# Patchwork: Multi-User Network Control of a Massive Modular Synthesizer


Brian Mayton
MIT Media Lab
75 Amherst St.
Cambridge, MA 02139
bmayton@media.mit.edu

Gershon Dublon
MIT Media Lab
75 Amherst St.
Cambridge, MA 02139
gershon@media.mit.edu

Nicholas Joliat
MIT Media Lab
75 Amherst St.
Cambridge, MA 02139
njoliat@media.mit.edu

Joseph A. Paradiso
MIT Media Lab
75 Amherst St.
Cambridge, MA 02139
joep@media.mit.edu



## ABSTRACT

We present Patchwork, a networked synthesizer module with tightly coupled web browser and tangible interfaces. Patchwork connects to a pre-existing modular synthesizer using the emerging cross-platform HTML5 WebSocket standard to enable low-latency, high-bandwidth, concurrent control of analog signals by multiple users. Online users control physical outputs on a custom-designed cabinet that reflects their activity through a combination of motorized knobs and LEDs, and streams the resultant audio. In a typical installation, a composer creates a complex physical patch on the modular synth that exposes a set of analog and digital parameters (knobs, buttons, toggles, and triggers) to the web-enabled cabinet. Both physically present and online audiences can control those parameters, simultaneously seeing and hearing the results of each other's actions. By enabling collaborative interaction with a massive analog synthesizer, Patchwork brings a broad audience closer to a rare and historically important instrument. Patchwork is available online at `http://synth.media.mit.edu`.



## Keywords

Modular synthesizer, HTML5, tangible interface, collaborative musical instrument


## 1. INTRODUCTION

Nearly every computer musician has interacted with some form of modular software synthesizer, but relatively few have had the opportunity to patch on a large, tangible analog synth. Modular synths are bulky, expensive, and exceedingly rare, confined for the most part to music departments and museums, or in the hands of a small number of enthusiasts and composers. Yet these instruments are increasingly sought after by musicians looking for tangible controls and a classic, gritty analog sound, and continue to fascinate audiences when displayed publicly. At the same time, it is difficult to create installations that allow the general public to directly manipulate or patch these instruments; a steep learning curve and delicate controls mean that visitors can watch and listen, but they can't touch. We encountered that particular issue when we were invited to install our massive modular synth [5], one of the world's largest, at a local museum.



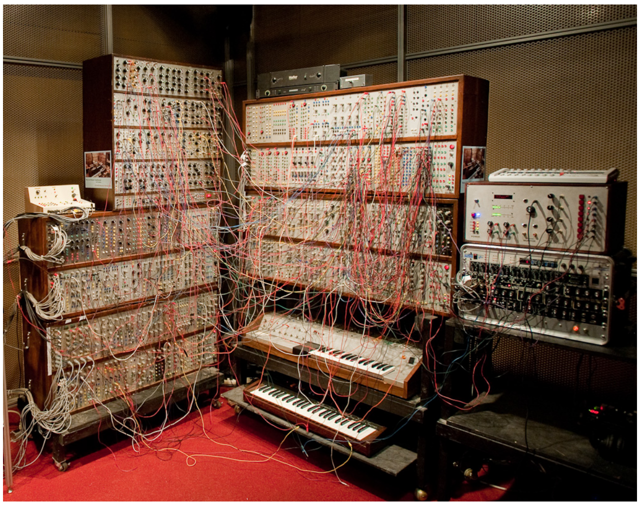

**Figure 1: Paradiso's modular analog synthesizer.**

Built by Dr. Joseph A. Paradiso throughout the 1970s and 80s, our synth consists of five large cabinets that together contain nearly 200 homemade modules, a number of home-built and custom-modified keyboard interfaces, and a pair of substantially augmented and fully integrated Moog synths. To date, Paradiso has been its only serious composer and player. In a typical installation, he assembles a complex patch over many hours, creating an evolving and often unpredictable, autonomous soundscape.

The synth has attracted a fair amount of public interest over its lifetime, and has recently enjoyed renewed attention. Recently, we created an audio stream that allows an online audience to listen to a stereo mix of the synth's output in real-time. As an online community formed around the stream, we created a Twitter feed that would provide regular listeners with news and updates, announcing new patches and live performances. Still, despite all the interest from onsite and online audiences, the synth remained a non-interactive instrument.

