Notes on dc42 fork of RepRapFirmware

Introduction

This is documentation for using my fork of RepRapFirmware, which is 3D printer firmware for the Duet printer electronics board.

The major enhancements in my fork compared to the RepRapPro original at the time of writing this are:

- Completely rewritten move algorithm, with generation of step pulses at the correct intervals (not Bresenham approximation) during all phases of movement. Some users report that this produces higher quality prints and a quieter sound. It also avoids the need for minimum speeds.
- Support for delta printers, with segmentation-free delta movement generation, and fast 7-factor auto-calibration.
- Support for CoreXY, CoreXZ and CoreYZ printers.

There are also a lot of other less dramatic improvements and bug fixes. See document “Changes in dc42 fork.txt” for the full list.

Migrating to my fork from the official RepRapFirmware

General

When migrating to my fork from official RRP firmware with the same major version number, you can if you wish continue using the same files in the /sys folder instead of migrating to mine. However if you have an Ormerod 1 then you will need to modify config.g to tell the firmware that the Z endstop is at the high end, as described in the next section.

Configuring the endstops

The official RepRapFirmware does not distinguish between high and low endstops. My fork does. By default, endstop microswitches are assumed to be at the low end of the corresponding axes. If you have an endstop at the high end of an axis, or the endstop switched produce active-low outputs instead of active-high, then you need to add a command in config.g to specify this. Here is the command needed for the Ormerod 1, which has an endstop switch at the high end of the Y axis:

```
M574 X0 Y2 Z0 S1    ; set endstop configuration (Y endstop only, at high end, active high)
```

The X0 and Z0 parameters specify that there are no endstop switches on the X and Z axes. The Y2 parameter specifies an endstop switch at the upper end of the Y axis (Y1 would specify an endstop switch at the lower end). S1 specifies active high endstop switches (S0 would specify active low).

Homing files

The official RepRapPro firmware relies on having G92 commands in the homing files after every homing move (a homing move is a G1 command with the S1 parameter included). You can use this technique with my fork, but it is not required. When a homing move completes by triggering an endstop, the head position is set to the axis minimum (for a high end switch) or the axis minimum (for a low end switch). So if you set up the correct axis maxima and minima, you do not need to use G92 commands after homing moves. The axis maxima are set using the M208 command, and the axis minima are set using M208 with the S1 parameter.

On a Cartesian (not CoreXY) printer, my fork supports simultaneous homing of the X and Y axes.

Additional GCodes and parameters supported by my fork (compared to RepRapPro version 1.09)

Gcode	Extra parameters	Purpose
G0, G1	S2	On a delta or CoreXY printer, S2 specifies that the individual motors (instead of the print head) should move according to the X, Y and Z parameters. Only valid for relative moves (see G91).
G30	S	At the end of a probe sequence, specifies how many factors to adjust for bed compensation (on a Cartesian or CoreXY printer) or delta calibration (on a delta printer). Default is 0 which has the special meaning “the number of points probed”. When doing delta calibration, this value must be 3, 4, 6 or 7. A negative value means don't perform any bed compensation or delta calibration, just print the height errors at the points probed.
G31	X, Y	Define Z probe X and Y offsets from head reference point. Should be left at zero on a delta printer, because when doing delta calibration it is the head position that matters, not the probe position.
M20	S2	S2 specifies that output shall be in JSON format (for PanelDue)
M36		Return file information in JSON format (for PanelDue)
M37		M37 S1 enters simulation mode. Subsequent G and M codes will not be acted upon, but the time they would take to execute will be calculated. M37 S0 leaves simulation mode. M37 with no S parameter returns the time taken by the simulation.

M117		Display message on PanelDue and/or web interface
M208	S1	Set axis minima
M220		Set speed factor override percentage
M221		Set extrusion factor override percentage
M300		Play beep sound (for PanelDue)
M301, M304	S, T	The S parameter sets the heater output multiplication factor. This is useful for adapting to different heaters powers or supply voltages without having to adjust all of the P, I, D parameters. The T parameter is used to pre-set the I term when switching from full-on to PID, for faster response.
M305	H, L, X	H = ADC high end offset, for correcting ADC errors at low temperatures. L = ADC low end offset, for correcting ADC errors at high temperatures. In theory, all thermistor channels should need the same H and L parameters. X = thermistor channel to use for this heater, default is same as heater number.
M558	R, H, S3, S4	R = Z probe modulation channel (use channel 0 for Duet 0.6, channel 1 for Duet 0.7). H = Z-probe dive height, default 3mm more than the Z probe trigger height. S3 specifies an alternative smart Z probe (modulation pin is driven low). S4 specifies a switch connected to the E0 endstop pins.
M563	S	Specifies a value to add to every tool number in any future GCode received from this input source. Purpose was to map tools 0 and 1 in GCode files generated by slic3r to tools 1 and 2. No longer needed, now that the standard tool numbering for RepRapFirmware starts at zero.
M572		Set extruder elasticity compensation
M574		Set endstop configuration
M575		Set serial communications parameters
M665		Set delta configuration
M666		Set delta endstop adjustment
M667		Set CoreXY/CoreXZ/CoreYZ mode
M906	I	Set percentage of normal motor current to use when idle, default 30
M999	S4321	M999 normally causes a software reset. If the S4321 parameter is present, then the processor goes into bootloader mode after resetting, ready for uploading new firmware.

