BendIt.io: A System for Networked Performance of Circuit-Bent Devices

Anthony T. Marasco
School of Music and CCT
Louisiana State University
Baton Rouge, LA, USA
amarasco@lsu.edu

Edgar Berdahl
School of Music and CCT
Louisiana State University
Baton Rouge, LA, USA
edgarberdahl@lsu.edu

Jesse Allison
School of Music and CCT
Louisiana State University
Baton Rouge, LA, USA
jtallison@lsu.edu

ABSTRACT

BendIt.io is a system that allows for wireless, networked performance of circuit-bent devices, giving artists a new outlet for performing with repurposed technology. In a typical setup, a user pre-bends a device using the BendIt.io board as an intermediary, replacing physical switches and potentiometers with the board's reed relays, motor driver, and digital potentiometer signals.

BendIt.io brings the networking techniques of distributed music performances to the hardware hacking realm, opening the door for creative implementation of multiple circuit-bent devices in audiovisual experiences. Consisting of a Wi-Fi-enabled I/O board and a Node-based networking server, BendIt.io provides performers with a wide option of digital performance interfaces for controlling their hacked devices. Moreover, the BendIt.io system is user-friendly, low-cost, and modular, making it a flexible toolset for artists of diverse experience levels.

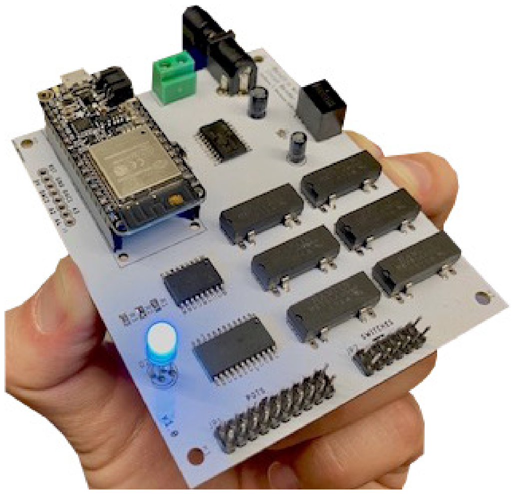

Figure 1: The BendIt.io board.

Author Keywords

NIME; circuit bending; networked performance; hardware; data-driven performance; servers; hardware hacking; smart musical instruments; IoMusT

CCS Concepts

•Hardware → Networking hardware; •Applied computing → Sound and music computing; Performing arts;

1. INTRODUCTION

1.1 Circuit Bending as Musical Practice

Defined as an improvisational approach to exploring electronics through hardware hacking by Reed Gazala [5], the aesthetic practice of circuit bending has blossomed into a well studied and well practiced art form. Numerous studies and books, both educational [3] and theoretical [13], have distilled circuit bending as a practice mixing pre-planned preparation and discovery with exploratory performance. Musically, circuit bending results in the creation of customized, avant-garde instruments made from typically non-performative electronic devices. In performance, simple physical interactions with these modified devices (flipping switches, pressing buttons, turning dials, etc.) generate unpredictable sonic results through real-time modification of a device's internal circuitry. From an aesthetic perspective, the act of performing with hacked hardware draws connections to the Upcycling movement—the act of artistically and practicality repurposing items or materials into new, more purpose-driven objects—as well as highlighting a nostalgic connection between the audience, performer, and the device.

As circuit-bending practices evolve into the 21st century, artists have begun to experiment with new directions to take their practice in the pursuit of overcoming some of the artform's inherent limitations. Circuit bending ensembles such as Oval and Loud Objects often perform with numerous hacked devices simultaneously shared between each performer in order to draw from a wider range of sonic palettes. This increase in the amount of devices in a performance does provide a roadblock for solo performers, where physical interactions with multiple devices are limited to what can be enacted with just two hands. Other artists have begun to experiment with new methods of interfacing with their creations in order to extend and/or mediate their performative gestures. Sam Battle (a UK-based artist performing under the moniker Look Mum No Computer), adds control-voltage inputs to his hacked GameBoys and Furbies in order to perform large quantities of these devices from a modular synthesizer and a piano keyboard respectively.