To address these concerns, and to bring the analog synthesizer closer to a broad worldwide audience, we developed

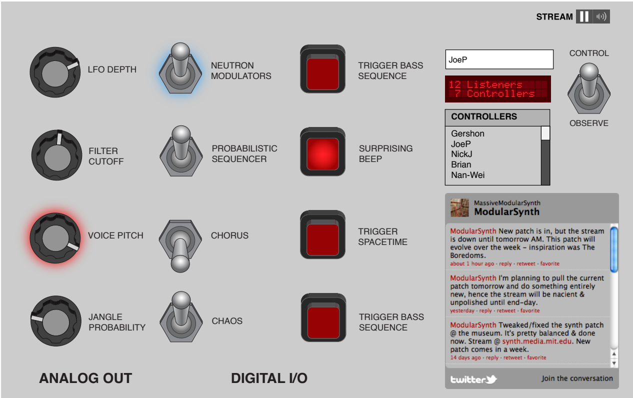


**Figure 2: The Patchwork web client: the knob highlighted in red is being controlled by a remote user, while a toggle switch is highlighted in blue as the local user hovers over it.**

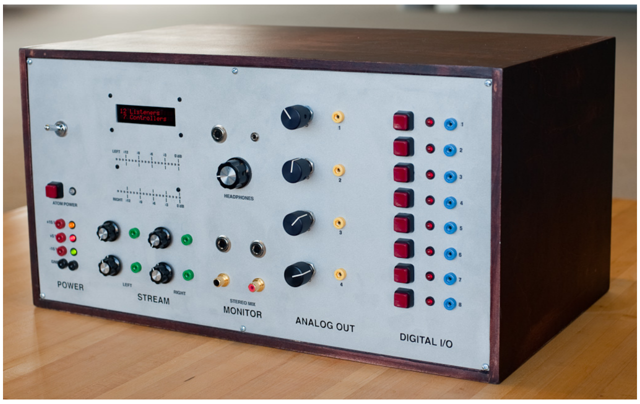


**Figure 3: The Patchwork module**

Patchwork, a networked synth module with tightly coupled web browser and tangible interfaces. Patchwork allows online and physical audiences to concurrently control motorized knobs, buttons, and switches to produce analog signal outputs from a physical synthesizer module. Designed to match the aesthetics of Paradiso's original synth cabinets, our new, web-enabled cabinet uses an embedded computer and custom hardware to drive the analog and digital outputs. The cabinet also contains a custom-designed mixer and audio interface to produce a live audio stream that is fed back to online users, who can also see each other's activity in the browser interface. Visitors to the physical installation see the moving knobs and flashing LEDs, hear the effects of remote users' actions, and can use their smartphones to simultaneously control the module.

A web server keeps track of each client, allowing users (both physical and remote) to take momentary control of individual knobs and buttons. To support this kind of cross-platform, multi-user interaction, we use the emerging HTML5 WebSocket standard and the jQuery JavaScript library, enabling low-latency, high-bandwidth control from nearly any modern computing device.

## 1.1 Related Work

As far back as the 1970s, composers and researchers were developing networked music systems that allowed multiple players on separate instruments to collaborate in performance [1]. The Reactable*, for example, allows physically present and remote users to collaborate on one or more tangible tabletop instruments [3, 4]. In [6], Weinberg develops a rigorous framework for the field, which he calls interconnected musical networks. Patchwork extends multiple virtual interfaces into the physical world with actuated controls, positioning it in a small category of single instruments that support multiple remote players. In some ways, our system is similar to performing robots like Hoffman and Weinberg's marimba player [2], in that it brings an existing, physical instrument onto the internet; Patchwork is unique in that it allows remote audiences to interact directly with a one-of-a-kind instrument and its creator-composer. Our instrument also evolves over time, with new patches and additions, and Patchwork adapts to those changing conditions.