Migrating to RepRapFirmware from Marlin firmware

RepRapFirmware differs from the popular Marlin firmware in some important respects:

- All important firmware parameters can be configured using GCodes. You do not have to rebuild the firmware from sources in order to make the usual configuration changes. You need only download and flash the binary, unless you want to make changes to the firmware.
- A single build is used for Cartesian, Delta and CoreXY printers.
- Although RepRapFirmware still uses some parts of the Arduino Due core, it is neither built nor programmed using the Arduino IDE. To upload firmware, you use the bossac program and associated USB driver. You must use the version installed with versions of the Arduino IDE that support the Arduino Due, because the official build from Shumatech does not yet support the ATSAM3X8E processor. Instructions for using bossac to upload firmware to the Duet can be found at https://reprappro.com/documentation/commissioning-introduction/commissioning-duet/.
- A number of firmware functions are programmed using macro files read from the /sys folder on the SD card. In particular, at start up the GCodes in file config.g is executed, or (if that file is not found), the commands in default.g. Homing of axes for a Cartesian printer causes one or more of files homex.g, homey.g, homez.g and homeall.g to be executed. When the firmware is configured for a delta printer, file homedelta.g is executed. So you can change the startup configuration and homing behaviour by editing these files. Similarly, G32 bed probing behaviour is controlled by the bed.g file. When you pause a print, file pause.g is executed, and file resume.g is run if you resume the print.
- RepRapFirmware can be controlled by three different interfaces. You can connect it to a PC using the native USB interface, and use a PC program such as Pronterface to control the printer. You can connect the printer's Ethernet port to your local network and use Chrome or FireFox browser to control the printer. And you can connect a PanelDue touch-screen control panel to the serial port on the expansion connector.

Configuring RepRapFirmware for a delta printer

I provide sample config.g, homedelta.g and bed.g files for the Mini Kossel. You can start from those and adjust the parameters to suit the size of your build. The following is a summary of the most important elements for a delta printer.

- To tell RepRapFirmware that your printer is a delta and to define its parameters, put command M665 L### R### H### B### in your config.g file, where ### represents a number. The L parameter is the diagonal rod length. The R parameter is the delta radius (the radius of the circle formed by the mid points between the top ends of each pair of diagonal rods, minus the radius of the circle formed by he mid points between the bottom ends of each pair of diagonal rods). The H parameter is the height of the nozzle above the bed when the printer is homed. The B parameter is the printable radius of the bed. All these values may include decimal portions.

- You need to set up the homing commands in file sys/homedelta.g on the SD card. When configured for a delta printer, RepRapFirmware will always home all three towers when any G28 command is processed, ignoring any X, Y or Z parameters. Typical contents of homedelta.g would be:

  ```
  G91   ; use relative positioning
  G1 S1 X250 Y250 Z250 F5000  ; move all carriages up at most 250mm, stopping at the endstops
  G1 S2 X-5 Y-5 Z-5 F1000   ; move all towers down 5mm
  G1 S1 X8 Y8 Z8 F500  ; move towers slowly up 8mm, stopping at the endstops
  G90  ; back to absolute positioning
  ```

  Adjust the “250” numbers in the first line to be somewhat greater than your maximum print height, to ensure that the carriages reach the endstops. During initial testing, you may wish to reduce the homing speed value “5000” in the second line to a lower value, to give yourself more time to press the reset or power button if something goes wrong.

  The above works because when the S1 or S2 parameter is used in a G0 or G1 command, the X, Y and Z values in the command are used as values for the individual motors instead of the actual head coordinates. In the case of S1, the endstops are activated too.

- You need to use the M574 command to specify that each of the X, Y and Z towers has an endstop switch at the high end., e.g. M574 X2 Y2 Z2 S1.

- You can test the X, Y and Z tower motors individually using sequences like this:

  ```
  G91   ; relative movement
  G1 X-10 S2  ; move the X motor down 10mm
  ```

- To perform automatic delta calibration, set up a bed probe sequence in the bed.g file. See the one I provide for the Mini Kossel for an example. The S parameter in the last G30 command defines how many factors you want to adjust. You can do 3 factor calibration (endstop adjustments only), 4 factor (endstops and delta radius), 6 factor (endstops and tower positions), or 7-factor (endstops, tower positions, and diagonal rod length).

Configuring RepRapFirmware for a CoreXY printer

To switch the printer into CoreXY mode, send command M667 S1. If any motors move the wrong way, you can either turn power off and reverse the connectors, or use the M569 command to alter the directions.

You can find sample config.g and homing files in folder SD-image/sys-CoreXY. Simultaneous X and Y homing of CoreXY printers is not yet suipported.

Last updated 2015-06-10.