

JuicyBoard

An open source, modular, robotic control system for building 3D printers, CNC routers, and other robotics applications

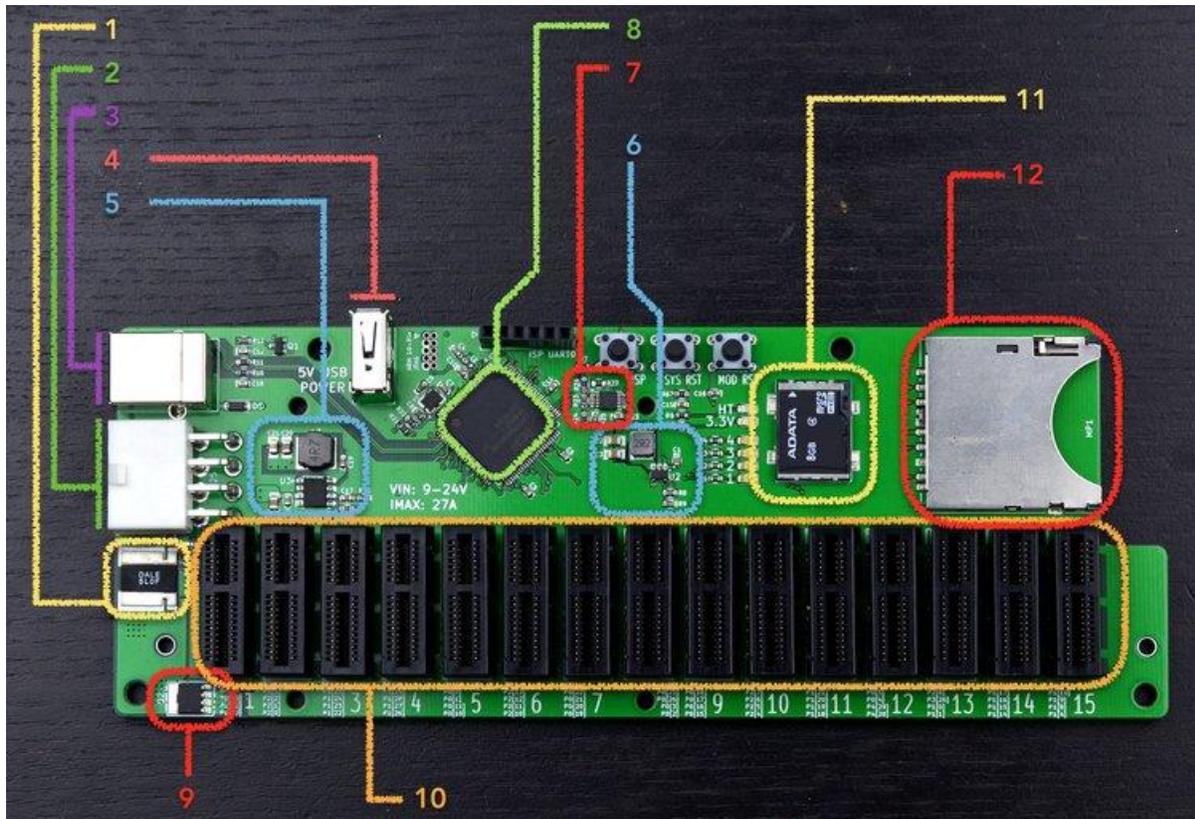
Overview

JuicyBoard is the foundation of a modular, open source platform that makes it easy for you to build a custom 3D printer, CNC router, or any other device driven by stepper motors. The idea is simple, unlike other control boards, JuicyBoard is not pre-populated with any drivers, switches, or other functions. Instead, it includes only the components for common, core functions (e.g., microcontroller, USB port, SD card, etc.), so you have control over adding just the necessary functional modules for your application. This modular approach gives the platform a number of advantages over static board designs, as we explain below. There two models of the base board:

