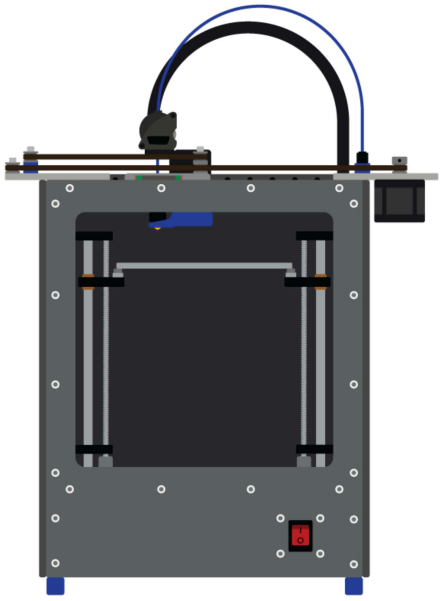
RailCore® Labs



RailCore® is a family of Core-XY based Reprap 3D printers designed by J. Steve White, Tony Akens, and Ben Withem.

View My GitHub Profile

Copying the RailCore Duet Bundle to your SD-card

1. Populate a www folder with the latest release from DWC
2. Populate sys, macros, filaments folders from the Railcore Github Config Repository

Wait a minute, there are several configs! Which config should I be using?

The RailCore project is all about flexibility and making your build your own. So, that is your choice, and down to how you want to approach your build.
RepRapFirmware 2 is the only one currently supported. RRF3 is being tested by several members.

- Filastruder config - For use with the Filastruder Kit. A good, simple base config to work from.

- Kraegars config - The original basic config aimed at self-sourcers. Can be a useful starting point for those that want to engineer everything themselves and keep all their Duet settings in a single file.
- Project R3D config - A slightly modified version of Kraegars config, for people that have purchased a Project R3D kit.
- RailCore Community Config - Originally based on Kraegars Duet config. Aimed at more advanced users.RRF2

Community based configuration information (only for more advanced users, use the Filastruder config if you are starting out with the Duet)

The “duet” folder contains a Duet configuration bundle for RepRapFirmware Duet (Ethernet/Wifi) for the RailCore II 300ZL and 300ZLT.

This bundle has been created by community members with the following goals in mind:-

1. Ease of use (where it does not conflict with other goals)
2. Standardisation of many settings across self-sourcers and kit builds.
3. Ease of upgrade of configuration, without disrupting the custom setting that make each users RailCore individual.
4. Safety during build and commissioning

In order to achieve this we have come up with the following approach, that is slightly different to the standard, monolithic Duet config.g file which each user then edits.

1. A main “config.g” file that provides the most basic, common and conservative settings for your RailCore build. This file is not intended to be edited by users, and can be updated over time from this repository to update or enable features.
2. A secondary “config-user.g” to provide the custom settings for each setup. Any setting not defined in this file will take it’s value from config.g. Any setting that is defined, will override any basic setting in “config.g” - the user is always in control.
3. The “config-user.g” has many sections commented out by default. These sections are intended to be individually enabled, tested and modified as commissioning progresses, in a safe and controlled manner.
4. Test macro(s) are available in order to test your build in a useful and standardised way during commissioning, in order to help pick up any build problems.

Usage Instructions for the community based configuration

Create your own or use an existing “config-user.something” file to base your own on,and name it “config-user.g”. Place this file in /sys to replace the existing one, which is there as a placeholder and warning system.

Examples and standard files for kits and self-builders are available, and there are also bundles for items where extra files are required (such as the BL-touch or sensorless homing) Each config-user file is enabled to allow the RailCore to be operational, and initally many conservative values will be inherited from the stock config.g To override these and unleash your Railcores’ speeds, size and/or features after building, you must during commissioning, test, measure and uncomment or modify various lines (in an organised and controlled fashion) to suit **your** RailCore build.

While we have done our best to comment the files as much as possible, we cannot put the full documentation for every command into these configuration files for obvious reasons. Do be sure to read the Duet G-code documentation on any G or M code line you are unsure of.

The included config-user.g in the /sys directory is not intended for use, and if run will stop your RailCore with a warning.

The only file you will want to change at this point is the wifi.g file in /sys if you have a DuetWifi.

After the SD card is populated with your chosen configuration

WARNING: The RailCore is a DIY project. Build safe, build smart and BE RESPONSIBLE. When in doubt, double check things. USE COMMON SENSE AND USE THESE CONFIG FILES AT YOUR OWN RISK.

It is a reasonable approach to get the Duet up and running before connecting the PSU, stepper motors, thermistors and any other parts. The Duet can run from USB power (assuming it can supply a steady 5V)

Important: Note about USB 5V power

> *When using the Duet with 5V power supplied by the USB cable only (i.e. no power supply unit – PSU – power), you may find that the board doesn’t respond correctly. The web interface, SD card reading, IR sensor may all be unreliable. Three different voltages are used on the Duet: 24V for motors, fans and heaters, 5V is used to drive the MOSFET gates, and is converted to 3.3V for most other parts of the board (the*

> *ARM processor chip, thermistors, stepper drivers, proximity sensor, SD card, USB, and Ethernet).*
>
> *It's possible that your 5V USB power source (usually your computer) doesn't supply enough voltage, or current, to power the Duet, particularly if you're working through a low quality USB power supply, an unpowered USB hub, or a laptop with low power USB ports. The Raspberry Pi, which uses a similar ARM chip, also has this problem. Plugging all the other connections in may also drain enough power to cause problems.*
>
> *The Duet will work correctly when supplied by the PSU, but don't plug this in yet. If your USB power is not good enough, then routing the USB from your computer through a powered USB hub will usually work.*

How to connect to the Duet

With luck, you should be able to connect via the web interface via http://railcore.local on your local network. This very much depends on the machines you are using and your local network topology.

If this does not work, then the next step is to interact with your Duet via USB and find its IP address. Full instructions can be found on the Duet documentation - Getting connected to your Duet

---