1.2 Networked Music Performance & IoMusT

While exploring new directions for performing with circuit bent instruments, we looked to the world of networked and distributed performance systems for inspiration. Networked

NIME'19, June 3-6, 2019, Federal University of Rio Grande do Sul, Porto Alegre, Brazil.

music performance (NMP) describes a series of tools that allow for collaborative music-making among many users on a shared, telecommunicative network [4]. Recent advances made to designing audience-performer networking topologies and data latency issues [6] have provided artists with new methods of incorporating mediated interaction into their works.

A related field of research known as the Internet of Musical Things (IoMusT) centers around the addition of augmented and Smart Musical Instruments to the participatory, data-driven performance environment created by NMP [15]. While NMP technologies allow for a bevy of Internet-enabled computer devices to be utilized for musical performance, the IoMusT field in particular opens the door for new and unconventional devices to be linked together through a central server, allowing for collaborative performance between disparate devices [14].

1.3 BendIt.io

The musical practice of circuit bending combined with the mediation techniques described above led us to devise a system that would embrace the eclectic mixture of potential sound-making devices implored by hardware-hacking artists and allow them to augment their circuit-bent instruments with an intermediary circuit to enable remote performance. Furthermore, coupling this new circuit with software-based solutions would be the key to bringing new and creative methods of mediated interactivity to the performance of hacked devices. Just as adding MIDI I/O to the organ opened the door for local network control and mediation of an instrument's physical, mechanical systems, artists can turn any device into a Smart Musical Instrument by hacking the BendIt.io system into their creative practice. The user starts by choosing a device to hack and plans their intended patch points to "bend." In lieu of adding physical switches, buttons, or dials to a device, soldering the bare-wired ends of each cable to the intended patch points allows the I/O board to patch or unpatch these points remotely.

The coupled I/O board (see Figure 1) and server software allow for performing hacked hardware in collaborative and generative compositions by syncing devices in a number of network configurations. The programming interface is accessible by every device on the network, which means that BendIt.io boards can communicate with other BendIt.io boards, multiple audience members or a single performer can communicate to any number of BendIt.io boards, and multimedia elements (visualizations, live data streams, social media feeds, etc.) can interact with BendIt.io-enabled devices (see Figure 2). The following sections of this paper provide a description of the BendIt.io system components, preliminary tests of the system in a performance setting, a speed and response test, and examples of possible interaction interfaces and applications.[1]

2. RELATED WORK

Over the past two decades, a handful of hybrid digital and analog systems have been developed to allow users to interact with hardware. One of the earliest relevant examples was the ICube System (which later morphed into the ICube X) by Axel Mulder [10]. Developed in 1995, the ICube System was designed to give artists an easy means of digitizing environmental sensor readings into their DSP environments. The devices transferred data through MIDI, but were primarily designed as input devices, providing minimal output capabilities for controlling hardware. In 2004, Eric Singer developed the MidiTron [12]. The MidiTron allowed for bidirectional MIDI communication for the purpose of reading sensor data as well as driving voltage-controlled devices such as actuators and motors. Singer's device allowed artists to remotely perform a large array of devices from a single interface and was an early example of a scalable, low-cost performance framework for hardware.

Some other platforms are related to the BendIt.io system due to their mixing of hardware performance and IoMusT paradigms. The AppiOSC (developed by Lawson, Smith, and Appio [7]) and the Patchwerk system (developed by Mayton, Dublon, et al. [8]) both allow for web-based control over modular synthesizers in slightly different manners. The AppiOSC generates control voltage signals based on live-coded data from a browser-based graphics generator while the Patchwerk system doing allows for multi-user networked control of a massive modular synthesizer through manipulation of graphical dials and switches. Furthermore, the Orchestra of Things (devised by Stephen Beck and Chris Branton) explores the use of a central network for connecting multiple SMI's, mobile devices, and speakers to create distributed data streams for ensemble performances [2]. While no circuit-building or hardware hacking occurs in any of these systems, they do present novel means of mediating the control of physical hardware parameters through localized and networked means.