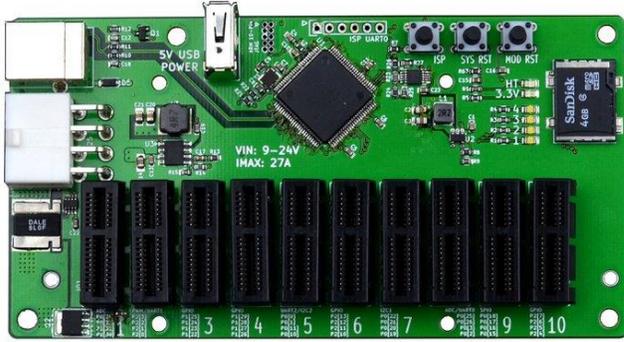
1. The R1000AX includes fifteen features slots and is 175 mm x 65.5 mm.
2. The R1000A is a more compact model that includes ten feature slots and is 125 mm x 65.5 mm.

JuicyBoard Features

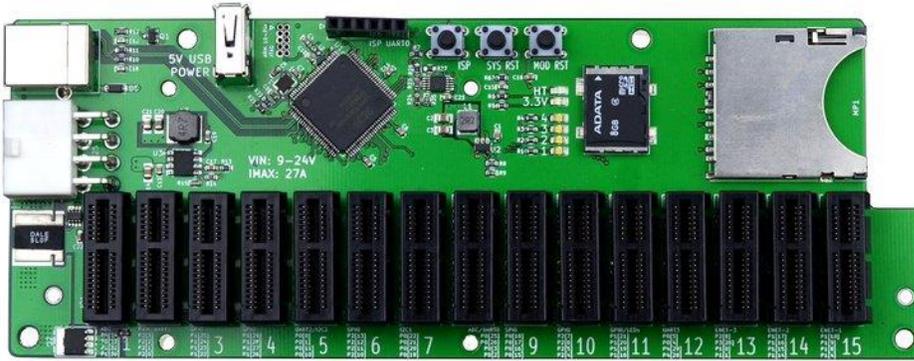
The JuicyBoard **R1000A** and **R1000AX** include the following features:



1. **Triple channel precision power monitors [R1000AX Only]** : measure voltage/current of all power domains
2. **ATX compatible input power connector (9 V ~ 24 V / 27 A)**
3. **USB Type-B data port:** main communication port
4. **USB Type-A connector for 5 V output:** power an external system like a Raspberry Pi
5. **5 V switching regulator (3 A output):** powers 5 V logic and external system
6. **3.3 V switching regulator (1.5 A output):** powers 3.3 V logic
7. **LM5060 hot swap controller [R1000AX Only]** : protects system from overshoots, undershoots, and overloading
8. **LPC1769 ARM Cortex M3 MCU @ 120MHz**
9. **100A PSMN1R2-30YLC, 115 NFET Switch [R1000AX Only]** : used for shutting down the system in case of faults or emergencies
10. **Feature Slots:** 15 for R1000AX and 10 for R1000A
11. **Micro SD Card Socket [R1000AX Only]**
12. **Full Size SD Card Socket [R1000AX Only]**



125 mm x 65.5mm



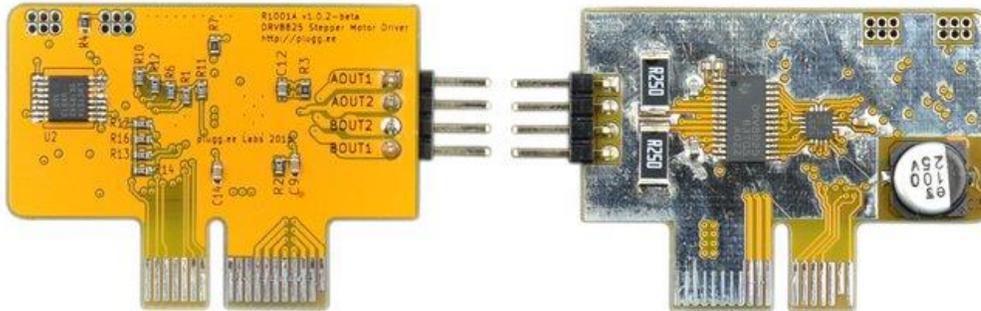
175 mm x 65.5 mm

Quick Comparison between R1000A and R1000AX

	R1000AX	R1000A
Number of Feature Slots	15	10
On-board Precision Power Monitoring	YES	NO
Electronic Fuse, Kill Switch	YES	NO
Undervoltage/Overvoltage/Overcurrent System Protection	YES	NO
Can Support Ethernet PHY Modules	YES	NO
Dimensions	175 mm x 65.5 mm	125 mm x 65.5 mm

