this case sit still doing nothing for 200 milliseconds. During delays the state of the machine (for example the temperatures of its extruders) will still be preserved and controlled.

G10: tool offset

Example: G10 P3 X17.8 Y-19.3 Z0.0 R140 S205

This sets the offset for tool (or in older implementations extrude head) 3 (from the P3) to the X and Y values specified. You can put a non-zero Z value in as well, but this is usually a bad idea unless the tools are loaded and unloaded by some sort of tool changer. When all the tools are in the machine at once they should all be set to the same Z height.

Remember that any parameter that you don't specify will automatically be set to the last value for that parameter. That usually means that you want explicitly to set Z0.0.

The R value is the standby temperature in oC that will be used for the tool, and the S value is its operating temperature. If you don't want the tool to be at a different temperature when not in use, set both values the same. See the T code (select tool) .

The NIST G-code standard mentions an additional L parameter, which is ignored.
This command is subject to discussion.

G20: Set Units to Inches

Example: G20
Units from now on are in inches.

G21: Set Units to Millimeters

Example: G21
Units from now on are in millimeters. (This is the RepRap default.)

G90: Set to Absolute Positioning

Example: G90

All coordinates from now on are absolute relative to the origin of the machine. (This is the RepRap default.)

G91: Set to Relative Positioning

Example: G91

All coordinates from now on are relative to the last position

G92: Set Position

Example: G92 X10 E90

Allows programming of absolute zero point, by resetting the current position to the values specified. This would set the machine's X coordinate to 10, and the extrude coordinate to 90. No physical motion will occur.

A G92 without coordinates will reset all axes to zero.

Immediate M and T commands

M0: Stop

Example: M0

The RepRap machine finishes any moves left in its buffer, then shuts down. All motors and heaters are turned off. It can be started again by pressing the reset button on the master microcontroller. See also M1, M112.

M1: Sleep

Example: M1

The RepRap machine finishes any moves left in its buffer, then shuts down. All motors and heaters are turned off. It can still be sent G and M codes, the first of which will wake it up again. See also M0, M112.

M3: Spindle On, Clockwise (CNC specific)

Example: M3 S4000

The spindle is turned on with a speed of 4000 RPM.

M4: Spindle On, Counter-Clockwise (CNC specific)

Example: M4 S4000

The spindle is turned on with a speed of 4000 RPM.

M5: Spindle Off (CNC specific)

Example: M5

The spindle is turned off.

M7: Mist Coolant On (CNC specific)

Example: M7

Mist coolant is turned on (if available)

M8: Flood Coolant On (CNC specific)

Example: M8

Flood coolant is turned on (if available)

M9: Coolant Off (CNC specific)

Example: M9

All coolant systems are turned off.

M10: Vacuum On (CNC specific)

Example: M10

Dust collection vacuum system turned on.

M11: Vacuum Off (CNC specific)

Example: M11

Dust collection vacuum system turned off. (CNC specific)

M17: Enable/Power all stepper motors

Example: M17

M18: Disable all stepper motors

Example: M18

Disables stepper motors and allows axis to move 'freely.' (due to inertia or gravity)

- Is this not the same as M84? -- MrAlvin 05:53, 20 September 2012 (UTC)

M20: List SD card

Example: M20

All files in the root folder of the SD card are listed to the serial port. One name per line, like:

ok Files: {SQUARE.G,SQCOM.G,}

‘, ’ mark is optional . Note that some firmwares list file names in upper case, but - when sent to the M23 command (below) they must be in lower case. This seems to be a function of the SD software. Go figure...

M21: Initialize SD card

Example: M21

The SD card is initialized. If an SD card is loaded when the machine is switched on, this will happen by default. SD card must be initialized for the other SD functions to work.

M22: Release SD card

Example: M22

SD card is released, so further (accidental) attempts to read from it are guaranteed to fail. Helpful, but not mandatory before removing the card physically.

M23: Select SD file

Example: M23 filename.gco

The file specified as filename.gco (8.3 naming convention is supported) is selected ready for printing.

M24: Start/resume SD print

Example: M24

The machine prints from the file selected with the M23 command.

M25: Pause SD print

Example: M25

The machine pauses printing file selected with the M23 command at the current position.

