G-code

From RepRapWiki

Contents

Introduction

This page describes the G Codes that the RepRap firmware uses and how they work.

The list of what can be done is extensible. But check this page first, and add your extension here first **before** you implement it.

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.

(For the technically minded, the end of line is marked by both a <nl> and a <cr>. If you want to manually enter GCodes in your reprap using the Arduino Serial interface, make sure to select "Both NL & CR" on the bottom of the screen.)

This isn't accurate, firmware accepts single <nl> as well as single <cr>, or combinations thereof. Hosts should accepts all four combinations as well. --Traumflug 09:34, 20 April 2011 (UTC)

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 the voltage to send to a motor
Pnnn	Command parameter, such as a time in milliseconds
Xnnn	An X coordinate, usually to move to
Ynnn	A Y coordinate, usually to move to
Znnn	A Z coordinate, usually to move to
Innn	Parameter - not currently used
Jnnn	Parameter - not currently used
Fnnn	Feedrate in mm per minute.
Rnnn	Parameter - not currently used
Qnnn	Parameter - not currently used
Ennn	Length of extrudate in mm. This is exactly like X, Y and Z, but for the length of filament to extrude.
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
```

Will be ignored by RepRap, as will blank lines. But it's better to strip these out in the host computer before the lines are sent. This saves bandwidth.

Individual commands

Checking

N and *

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

These are the line number and the checksum. The RepRap firmware checks the checksum against a locally-computed value and, if they differ, requests a repeat transmission of the line of the given number.

You can leave both of these out - RepRap will still work, but it won't do checking. You have to have both or neither though.

The checksum "cs" for a GCode string "cmd" (including its line number) is computed by exor-ing the bytes in the string up to and not including the * character as follows:

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

and the value is appended as a decimal integer to the command after the * character.

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).

Buffered G Commands

The RepRap firmware stores these 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

In this case move rapidly to X = 12 mm. In fact, the RepRap firmware uses exactly the same code for rapid as it uses for controlled moves (see G1 below), as - for the RepRap machine - this is just as efficient as not doing so. (The distinction comes from some old machine tools that used to move faster if the axes were not driven in a straight line. For them G0 allowed any movement in space to get to the destination as fast as possible.)

G1: Controlled move

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

Go in a straight line from the current (X, Y) point to the point (90.6, 13.8), extruding material as the move happens from the current extruded length to a length of 22.4 mm.

RepRap does subtle things with feedrates. Thus:

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

Will set a feedrate of 1500 mm/minute, then do the move described above at that feedrate. But

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

Will set a feedrate of 1500 mm/minute, then do the move described above accelerating to a feedrate of 3000 mm/minute as it does so. The extrusion will accelerate along with the X, Y movement so everything stays synchronized.

RepRap thus treats feedrate as simply another variable (like X, Y, Z, and E) to be linearly interpolated. This gives complete control over accelerations and decelerations in a way that ensures that everything moves together and the right volume of material is extruded at all points.

The first example shows how to get a constant-speed movement. The second how to accelerate or decelerate. Thus

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

Will do the first movement accelerating as before, and the second decelerating from 3000 mm/minute back to 1500 mm/minute.

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 G1 to send an E value that is less than the currently extruded length.

G28: Move to Origin

Example: G28

This causes the RepRap machine to move back to its X, Y and Z zero endstops. It does so accelerating, so as to get there fast. But when it arrives it backs off by 1 mm in each direction slowly, then moves back slowly to the stop. This ensures more accurate positioning.

If you add coordinates, then just the axes with coordinates specified will be zeroed. Thus

G28 X0 Y72.3

will zero the X and Y axes, but not Z. The actual coordinate values are ignored.

Unbuffered G commands

The following commands are not buffered. When one is received it is stored, but it is not acknowledged to the host until the buffer is exhausted and then the command has been executed. Thus the host will pause at one of these commands until it has been done. Short pauses between these commands and any that might follow them do not affect the performance of the machine.

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.

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 reseting 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.

Unbuffered 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 M112.

M20: List SD card

Example: M20

All files in the root folder of the SD card are listed to the serial port.

M21: Initialise SD card

Example: M21

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

M22: Release SD card

Example: M22

SD card is released and can be physically removed.

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 pause printing at the current position within the file selected with the M23 command.

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.

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.

M104: Set Extruder Temperature (Fast)

Example: M104 S190

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

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

Turn on the cooling fan (if any).

M107: Fan Off

Example: M107

Turn off the cooling fan (if any).

M108: Set Extruder Speed

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

M109: Set Extruder Temperature

Example: M109 S190

Set the temperature of the current extruder to 190ºC and wait for it to reach that value before sending an acknowledgment to the host. In fact the RepRap firmware waits a while after the temperature has been reached for the extruder to stabilise - typically about 40 seconds. This can be changed by a parameter in the firmware configuration file when the firmware is compiled. See also M104 and M116.

M110: Set Current Line Number

Example: N123 M110

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.)

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.

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
```

This M115 code is NOT yet implemented in any released firmware, and should not be relied upon. 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.)