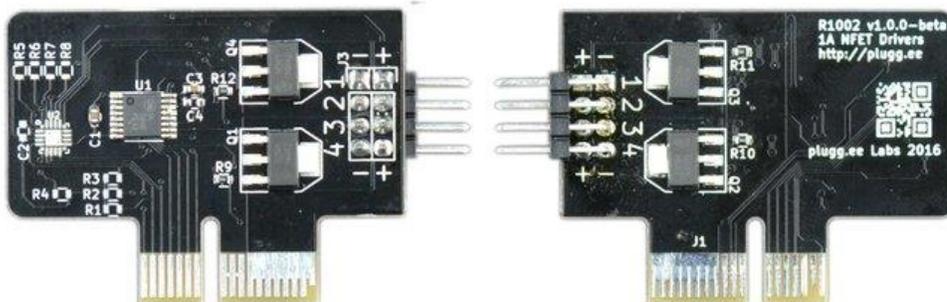
Available Modules

R1001: Smart Stepper Motor Driver



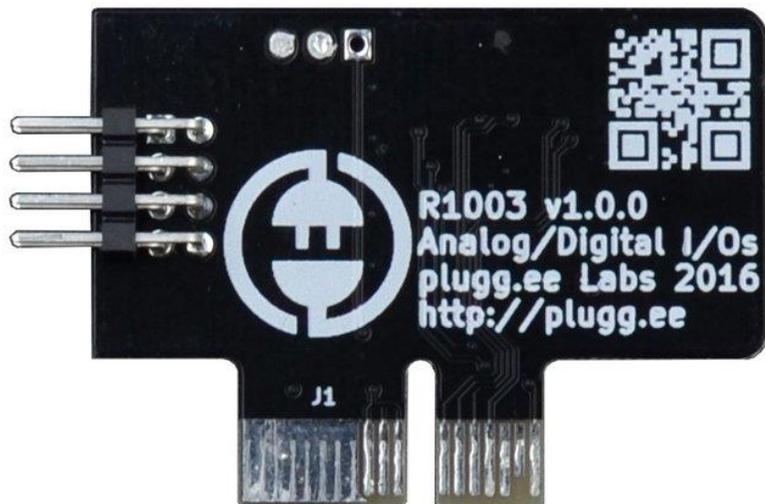
This is a NEMA stepper motor driver based on TI's DRV8825 chip. It's a high-performance stepper driver capable of micro-stepping at 1/32" resolution. Current can be digitally set for every driver in 10 mA increments. The R1001 includes temperature/voltage/current monitors for realtime performance measurement and fault detection.

R1002: Quad 1A DC Low Side NFET Control Switches



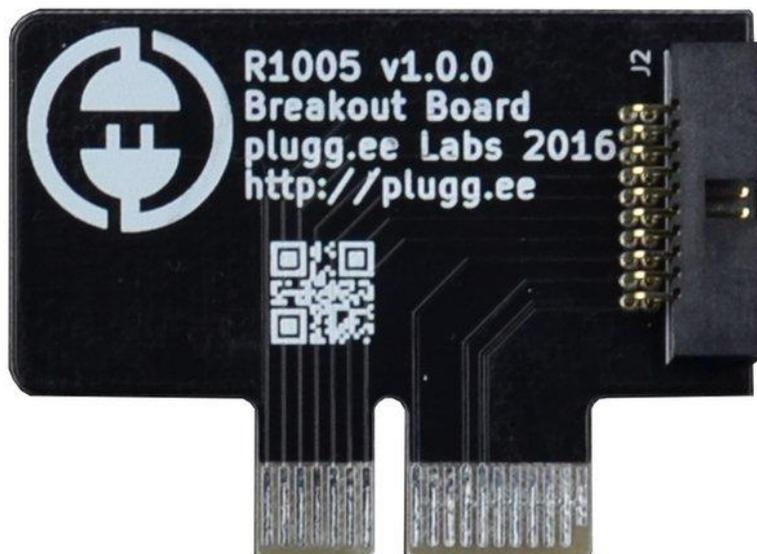
This module can control four small functions (e.g., fans, LEDs) that consume less than 1 A and run at the high input voltage.