M26: Set SD position

Example: M26

Set SD position in bytes (M26 S12345).

M27: Report SD print status

Example: M27

Report SD print status.

M28: Begin write to SD card

Example: M28 filename.gco

File specified by filename.gco is created (or overwritten if it exists) on the SD card and all subsequent commands sent to the machine are written to that file.

M29: Stop writing to SD card

Example: M29 filename.gco

File opened by M28 command is closed, and all subsequent commands sent to the machine are executed as normal.

M30: Delete a file on the SD card

Example: M30 filename.gco

filename.gco is deleted.

M40: Eject

If your RepRap machine can eject the parts it has built off the bed, this command executes the eject cycle. This usually involves cooling the bed and then performing a sequence of movements that remove the printed parts from it. The X, Y and Z position of the machine at the end of this cycle are undefined (though they can be found out using the M114 command, q.v.).

See also M240 and M241 below.

M41: Loop

Example: M41

If the RepRap machine was building a file from its own memory such as a local SD card (as opposed to a file being transmitted to it from a host computer) this goes back to the beginning of the file and runs it again. So, for example, if your RepRap is capable of ejecting parts from its build bed then you can set it printing in a loop and it will run and run. Use with caution - the only things that will stop it are:

1. When you press the reset button,
2. When the build material runs out (if your RepRap is set up to detect this), and
3. When there's an error (such as a heater failure).

M42: Stop on material exhausted / Switch I/O pin

M42 in ???

Example: M42

If your RepRap can detect when its material runs out, this decides the behaviour when that happens. The X and Y axes are zeroed (but not Z), and then the machine shuts all motors and heaters off. You have to press reset to reactivate the machine. In other words, it parks itself and then executes an M0 command (q.v.).

M42 in Marlin/Sprinter

Example: M42 P7 S255

M42 switches a general purpose I/O pin.

M42 in Teacup

Not needed. General purpose devices are handled like a heater, see M104.

M43: Stand by on material exhausted

Example: M43

If your RepRap can detect when its material runs out, this decides the behaviour when that happens. The X and Y axes are zeroed (but not Z), and then the machine shuts all motors and heaters off except the heated bed, the temperature of which is maintained. The machine will still respond to G and M code commands in this state.

M80: ATX Power On

Example: M80

Turns on the ATX power supply from standby mode to fully operational mode. No-op on electronics without standby mode.

Note: some firmwares, like Teacup, handle power on/off automatically, so this is redundant there.

M81: ATX Power Off

Example: M81

Turns off the ATX power supply. Counterpart to M80.

M82: Set extruder to absolute mode

Example: M82

makes the extruder interpret extrusion as absolute positions.
This is the default in repetier.

M83: Set extruder to relative mode

Example: M83

makes the extruder interpret extrusion values as relative positions.

M84: Stop idle hold

Example: M84

Stop the idle hold on all axis and extruder. In some cases the idle hold causes annoying noises, which can be stopped by disabling the hold. Be aware that by disabling idle hold during printing, you will get quality issues. This is recommended only in between or after printjobs.

M92: set axis_steps_per_unit

Example: M92 X<newsteps> Sprinter and Marlin
Allows programming of steps per unit of axis till the electronics are reset for the specified axis. Very useful for calibration.

M98: get axis_hysteresis_mm

Example: M98

Report current hysteresis value of XYZ axis.
Proposed for Marlin

M99: set axis_hysteresis_mm

Example: M99 X<mm> Y<mm> Z<mm> E<mm>

Mechanical pulleys, gears and threads can have hysteresis when they change direction. That is, a certain number of steps occur before movement occurs. You can measure how many mm are lost to hysteresis and set their values with this command. Every time an axis changes direction, these extra mm will be added to compensate for the hysteresis.
Proposed for Marlin

M101 Turn extruder 1 on (Forward), Undo Retraction

M101 in Teacup firmware

If a DC extruder is present, turn that on. Else, undo filament retraction, which means, make the extruder ready for extrusion. Complement to M103.

M101 in other firmwares

Deprecated. Regarding filament retraction, see M227, M228, M229.

M102 Turn extruder 1 on (Reverse)

Deprecated.

M103 Turn all extruders off, Extruder Retraction

M103 in Teacup firmware

If a DC extruder is present, turn that off. Else, retract the filament in the hope to prevent nozzle drooling. Complement to M101.