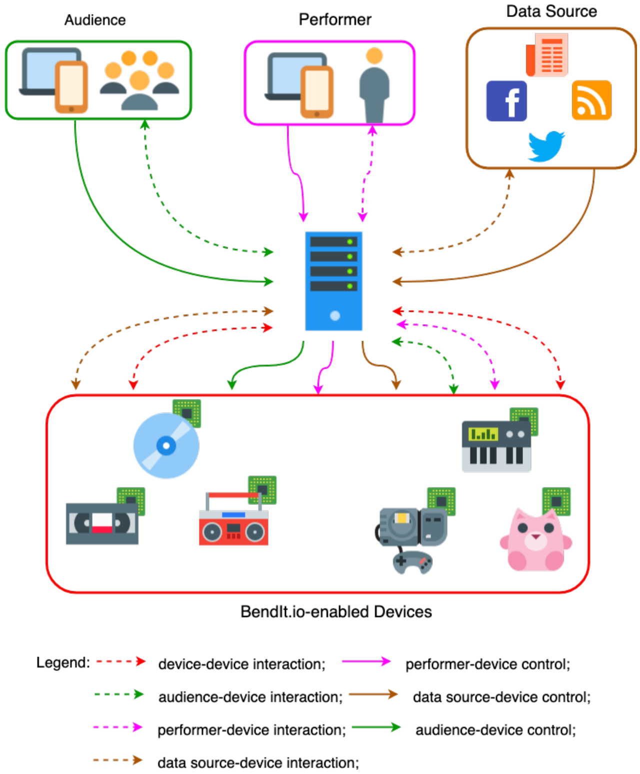

Figure 2: Interaction/control possibilities between connected users and hacked devices using the BendIt.io system.

3. SYSTEM DESIGN

The BendIt.io system consists of three discrete parts: a Wi-Fi-enabled I/O board (see Figure 1) wired to a hacked device (see Figure 3), a Node-based web server, and an web-enabled digital performance interface of the user's choosing. Both the hardware and software components of BendIt.io

[1]More information on the system, including video examples a user guide, can be found online at
`https://bit.ly/2I9sohF`

are designed to be modular and configurable during performance: each board automatically searches for a preset Wi-Fi network and available server upon powering up, meaning that new devices can be added and dropped from the network seamlessly. This flexible client/server communication model for distributed music performance draws inspiration from frameworks such as NexusHUB [1] and Rhizome [11].

3.1 The BendIt.io Board

The BendIt.io board contains an ESP32 microcontroller which acts as a client-side device and communicates with the server through socket.io. Additional peripherals include a six-channel digital potentiometer IC (capable of outputting 0kΩ to 100kΩ acting as either resistance dividers or a voltage dividers based on the number of ribbon-cable pins they decide to connect to their circuit-bent device), six reed relay switches, and a dual-H bridge motor driver. A breakout area include buses for DAC, ADC, and 3.3 volt lines provides users with additional I/O options when designing their hacked device modifications. A dedicated 12 to 24 volt power line is supplied for powering DC motors or solenoids, which can be used to control the speed and direction of turntable motors or the press and release of tape machine read/record heads. All output signals from the board terminate in multi-pin headers or screw terminal connectors. Using ribbon cables, users can disconnect one hacked device from their board and swap it out for a different one without needing to restart or reprogram the hardware. An example of connecting the board to a hacked device can be seen in Figure 3.

The ESP32 was chosen for its on-board ADC/DACs, low power consumption, multiple wirelss radios, complex GPIO multiplexing abilities, and its low price point (circa $6 to $10 for generic development boards, which v2.0 BendIt.io boards will utilize over the Adafruit model used in v1.0). It was a better option than other embedded boards (such as the Raspberry Pi Zero W or the Pocket Bela/Pocket Beaglebone combination) due to its low cost and built-in peripherals.

3.2 The Server

The BendIt.io Server works as an intermediary connection between web-enabled performance interfaces and any BendIt.io-enabled hacked devices. Performers can choose to host the server locally on their machine or remotely for globally-connected performances. When a new BendIt.io board connects, the server collects the unique socket ID and MAC address and stores it into an array. It then replaces this ID with a nickname (which can be displayed in the user's performance interface) and device color (displayed with the on-board LED), allowing for visual identification during performance. If the user is playing with a large number of circuit-bent devices, the server code provides the ability to group board ID's into smaller subgroups, giving them the ability to enact a particular bend only on the devices in a subgroup, while executing different control commands over the rest of the ensemble. This is helpful when performing with a mixed collection of hacked devices, providing the performer the ability to communicate with their devices as if they were separate sections of an orchestra.