The Patchwork module is not the first digital controller for an analog synth. Traditionally, Control Voltage/Gate (CV/Gate) and MIDI-to-Control-Voltage devices have allowed performers to bring older analog synthesizers into the digital domain, but these devices are designed for single-user, local control, and don't integrate aesthetically into the synth or connect to the web like Patchwork does. MOTU's Volta software plugin also allows users to drive control voltages through their digital audio workstation (DAW), but it too represents only one part of the Patchwork system. More than anything, its recent release is an indication that consumer interest in analog synthesis is still in resurgence. Patchwork capitalizes on that interest, bringing users together to collaboratively control live-generated music through a shared physical resource.

# 2. SYSTEM DESIGN

The Patchwork system consists of 3 parts: an HTML5 interface for the browser, a web server, and a wooden synth cabinet containing a low-power embedded computer and a custom printed circuit board (PCB). The components are designed to be modular and easily configurable; for each new patch, the composer edits a configuration file that sets the numbers and types of each input and output, as well as the analog voltage ranges on the outputs. Each component reads that file from the server and sets itself up accordingly. Because the physical knob positions are read by the microcontroller in the cabinet, they can be decoupled from the analog output; their ranges are scaled in software to maximize the control resolution over the range given in the configuration file. The system can also be switched into patching mode, in which the web interface becomes read-only and the composer is given the full voltage range.

## 2.1 A Synth Interface for the Web

Patchwork provides a web browser-based interface to remote users, shown in Figure 2. The web client is built using HTML5 and the new WebSocket standard and is therefore available via any modern browser, including for mobile phone and tablet users. The sockets are used for low-latency interaction, so that controlling users can achieve near-seamless control over the synth, and so that other users can see the synth module updating in real time.

The web client is configurable such that the composer using the modular synth can simply specify in a configuration file which, and how many, control components should be exposed on the web interface, selecting among knobs, keyboards, buttons, switches, and triggers.

The web client interaction allows for fine-grained concurrent control. Our server mediates between clients to assign control of components like knobs and buttons, which can only be directly controlled by one user at a time. When a user tries to control a knob, the client requests ownership from the server. Once the user is explicitly handed control of a component, other users can see the component updating in real time, and are not able to control it. When multiple users try to take control of a single component at the same time, the web client does not update the view until the server has acknowledged one user's ownership; in this way we avoid giving misleading visual feedback to the other user. Once the server has acknowledged ownership, the client can update the component without waiting for more server acknowledgments, thereby maintaining low-latency UI responsiveness.

## 2.2 The Patchwork Server

The Patchwork server mediates between online clients, who connect through their browsers using the HTML5 WebSocket standard, and the Patchwork synth module, which drives the physical hardware and I/O. The server system can be configured in two ways: for small numbers of clients, the low-power computer in the cabinet can act as the sole server and controller; to support large numbers of clients, an off-site middleware server acts as a buffer between those users and the cabinet controller, managing the client list and assignments, and relaying control messages to the cabinet over a socket.

Our web server design is configurable and extensible, in order to support different synthesizer configurations and back-end implementations. This means that while working on a patch, the composer can, without editing any code, choose how the I/O maps to different behaviors (e.g. toggles or triggers) and front-ends (e.g. knobs or keyboard interfaces in the browser), and how the voltages scale with control parameters. Switching to a new backend (e.g. from the embedded server to the larger web server with a network backend) can be done by implementing a new backend subclass and without editing the existing framework.

The Patchwork web server mediates between concurrent controlling users. To facilitate effective collaborative control, we allow a small set of users to control the synth at a given time, while maintaining a queue of users who have requested to control it. The web server maintains this information in shared data structures, and we use the Autobahn and Twisted Matrix Python web frameworks to ensure serial accesses of these structures. Users who request control may enter their names, so that non-controlling observers can see who is controlling the instrument.

## 2.3 The Patchwork Synth Module

The Patchwork module (Figure 3) is built around a low-power embedded Intel Atom board running Linux and our server software. Input and output functionality is handled by a custom I/O board (Figure 4) that interfaces with the Atom over USB. It is housed in a wooden cabinet with a front panel that aesthetically matches the existing synthesizer.