M103 in other firmwares

Deprecated. Regarding extruder retraction, see M227, M228, M229.

M104:Set Extruder Temperature

Support	Five D	Teacup	Sprinter	Marlin	Repetier
	yes	yes	yes	yes	yes

Example: M104 S190

Set the temperature of the current extruder to

190oC and return control to the host (PC) immediately (*i.e.* before that temperature has been reached by the extruder). See also M109.

This is deprecated because temperatures should be set using the G10 and T commands (q.v.).

Deprecation is subject to discussion. --Traumflug 11:33, 19 July 2012 (UTC)

M104 in Teacup Firmware

In Teacup Firmware, M104 can be additionally used to handle all devices using a temperature sensor. It supports the additional P parameter, which is a zero-based index into the list of sensors in config.h. For devices without a temp sensor, see M106.

Example: M104 P1 S100

Set the temperature of the device attached to the second temperature sensor to 100 °C.

M105: Get Extruder Temperature

Example: M105

Request the temperature of the current extruder and the build base in degrees Celsius. The temperatures are returned to the host computer. For example, the line sent to the host in response to this command looks like: ok T:201 B:117

M106: fan on

Example: M106 S127

Support	Five D	Teacup	Sprinter	Marlin	Repetier
		yes	yes	yes	

Turn on the cooling fan at half speed.

'Mandatory parameter 'S' declares the PWM value (0-255). M106 S0 turns the fan off.

M106 in Teacup Firmware

Additionally to the above, Teacup Firmware uses M106 to control general devices. It supports the additional P parameter, which is an zero-based index into the list of heaters/devices in config.h.

Example: M106 P2 S255

Turn on device #3 at full speed/wattage.

Note: When turning on a temperature sensor equipped heater with M106 and M104 at the same time, temperature control will override the value given in M106 quickly.

M107: Fan Off

Deprecated in Teacup firmware. Use M106 S0 instead..

M108: Set Extruder Speed

Sets speed of extruder motor. (Deprecated in FiveD firmware, see M113)

M109: Set Extruder Temperature and Wait

Support	Five D	Teacup	Sprinter	Marlin	Repetier
	???	not needed	see text	yes	???

M109 in Teacup

Not needed. To mimic Marlin behaviour,

use M104 followed by M116.

M109 in Marlin, Sprinter (ATmega port)

Set Extruder Temperature and Wait

Example: M109 S185

M109 in Sprinter (4pi port)

Parameters: **S** (optional), set target temperature value. If not specified, waits for the temperature set by M104. **R** (optional), sets target temperature range maximum value.

Example: M109 S185 R240 //sets extruder temperature to 185 and waits for the temperature to be between 185 - 240.

If you have multiple extruders, use **T** or **P** parameter to specify which extruder you want to set/wait. Another way to do this is to use G10.

M110: Set Current Line Number

Support	Five D	Teacup	Sprinter	Marlin	Repetier
	???	not needed	???	???	???

Example: M110 N123

Set the current line number to 123. Thus the expected next line after this command will be 124.

M111: Set Debug Level

Example: M111 S6

Set the level of debugging information transmitted back to the host to level 6. The level is the OR of three bits:

```
#define DEBUG_ECHO (1<<0)
#define DEBUG_INFO (1<<1)
#define DEBUG_ERRORS (1<<2)
```

Thus 6 means send information and errors, but don't echo commands. (This is the RepRap default.)

Example: M253

Support	Five D	Teacup	Sprinter	Marlin	Repetier
		Debug			

M112: Emergency Stop

Example: M112

Any moves in progress are immediately terminated, then RepRap shuts down. All motors and heaters are turned off. It can be started again by pressing the reset button on the master microcontroller. See also M0 and M1.

M113: Set Extruder PWM

Example: M113

Set the PWM for the currently-selected extruder. On its own this command sets RepRap to use the on-board potentiometer on the extruder controller board to set the PWM for the currently-selected extruder's stepper power. With an S field:

M113 S0.7

it causes the PWM to be set to the S value (70% in this instance). M113 S0 turns the extruder off, until an M113 command other than M113 S0 is sent.

M114: Get Current Position

Example: M114

This causes the RepRap machine to report its current X, Y, Z and E coordinates to the host.