3.3 Interfacing with BendIt.io

The server supports bidirectional messaging through socket.io, any internet-enabled software or custom-made web app that can format messages accordingly can serve as a means of communicating with devices on the network. Examples include web-based interfaces (e.g. web pages, social network/data API's, Node-Red), MaxMSP, PureData, SuperCollider, and other microcontrollers or microprocessors (e.g. BeagleBone Black, Raspberry Pi, Particle Argon, etc.). The server responds to OSC-style messaging schemes with a focus on keeping the syntax streamlined and easy to format. Each message must begin by addressing the assigned device number or subgroup name of the desired BendIt board (ex: `device/1/`). The rest of the message addresses the specific switch, pot channel, or motor channel to be activated, followed by a specific action and value. Table 1 shows a sample of common messages and resulting actions for performing devices attached to BendIt.io boards. A complete list can be found in the user guide.

Table 1: Programming interface examples for BendIt.io boards

Message	Action
switch/0/toggle 1	close switch 1
switch/0/metro 500	toggle switch every 500ms
pots/3/value 0.75	set pot 3 resistance value
pots/5/value 0.25 5000	ramp pot value over time
pots/0/metro 0.25 0.75 500	change between two values every 500ms
motor/0/pulse 1	throw solenoid

4. PERFORMANCE APPLICATIONS

There are a wide variety of performance applications for the BendIt.io system. A connected web-client could monitor social media feeds or changing weather data, route that data to the server, and trigger bend events on an ensemble of devices. Live performers experimenting with the board's DAC outputs could create on-board LFO generators and control voltage outputs useful for interacting with toy keyboards or tabletop synthesizers. These additional outputs could also be useful for hacking the optical pickup unit on portable CD players or the processing chips on video equipment, allowing for more complex datamoshing in live performance. Signals from points on a hacked device can also be "sniffed" and sampled into the microcontroller's memory through the ADC inputs. In performance, users can store them in a buffer and replicate them as method of modifying a digital potentiometer channel, creating performative feedback loops between the circuit bent device and the i/o board.

5. LATENCY TESTING

We ran a test with a single BendIt.io board and a single performer-device sending commands to the board via a webpage interface. Both devices were connected to the system server through a closed Wi-Fi network, broadcasted on a Linksys E2500's 2.4 GHz band. On a separate computer, an audio file was recorded into a DAW, and a switch output from the BendIt.io board was used to connect and break the audio signal remotely. For reference, a second audio track was used to record the sound of a mouse click on each mouse-up event. This allowed logging of the amount of time between triggering the toggle state from a mouse-up event and the mechanical switching of the board's relays.

The results of the test found that out of 21 toggle events, the average latency time between message and switch event was 87 milliseconds, with the fastest time logged at 33 milliseconds and the slowest time at 121 milliseconds. This means there was a significant amount of jitter – the standard deviation of the samples was 25 milliseconds when estimated using the `std` function in Octave. This latency could be due to a variety of sources. Notably, the Wi-Fi

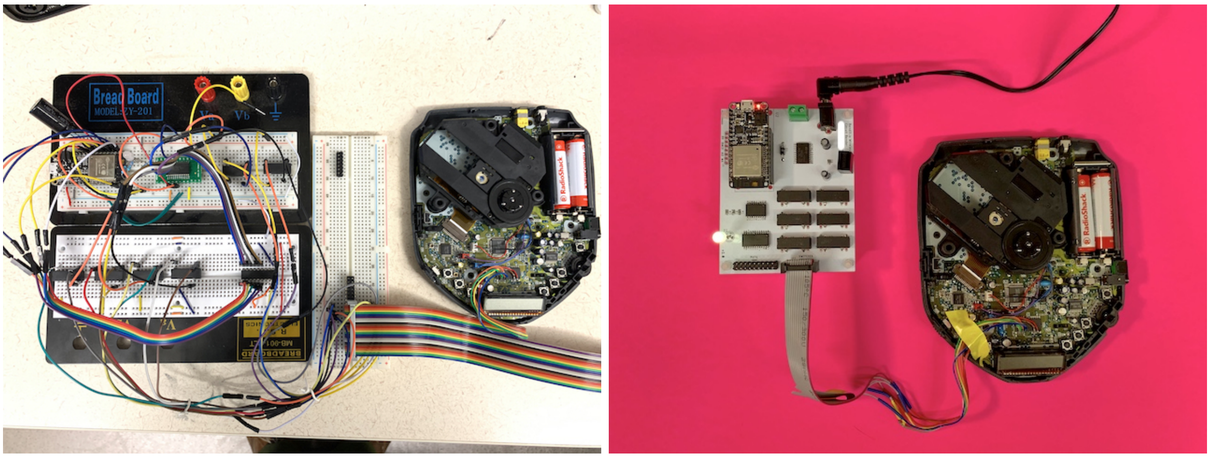

Figure 3: On the left, a prototype version of circuit-bending a CD player with a free wire and breadboard set-up. On the right, the same process is simplified and the player is able to be controlled through the network with the use of the BendIt.io board.

network is likely to be contributing significantly to the latency. To match or exceed these results in performance, we recommend connecting all devices to a single, closed Wi-Fi network and to take necessary steps to set-up the most robust wireless environment possible based on recent research for optimal music performance Wi-Fi practices [9].