Four analog outputs allow control of continuous parameters in the patch. Each one is controlled both by a physical knob on the front panel of the module and a virtual knob on the web interface; turning either adjusts the output. The potentiometers on the physical panel are motorized to reflect changes from web users. The output voltage range can be mapped anywhere within $\pm 10$ volt rails, enabling the composer to constrain the parameter to appropriate values. Each output can be separately configured; for example, one could be driven from a virtual keyboard in the web interface and output 1 volt/octave to drive a voltage-controlled oscillator (VCO), while another might output a linear voltage to control the gain of a voltage-controlled amplifier (VCA). Voltage steps can be arbitrarily quantized via lookup tables, enabling an output to play specific notes on a given scale, for example.

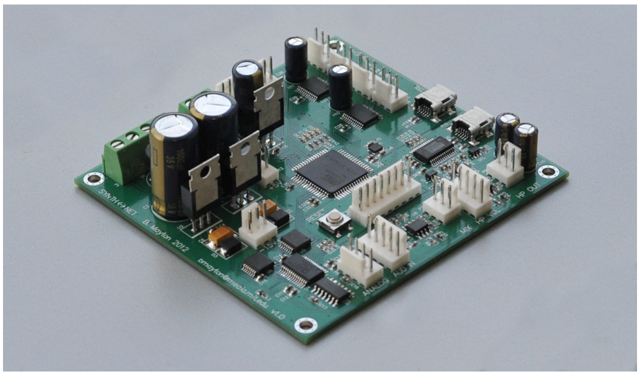

**Figure 4: The custom I/O board.**

Eight digital ports can be configured as either inputs or outputs. As inputs, control signals in the synthesizer patch can be visualized on the web interface. As outputs, they can be configured as triggers (generating a short pulse when pressed), momentary pushbuttons (on as long as the button is held down), or toggles (switched between on/off each time the button is pressed), with an appropriate representation shown in the web interface for each mode.

The module also accepts quad-channel audio input which is mixed down to stereo for the live audio stream to the internet. The levels of each channel can be individually adjusted by the composer to create the stereo mix. A stereo VU meter on the front panel both aids in setting the levels and provides an eye-catching visual representation of the audio to exhibit visitors. The audio is digitized by a USB audio codec on the I/O board and encoded to MP3 and Ogg Vorbis by the Atom for live streaming to the internet.

# 3. FUTURE WORK

Patchwork is new to us and the synth community, and we have only begun to explore the space of applications for it. Still, there are a number of concrete steps and novel applications we are planning for the near future. First, we are developing a user study to investigate the effect of Patchwork on both the online and onsite audiences' understanding of and engagement with the synth. Does an online user feel differently about controlling a remote instrument over a software synth, even if they can't see the physical device? Audiences tend to have trouble understanding how complex, evolving behaviors can be produced by the interplay of signals generated by parallel, independent modules. Does individual control over a set of signal parameters improve audiences' understanding of those analog mechanics?

We are also developing a number of new applications for the Patchwork server. In live performance, for example, Patchwork can be used in conjunction with the synth to allow audiences or band members to patch outboard instruments through live effects, using mobile devices to control the assignments. In our experience performing as a band with the synth, it can be challenging to coordinate parameters like key and tempo with the keyboardist-patcher; as a complex system of parallel modules, the synth tends to have a lot of inertia. In this context, Patchwork could be used by band members to tune parameters from across the stage without interrupting the performance.

Patchwork's open-ended design allows it to interface to almost any device. We envision smartphone applications that would map internal sensors to synth parameters, or turn touchscreens into portable dynamic-phrase synth con-