For example, the machine returns a string such as:

ok C: X:0.00 Y:0.00 Z:0.00 E:0.00

M115: Get Firmware Version and Capabilities

Example: M115

Request the Firmware Version and Capabilities of the current microcontroller The details are returned to the host computer as key:value pairs separated by spaces and terminated with a linefeed.

sample data from firmware:

```
ok PROTOCOL_VERSION:0.1 FIRMWARE_NAME:FiveD
FIRMWARE_URL:http%3A//reprap.org MACHINE_TYPE:Mendel
EXTRUDER_COUNT:1
```

This M115 code is inconsistently implemented, and should not be relied upon to exist, or output correctly in all cases. An initial implementation was committed to svn for the FiveD Reprap firmware on 11 Oct 2010. Work to more formally define protocol versions is currently (October 2010) being discussed. See M115_Keywords for one draft set of keywords and their meanings.

M116: Wait

Example: M116

Wait for all temperatures and other slowly-changing variables to arrive at their set values. See also M109.

M117: Get Zero Position

Example: M117

This causes the RepRap machine to report the X, Y, Z and E coordinates *in steps not mm* to the host that it found when it last hit the zero stops for those axes. That is to say, when you zero X, the *x* coordinate of the machine when it hits the X endstop is recorded. This value should be 0, of course. But if the machine has drifted (for example by dropping steps) then it won't be. This command allows you to measure and to diagnose such problems. (E is included for completeness. It doesn't normally have an endstop.)

M117 in Marlin: Display Message

Example: M117 Hello World

This causes the given message to be shown in the status line on an attached LCD. The above command will display Hello World.

M118: Negotiate Features

Example: M118 P42

This M-code is for future proofing. NO firmware or hostware supports this at the moment. It is used in conjunction with M115's FEATURES keyword.

See Protocol_Feature_Negotiation for more info.

M119: Get Endstop Status

Support	Five D	Teacup	Sprinter	Marlin	Repetier
		yes		yes	

Example: M119

Returns the current state of the configured X, Y, Z endstops. Takes into account any 'inverted endstop' settings, so one can confirm that the machine is interpreting the endstops correctly.

M120: Push

Push the state of the RepRap machine onto a stack. Exactly what variables get pushed depends on the implementation (as does the depth of the stack - a typical depth might be 5). A sensible minimum, however, might be

1. Current feedrate, and
2. Whether moves (and separately extrusion) are relative or absolute

M121: Pop

Recover the last state pushed onto the stack.

M126: Open Valve

Example: M126 P500

Open the extruder's valve (if it has one) and wait 500 milliseconds for it to do so.

M127: Close Valve

Example: M127 P400

Close the extruder's valve (if it has one) and wait 400 milliseconds for it to do so.

M128: Extruder Pressure PWM

Example: M128 S255

PWM value to control internal extruder pressure. S255 is full pressure.

M129: Extruder pressure off

Example: M129 P100

In addition to setting Extruder pressure to 0, you can turn the pressure off entirely. P400 will wait 100ms to do so.

M130: Set PID P value

Support	Five D	Teacup	Sprinter	Marlin	Repetier
		yes			

Example: M130 P 0 S 8.0 # Sets heater 0 P factor to 8.0

M131: Set PID I value

Example: M130 P 1 S 0.5 # Sets heater 1 I factor to 0.5

Support	FiveD	Teacup	Sprinter	Marlin	Repetier
		yes			

M132: Set PID D value

Example: M130 P 0 S 24 # Sets heater 0 D factor to 24.0

Support	FiveD	Teacup	Sprinter	Marlin	Repetier
		yes			

M133: Set PID I limit value

Example: M130 P 1 S 264 # Sets heater 0 I limit value to 264

Support	FiveD	Teacup	Sprinter	Marlin	Repetier
		yes			

M134: Write PID values to EEPROM

Example: M134

Support	FiveD	Teacup	Sprinter	Marlin	Repetier
		yes			

M136: Print PID settings to host

Example: M136 P1 # print heater 0 PID parameters to host

Support	FiveD	Teacup	Sprinter	Marlin	Repetier
		Debug			

M140: Bed Temperature (Fast)

Example: M140 S55

Set the temperature of the build bed to 55°C and return control to the host immediately (*i.e.* before that temperature has been reached by the bed).