R1003: Quad Analog/Digital I/Os



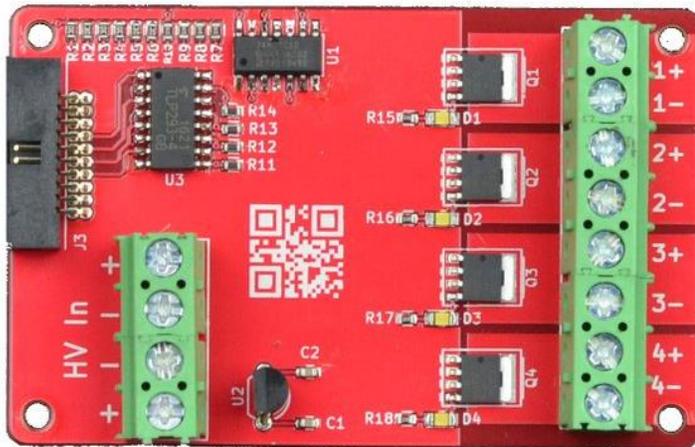
This module connects four microcontroller signals to any external component (such as a thermistor or end-stop switch). It can be used for analog and digital signals.

R1005: General Purpose Breakout/Extension



This breakout board allows for use of a 20-pin ribbon cable to connect all power, I2C, and microcontroller signals to external modules that don't fit in a slot. It's necessary for modules like the R1007.

R1007: Quad 20A DC Low Side NFET Control Switches



This module can drive four heaters with strong **100A PSMN1R2-30YLC,115 NFETs** from NXP, which allows it to run cool, steady 300+ watt heat beds. The R1007 must be powered through its own auxiliary supply input. It is optically isolated from JuicyBoard (meaning, it can have its own separate power and ground, and can be powered by up to 24 V).

R1008: Dual Channel Platinum RTD PT100 Temperature Sensors



This module provides an interface to two PT100 platinum resistor temperature detectors (RTDs) for precision temperature sensing (as accurate as less than 0.2° C error). It can work in 2-wire, 3-wire or 4-wire configurations.

Other Modules In the Pipeline

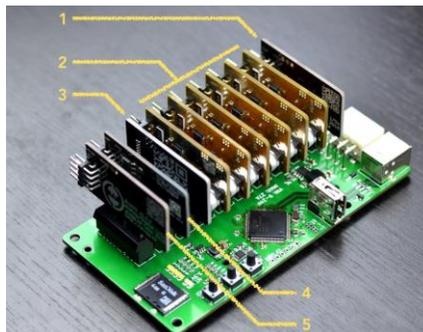
We'd like to hear your feedback and suggestions for new modules. This is our current wish list:

- Optically isolated Quad Channel AC PWM power control with zero-crossing capability, for controlling AC heat elements, which is much more efficient for large and strong heaters
- Optically isolated AC spindle control
- Particle.io Photon/P1 housing module, to wirelessly connect Juicyboard to the internet

We're very open to working with other people who have cool modules that can work nicely with JuicyBoard, lately we spoke with **ODrive team**, a very interesting brushless motor driver with position control that can interface with JuicyBoard through CAN. We're planning to write **Juicyware** drivers that allows using brushless DC servos in your application! Check out this cool video of **ODrive** actuating XY axes of a router.