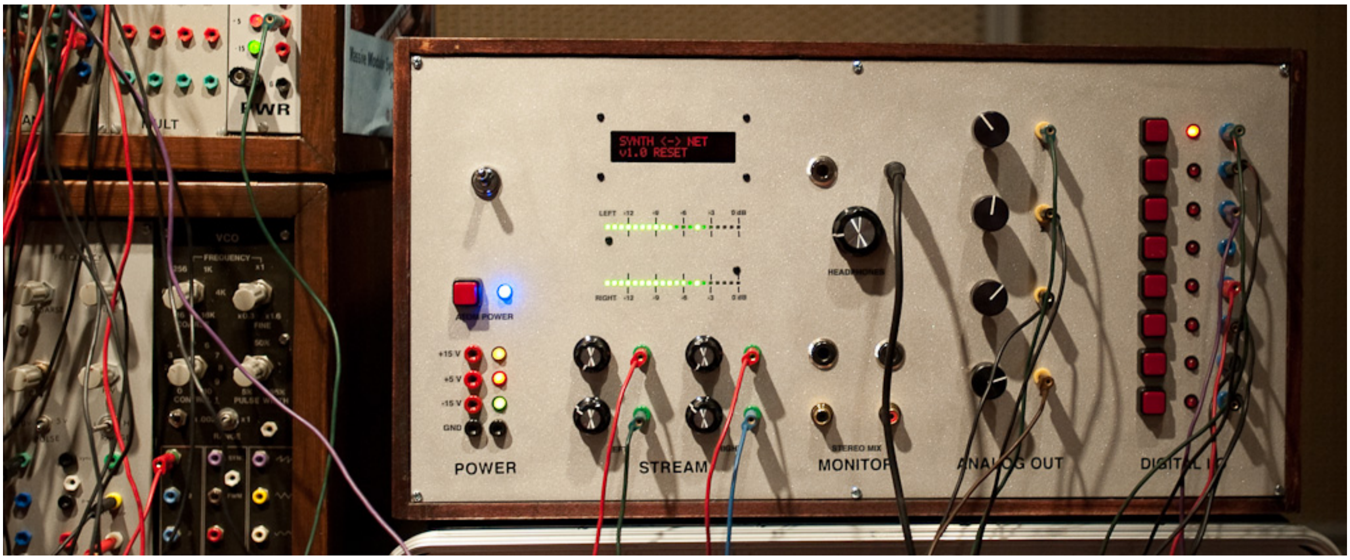


**Figure 5: The Patchwork module integrated into the synth patch.**

trollers, mimicking Korg's Kaosillator on an analog instrument. Patchwork could serve as a bridge to outboard sensors of all kinds, or feed data from the web, like weather patterns, into the physical patch.

On the client side, we are developing a JavaScript keyboard interface that allows multiple online users to play synth voices. We are also interested in using Patchwork to further develop the online community around the synth, using Twitter and other means to broaden audiences' understanding and engagement. Finally, for the module itself, we imagine a family of USB peripherals that would support interactive installations. A camera would allow us to track audiences and performers using the embedded computer. A new, smaller cabinet could perform crosspoint switching controlled over the web, exposing a simple patch bay on the front and its own USB interface on the back.

## 4. CONCLUSIONS

We created Patchwork, an extension to a one-of-a-kind massive modular synth that seamlessly brings the analog instrument into the digital, networked domain, while integrating functionally and aesthetically into the original. Our system engages a global community of synth enthusiasts, many of whom have never seen, much less interacted with anything like Paradiso's instrument. It also allows physical audiences to engage with the synthesizer in a much more direct way than they typically can, and composers to patch in dialogue with audiences. The Patchwork web interface serves as a hub for the growing online community that has formed around the synth's live stream, and its cross-platform HTML5 design means that it can run on nearly any modern computing device. As a system of easily configurable components, Patchwork opens up a world of applications that extend the reach of the massive modular synthesizer without compromising its unique sound, tangible interface, or aesthetics, bringing the historically important instrument to an eager, global audience.

Patchwork is available online at `http://synth.media.mit.edu`.

## 5. ACKNOWLEDGMENTS

Thanks to the MIT Museum for presenting the synthesizer and providing a platform for engaging the public with this work.

## 6. REFERENCES

[1] Á. Barbosa. Displaced soundscapes: A survey of network systems for music and sonic art creation. *Leonardo Music Journal*, pages 53–59, 2003.

[2] G. Hoffman and G. Weinberg. Shimon: An interactive improvisational robotic marimba player. *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, pages 3097–3102, 2010.

[3] S. Jorda, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146. ACM, 2007.

[4] M. Kaltenbrunner, S. Jorda, G. Geiger, and M. Alonso. The reactable*: A collaborative musical instrument. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE'06. 15th IEEE International Workshops on*, pages 406–411. IEEE, 2006.

[5] J. Paradiso. Modular Synthesizer. *Timeshift: Proceedings of Ars Electronica*, pages 364–370, 2004.

[6] G. Weinberg. Interconnected musical networks: Toward a theoretical framework. *Computer Music Journal*, 29(2):23–39, 2005.