Support	Five D	Teacup	Sprinter	Marlin	Repetier
		yes	yes	yes	yes

M141: Chamber Temperature (Fast)

Example: M141 S30

Set the temperature of the chamber to 30°C and return control to the host immediately (*i.e.* before that temperature has been reached by the chamber).

M142: Holding Pressure

Example: M142 S1

Set the holding pressure of the bed to 1 bar.

The holding pressure is in bar. For hardware which only has on/off holding, when the holding pressure is zero, turn off holding, when the holding pressure is greater than zero, turn on holding.

M143: Maximum hot-end temperature

Example: M143 S275

Set the maximum temperature of the hot-end to 275C

When temperature of the hot-end exceeds this value, take countermeasures, for instance an emergency stop. This is to prevent hot-end damage.

M160: Number of mixed materials

Example: M160 S4

Set the number of materials, N, that the current extruder can handle to the number specified. The default is 1.

When N >= 2, then the E field that controls extrusion requires N+1 values separated by spaces after it like this:

```
M160 S4
G1 X90.6 Y13.8 E22.4 0.1 0.1 0.1 0.7
G1 X70.6 E42.4 0.0 0.0 0.0 1.0
G1 E42.4 1.0 0.0 0.0 0.0
```

The second line moves straight to the point (90.6, 13.8) extruding 22.4mm of filament. The mix ratio at the **end** of the move is 0.1:0.1:0.1:0.7.

The third line moves back 20mm in X extruding 20mm of filament. The mix varies linearly from 0.1:0.1:0.1:0.7 to 0:0:0:1 as the move is made.

The fourth line has no physical effect, but sets the mix proportions for the start of the next move to 1:0:0:0.

M190: Wait for bed temperature to reach target temp

Support	Five D	Teacup	Sprinter	Marlin	Repetier
		obsolete, see M116	yes	yes	

Example: M190 S60

This will wait until the bed

temperature reaches 60 degrees,

printing out the temperature of the hot end and the bed every second.

M200 - Set filament diameter / Get Endstop Status

Support	Five D	Teacup	Sprinter	Marlin	Repetier
				yes	

M200 Set filament diameter.

Question: what does a firmware do with

filament diameter? Has this an effect on how much an E command moves the extruder motor?

--Traumflug 11:34, 14 October 2012 (UTC)

M201 - Set max printing acceleration

in units/s^2 for print moves (M201 X1000 Y1000)

M202 - Set max travel acceleration

in units/s^2 for travel moves (M202 X1000 Y1000) Unused in Marlin!!

M203 - Set maximum feedrate

Sets the maximum feedrates that your machine can do in mm/min.

Example: (M203 X200 Y200 Z300 E10000)

Note: this should be in units/minute, just like the F code.

M204 - Set default acceleration

S normal moves T filament only moves (M204 S3000 T7000) im mm/sec^2 also sets minimum segment time in ms (B20000) to prevent buffer underruns and M20 minimum feedrate

M205 - Advanced settings

minimum travel speed S=while printing T=travel only, B=minimum segment time X= maximum xy jerk, Z=maximum Z jerk, E=maximum E jerk

M206: Set home offset

Example: M206 X10.0 Y10.0 Z-0.4

The values specified are added to the endstop position when the axes are referenced. The same can be achieved with a G92 right after homing (G28, G161).

With Marlin firmware, this value can be saved to EEPROM using the M500 command.

A similar command is G10, aligning these two is subject to discussion.

M207: Calibrate z axis by detecting z max length

Example: M207

After placing the tip of the nozzle in the position you expect to be considered Z=0, issue this command to calibrate the Z axis. It will perform a z axis homing routine and calculate the distance traveled in this process. The result is stored in EEPROM as z_max_length. For using this calibration method the machine must be using a Z MAX endstop.

This procedure is usually more reliable than mechanical adjustments of a Z MIN endstop.

M208: Set axis max travel

Example: M208 X250 Y210 Z180

With Marlin firmware, this value can be saved to EEPROM using the M500 command.

The values specified set the software limits for axis travel in the positive direction.

With Marlin firmware, this value can be saved to EEPROM using the M500 command.