Use Cases

So far, we've used JuicyBoard to build a couple of 3D printers and a CNC machine. Here's a case where we used it to build a system for controlling a triple extruder, triple-head 3D printer (using [polystrooder](#)) that can print three different materials. Here is how we configured a 10-slot prototype JuicyBoard to drive this printer:



1. **R1003:** wired as analog inputs to read the temperature through thermistors, three hot-ends and one heated bed.
2. **R1001:** 6x stepper motor drivers, three are actuating the motions axes and the other three are actuating the extruders.
3. **R1005:** this connects to an R1007 module that is independently powered to switch the hot-end and hot-bed heaters.
4. **R1002:** we used one channel to switch the layer fan.
5. **R1003:** Wired as digital inputs, three channels were used to read inputs from three end-stop switches, the fourth was used to read from a z-probe (this helps a lot with bed leveling).

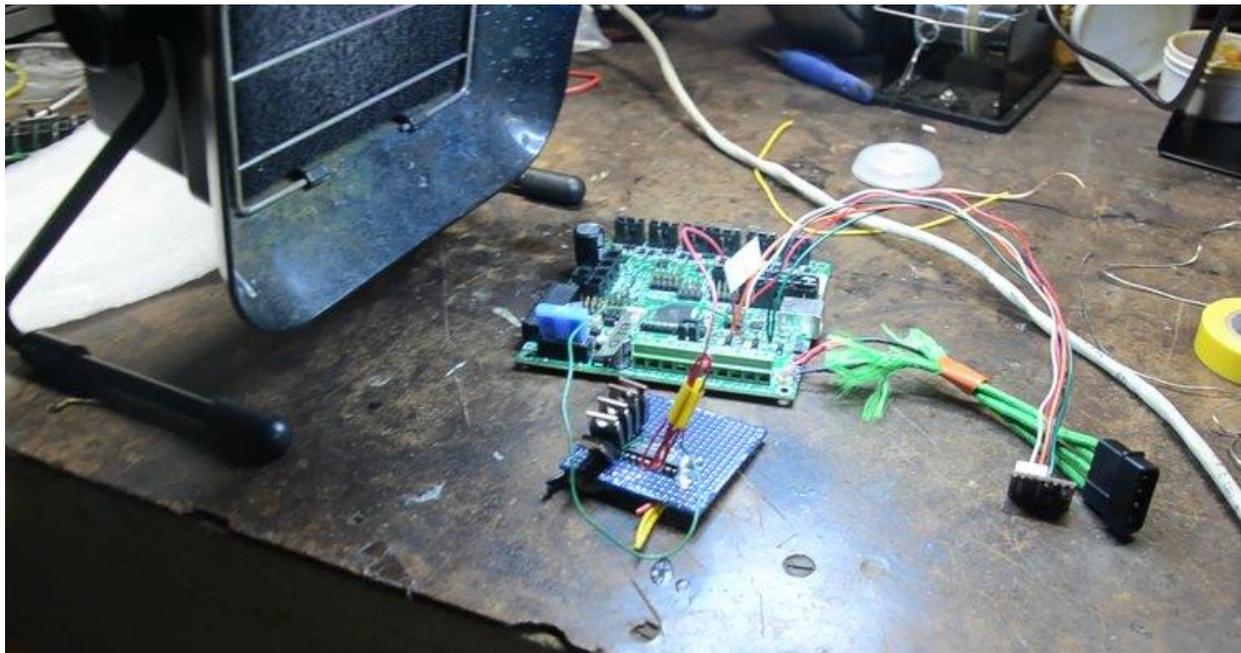
R1008 is still new, we're planning to upgrade the machine with PT100 sensors instead of thermistors. It will take some extra machining as well.

Check out our video above for more applications. We'll be releasing updates illustrating other applications as well, so make sure you're signed up.

History and Purpose

The idea first came up after we attempted to build a custom 3D printer. At first, we tried several off-the-shelf boards, but these caused several points of frustration:

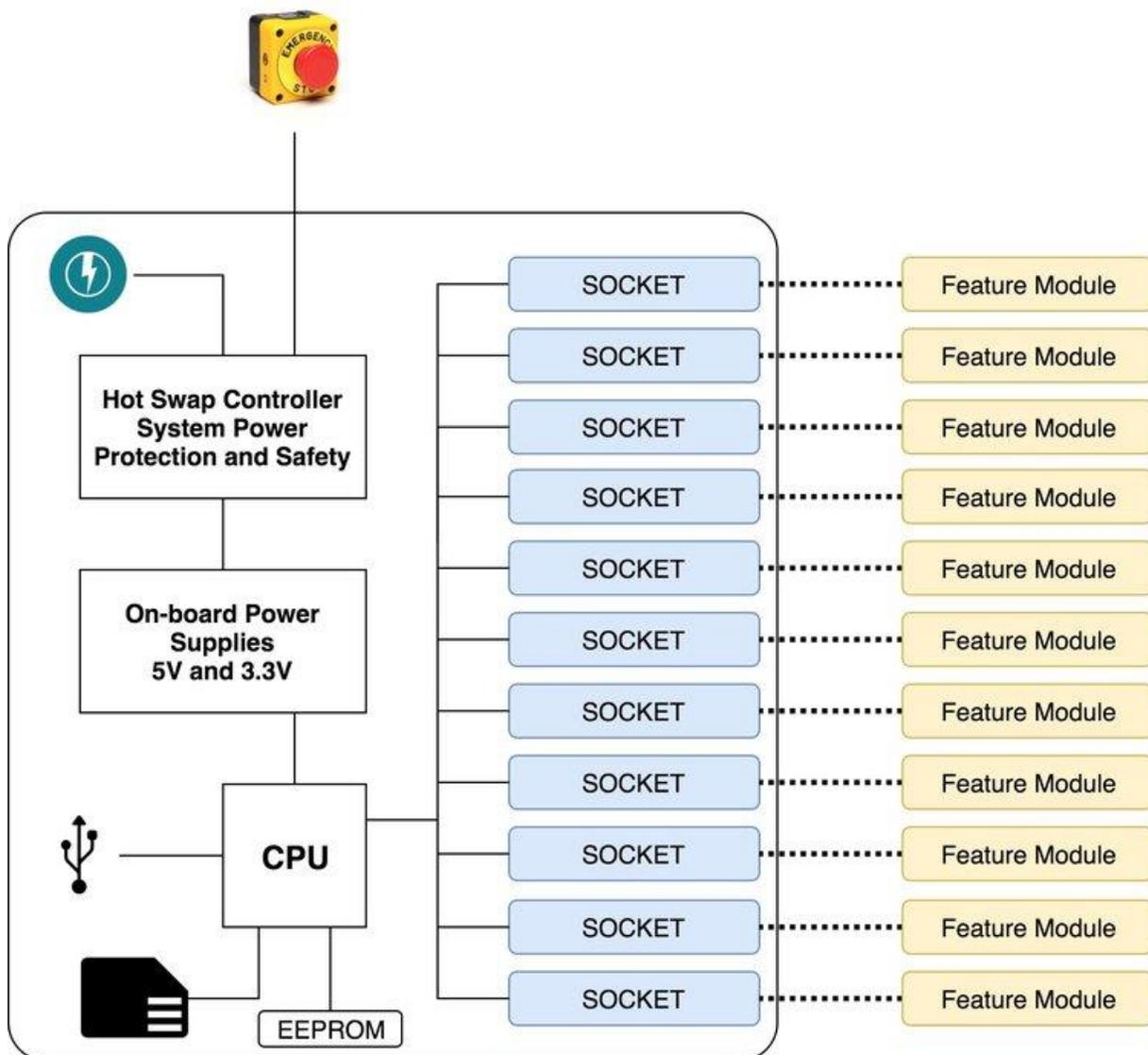
- Since our printer was custom, the available boards did not include all the features that we needed.
- As a result, we had to patch the setup and solder external modules to complement the board functions. Since everything did not optimally fit together, mechanically speaking, the system ended up as an unreliable hack of wires and boards.
- We also could not use all the features on the board, which was a waste of resources.
- Because the setup was experimental, as we were coding our custom functions the first board failed. It took a lot of effort, and burning two more boards to figure out what went wrong. Debugging a single complex board with everything on it is not an easy task.
- None of the boards we found included any serious system protection mechanisms that can detect power problems (like supply over-shoot or under-shoot, or on-board short circuiting) and shut down the system. That can cause damage to all the other expensive system components (motors, power supplies, other logic components).