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

Example: M119

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

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.

M140: Bed Temperature (Fast)

Example: M140 S55

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

M141: Chamber Temperature (Fast)

Example: M141 S30

Set the temperature of the chamber to 30ºC 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.

M226: Gcode Initiated Pause

Example: M226

Initiates a pause in the same way as if the pause button is pressed.

M227: Enable Automatic Reverse and Prime

Example: M227 P1600 S1600

P and S are steps.

M228: Disable Automatic Reverse and Prime

Example: M228

M229: Enable Automatic Reverse and Prime

Example: M229 P1.0 S1.0

P and S are extruder screw rotations.

M230: Disable / Enable Wait for Temperature Change

Example: M230 S1

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

T: Select Tool

Example: T1

Select extruder number 1 to build with. Extruder numbering starts at 0.

RepRap G-Code Summary Table

Command	Class	Buffered?	Parameters	Action
G0	G	Y	Xnnn Ynnn Znnn Ennn Fnnn	Rapid move to position specified by parameters
G1	G	Y	Xnnn Ynnn Znnn Ennn Fnnn	Controlled move to position specified by parameters

G4	G	N	Pnnn	Delay for a period of nnn milliseconds
G20	G	N	none	Set distance units for motion and position values to inches
G21	G	N	none	Set distance units for motion and position values to millimeters (default)
G28	G	Y	Xnnn Ynnn Znnn	Move to origin (on specified axes only, if X/Y/Z parameters are present)
G90	G	N	none	Set absolute positioning (for subsequent motion commands) (default)
G91	G	N	none	Set relative positioning (for subsequent motion commands)

G92	G	N		Xnnn Ynnn Znnn Ennn	Define current absolute position
M0	M	N		none	Stop (after completing any buffered commands)
M104	M	N		Snnn	Set current extruder temperature (in Celsius) to value of parameter, and return
M105	M	N		none	Request current extruder and base temperatures (in Celsius)
M106	M	N		none	Set cooling fan off
M107	M	N		none	Set cooling fan on
M108	M	N		Snnn	Set extruder motor speed (DEPRECATED)
M109	M	N		Snnn	Set extruder temperature (in Celsius), and wait for it

M110	M	N	none	Set current line number to Nxxx value preceeding command
M111	M	N	Snnn	Set debug level bitfield to value of parameter (default 6)
M112	M	N	none	Emergency stop (stop immediately, discarding any buffered commands)
M113	M	N	Snnn	Set Extruder PWM (to value defined by pot, or to parameter value if present)
M114	M	N	none	Get Current Position (return current X, Y, Z and E values)
M115	M	N	none	Get firmware version (to be implemented)
M116	M	N	none	Wait for all temperature and slow-changing

				variables to reach set values
M117	M	N	none	Get Zero Position (return X, Y, Z and E values of endstop hits)
M119	M	N	none	Get Endstop Status (return current X,Y,Z min and max(if configured) endstop values)
M126	M	N	Pnnn	Open Valve and wait for nnn milliseconds for it to open
M127	M	N	Pnnn	Close Valve and wait for nnn milliseconds for it to close
M140	M	N	Snnn	Set build bed temperature (in Celsius) to value of parameter, and return
M141	M	N	Snnn	Set chamber temperature (in Celsius) to value of parameter,

				and return
M142	M	N	Snnn	Set holding pressure of bed to value of parameter
M143	M	N	Snnn	Set maximum hot-end temperature
M226	M	N	none	Gcode initiated pause
M227	M	N	Pnnn Snnn	Enable Automatic Reverse and Prime (steps parameters)
M228	M	N	none	Disable Automatic Reverse and Prime
M229	M	N	Pnnn Snnn	Enable Automatic Reverse and Prime (rotations parameters)
M230	M	N	Sn	Disable / Enable Wait for Temperature Change

Tn	T	N	none	Select Tool n (T0 selects tool 0) (default extruder is 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*).

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]

where **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.

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

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. We should implement an M code to request those.

Retrieved from "http://reprap.org/wiki/G-code"
Category: Reference

- This page was last modified on 30 June 2011, at 18:39.
- Content is available under GNU Free Documentation License 1.2.