M209: Enable automatic retract（Retraction）

Example: M209 S1

This boolean value S 1=true or 0=false enables automatic retract detect if the slicer did not support G10/11: every normal extrude-only move will be classified as retract depending on the direction.

M220:set speed factor override percentage

Example: M220 S80

S<factor in percent>- set speed factor override percentage

M221: set extrude factor override percentage

Example: M221 S70

S<factor in percent>- set extrude factor override percentage

M226: Gcode Initiated Pause

Example: M226

Initiates a pause in the same way as if the pause button is pressed. That is, program execution is stopped and the printer waits for user interaction. This matches the behaviour of M1 in the NIST RS274NGC G-code standard and M0 in Marlin firmware.

M227: Enable Automatic Reverse and Prime

Example: M227 P1600 S1600

P and S are steps.

"Reverse and Prime" means, the extruder filament is retracted some distance when not in use and pushed forward the same amount before going into use again. This shall help to prevent drooling of the extruder nozzle. Teacup firmware implements this with M101/M103.

M228: Disable Automatic Reverse and Prime

Example: M228

See also M227.

M229: Enable Automatic Reverse and Prime

Example: M229 P1.0 S1.0

P and S are extruder screw rotations. See also M227.

M230: Disable / Enable Wait for Temperature Change

Example: M230 S1

S1 Disable wait for temperature change S0 Enable wait for temperature change

M240: Start conveyor belt motor / Echo off

Support	Five D	Teacup	Sprinter	Marlin	Repetier
		Debug: Echo off			

Example: M240

The conveyor belt allows to start

mass production of a part with a reprap.

Echoing may be controlled in some firmwares with M111

M241: Stop conveyor belt motor / echo on

Support	Five D	Teacup	Sprinter	Marlin	Repetier
		Debug: Echo on			

Example: M241

Echoing may be controlled in

some firmwares with M111

M245: Start cooler

Example: M245

used to cool parts/heated-bed down after printing for easy remove of the parts after print

M246: Stop cooler

Example: M246

M300: Play beep sound

Usage: M300 S<frequency Hz> P<duration ms>

Example: M300 S300 P1000

Play beep sound, use to notify important events like the end of printing. See working example on R2C2 electronics.

M301: Set PID parameters - Hot End

Support	Five D	Teacup	Sprinter	Marlin	Repetier
				PID	

Example: M301 P1 I2 D3 C5

Sets Proportional, Integral and Derivative

values for hot end, the value C refers to an extrusion rate.

Alternate implementation

Example: M301 W125

M303: Run PID tuning

Support	Five D	Teacup	Sprinter	Marlin	Repetier
				PID	PID

Hotend Usage: M303

S<temperature> C<cycles> Bed Usage: M303 E-1 C<cycles> S<temperature> Example: M303 C8 S175

Generate Proportional, Integral and Derivative values for the hotend or bed (E-1). Send the appropriate code and wait for the output to update the firmware.

M304: Set PID parameters - Bed

Support	Five D	Teacup	Sprinter	Marlin	Repetier
				PID	

Example: M304 P1 I2 D3

Sets Proportional, Integral and Derivative values for bed

M420: Set RGB Colors as PWM

Usage: M420 R<Red PWM (0-255)> E<Green PWM (0-255)> B<Blue PWM (0-255)>

Example: M420 R255 E255 B255

Set the color of your RGB LEDs that are connected to PWM-enabled pins. Note, the Green color is controlled by the E value instead of the G value due to the G code being a primary code that cannot be overridden.

T: select tool

Example: T1

Select tool (or in older implementations extruder) number 1.

The sequence followed is:

1. Set the current extruder to its standby temperature specified by G10 (see above),
2. Set the new extruder to its operating temperature specified by G10 and wait for **all** temperatures to stabilise,
3. Apply any X, Y, Z offset for the new extruder specified by G10,
4. Use the new extruder.

Selecting a non-existent tool (100, say) just does Step 1. above. That is to say it leaves all tools in their standby state. You can, of course, use the G10 command beforehand to set that standby temperature to anything you like.