Hacking an off-the-shelf board

To solve these problems, we came up with a completely different architecture and approach

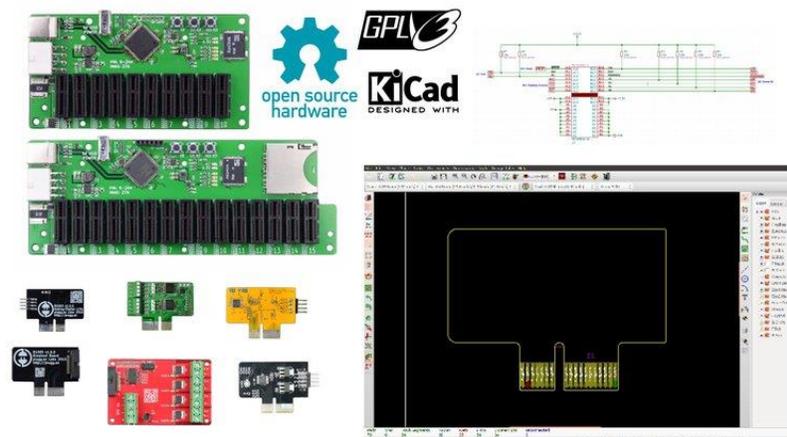
- We created a platform that has a base board (JuicyBoard) that includes all the essential basic functions of every system (the main processor, power supplies, communication port and flash storage). Then we arranged a set of sockets that allow users to plug in modules that perform only the features they need in a given application.
- We designed the system to be as flexible as possible. For example, a stepper motor driver can be plugged into any slot. However, some special function modules (like the R1005 Analog Inputs) must be plugged into specific slots that connect to the CPU's analog pins.



- We added a full suite of telemetry functions (measuring voltage, current, and temperature) to the base board and to some modules. That lets users monitor the health of the system in real time, so they can detect and isolate failures. It's easy to replace the faulty modules without reworking the system.
- We also added in a hot swap controller, which acts like an electronic fuse that detects abnormal power supply conditions (like over-voltage, under-voltage or current overloading) and shuts down the system. The controller has an input for attaching a **mechanical kill** switch, which is the most reliable way to shut down the system in case of emergencies.
- **Another benefit** we figured out after working with JuicyBoard for awhile is the ability to upgrade the system with minimum reworking of hardware and firmware. For example, say we build a 3D printer with NEMA 17 motors and we later want to make a mega-size of the same 3D printer (say with NEMA 34 steppers). With JuicyBoard, we only need to design a larger motor driver module that accepts the same input control as R1001, then we can upgrade the system with the new motors with minimal revisions to firmware. Similarly, users could upgrade their system with a later revision of JuicyBoard that has an updated CPU or power capabilities, while keeping their add-on modules largely unchanged.

Open Source and Future Expandability

We love open source. A fundamental tenet of our philosophy is a belief in the power of the community to drive improvements of any designs or products out there. We're exclusively using open source PCB design tools to create our boards. And, we've decided to open source all JuicyBoard and module hardware and firmware under GPL v3.0, so others can further evolve the platform and possibly enrich it with modules we've never thought of before. To that end, we have created a template KiCAD PCB project that's an easy starting point for anyone who wants to design and build their own module.



Firmware

JuicyBoard runs Juicyware, which is a fork of Smoothieware and is open sourced under GPL v3.0. Similar to our hardware architecture, Smoothieware is modularly structured firmware that allows adding and modifying functions easily and efficiently. It's based on ARM's mbed library, which means it has a path to portability to other microcontrollers in the future. Juicyware can be found in [GitHub](#).

Risks and Challenges

We have been through several design iterations of JuicyBoard, since we started working on it in December 2015. The first revision was shown in the Bay Area 2016 Makerfaire, where it successfully ran a couple of 3D printers.



Early prototype

We've learned a lot since then and incorporated all the feedback we've gotten over four design cycles until we reached a point where all the problems are addressed (mechanical, thermal, and electrical). We will be releasing the final revision gerber files (which include very minor tweaking and final silkscreen art) in a couple of weeks and will verify the revision before pushing the button on production.