6. FUTURE WORK & CONCLUSIONS

In future performances, the first author is planning to use the BentIt.io system to bend a wide variety of systems simultaneously in his upcoming performances and installations. The emphasis will be on connecting digital and Internet-based systems with classical circuit-bent hardware, to situate these kinds of works in a new light and adapt them for the 21st century. Future projects will also expand the number of simultaneously-connected BendIt.io boards through the same Wi-Fi network to realize ensembles of circuit bent devices and evaluate the network behavior. Latency will need to be reduced for precise, percussive performances; however, the system should operate fast enough for a very wide variety of kinds of experimental music practice, including interfacing traditional circuit bending approaches with modern networked systems.

BendIt.io brings together the DIY, lo-fi aesthetics of circuit bending and the networked complexities of distributed music paradigms. The system provides a low cost, user-friendly means of interacting with hacked devices for artistic means, and it allows hardware hackers and digital artists to work together in collaborative performances. It is our hope that this new hybrid system provides artists with a new outlet for experimenting with hacked devices in new and large-scale artistic ventures now and into the future.

7. ACKNOWLEDGMENTS

The authors would like to thank the Louisiana State University Center for Computation & Technology and School of Music for their support of this research.

8. REFERENCES

[1] J. T. Allison, Y. Oh, and B. Taylor. Nexus: Collaborative performance for the masses, handling instrument interface distribution through the web. In *NIME*, pages 1–6, 2013.

[2] S. D. Beck and C. Branton. The orchestra of things: a new approach to managing, controlling, and composing for laptop and mobile device ensembles. Presented at the 30th Annual SEAMUS conference, Virginia Tech, Blacksburg, VA, 2015.

[3] N. Collins. *Handmade electronic music: the art of hardware hacking*. Routledge, 2014.

[4] L. Gabrielli and S. Squartini. *Wireless networked music performance*. SpringerBriefs. Springer, 2016.

[5] R. Ghazala. *Circuit-Bending: Build your own alien instruments*, volume 15. John Wiley and Sons, 2005.

[6] E. Lakiotakis, C. Liaskos, and X. Dimitropoulos. Improving Networked Music Performance Systems Using Application-Network Collaboration. *arXiv:1808.09405 [cs, eess]*, Aug. 2018. arXiv: 1808.09405.

[7] S. Lawson, R. R. Smith, and F. Appio. Closing the circuit: Live coding the modular synthesizer. In *Proceedings of the International Conference on Live Coding*, 2016.

[8] B. D. Mayton, G. Dublon, N. Joliat, and J. A. Paradiso. Patchwork: Multi-user network control of a massive modular synthesizer. In *NIME*, 2012.

[9] T. Mitchell, S. Madgwick, S. Rankine, G. S. Hilton, A. Freed, and A. R. Nix. Making the most of wi-fi: Optimisations for robust wireless live music performance. In *NIME*, pages 251–256, 2014.

[10] A. Mulder. The i-cube system: moving towards sensor technology for artists. In *Proc. of the Sixth Symposium on Electronic Arts (ISEA 95)*, 1995.

[11] T. Shaw, S. Piquemal, and J. Bowers. Fields: an exploration into the use of mobile devices as a medium for sound diffusion. In *NIME*, pages 281–284, 2015.

[12] E. Singer. Miditron midi to real-world interface. In *ICMC*, 2004.

[13] J. Skjulstad. Circuit bending as an aesthetic phenomenon. Master's thesis, University of Oslo, 2016.

[14] L. Turchet. Smart Musical Instruments: Vision, Design Principles, and Future Directions. *IEEE Access*, 7:8944–8963, 2019.

[15] L. Turchet, C. Fischione, G. Essl, D. Keller, and M. Barthet. Internet of Musical Things: Vision and Challenges. *IEEE Access*, 6:61994–62017, 2018.