Note that you may wish to move to a parking position *before* executing a T command in order to allow the new extruder to reach temperature while not in contact with the print. It is acceptable for the firmware to apply a small offset [by convention (-1mm x tool-number) in Y] to the current position when the above sequence is entered to allow temperature changes to take effect just away from the parking position. Any such offset must, of course, be undone when the procedure finishes.
If the Z value changes in the offsets and the head moves up, then the Z move is made before the X and Y moves. If Z moves down, X and Y are done first.
After a reset extruders will not start heating until they are selected. You can either put them all at their standby temperature by selecting them in turn, or leave them off so they only come on if/when you first use them. The M0, M1 and M112 commands turn them all off. You can, of course, turn them all off with the M1 command, then turn some back on again. Don't forget also to turn on the heated bed (if any) if you use that trick.
Extruder numbering starts at 0.

Proposed EEPROM configuration codes

BRIEFLY: each RepRap has a number of physical parameters that should be persistent, but easily configurable, such as extrusion steps/mm, various max values, etc. Those parameters are currently hardcoded in the firmware, so that a user has to modify, recompile and re-flash the firmware for any adjustments. These configs can be stored in MCU's EEPROM and modified via some M-codes. Please see the detailed proposal at M-codes for EEPROM config. (*This is proposed by --AlexRa on 11-March-2011. There is currently no working implementation of the proposed commands*).
Marlin uses these codes to manipulate EEPROM values.
Sprinter has implemented the following commands to manipulate EEPROM Commit message.
Teacup uses codes M130-M136 to set, read, and save some parameters.

M500: Store parameters in EEPROM

M501: Read parameters from EEPROM

If you need to reset them after you changed them temporarily

M502: Revert to the default "factory settings."

It must be followed with M501to store this setting.

M503: Print settings

Replies from the RepRap machine to the host computer

All communication is in printable ASCII characters. Messages sent back to the host computer are terminated by a newline and look like this:
xx [line number to resend] [T:93.2 B:22.9] [C: X:9.2 Y:125.4 Z:3.7 E:1902.5] [Some debugging or other information may be here]
xx can be one of:

ok

rs

!!

ok means that no error has been detected.

rs means resend, and is followed by the line number to resend.

!! means that a hardware fault has been detected. The RepRap machine will shut down immediately after it has sent this message.

The **T:** and **B:** values are the temperature of the currently-selected extruder and the bed respectively, and are only sent in response to M105. If such temperatures don't exist (for example for an extruder that works at room temperature and doesn't have a sensor) then a value below absolute zero (-273^{o}C) is returned.

C: means that coordinates follow. Those are the **X: Y:** etc values. These are only sent in response to M114 and M117.

The RepRap machine may also send lines that look like this:

// **This is some debugging or other information on a line on its own. It may be sent at any time.**

Such lines will always be preceded by //.

The most common response is simply:

ok

When the machine boots up it sends the string

start

once to the host before sending anything else. This should not be replaced or augmented by version numbers and the like. M115 (see above) requests those.

All this means that every line sent by RepRap to the host computer except the start line has a two-character prefix (one of **ok**, **rs**, **!!** or //). The machine should never send a line without such a prefix.

Exceptions: Marlin 1.0.0 Gen6 Firmware does not follow the two character rule. 'rs' is actually 'Resend' and '!!' is 'Error'. Example Lines:

- Error: Line Number is not current line + 1. Last Line: 7
- Resend: 8
- Writing to File: print.gco
- Done saving file.
- File opened:print.gco Size:22992
- File selected

When in the code base did this change take place and what other firmwares are affected?

Proposal for sending multiple lines of G-code

So far, this is a proposal, open for discussion.

Problem to solve

Each line of G-code sent from the host to the controller is answered with an **ok** before the next line can be sent without locking communcations up. This makes operations very slow, as the usual USB-TTL converters and probably also the host's operating system drivers come with substantial latency, often 10 milliseconds.

For more details on this proposal, and some suggested solutions, and comments please see GCODE_buffer_multiline_proposal

Alternatives to G-code

Main article: Firmware/Alternative#alternatives to G-code

- Wikipedia: STEP-NC: "STEP-NC was designed to replace ... G-codes ... adding tolerance data ... [with a] XML format."
- Elegant multispline motion controller "will not use G-code. It will use a custom language based on cubic Bezier curves. This allows for much better description of arcs and will result in much higher quality prints with a much lower data throughput requirements."

Category:

- Model manufacturing software/zh cn