Seraph: An Educational Framework for Building Sensor-Driven Interactive Art and Music Projects

Solomon Rosenthal[1]
solrosenthal@alum.calarts.edu
California Institute of the Arts[1]
Valencia, California, United States

Andrew Piepenbrink[1]
apiepenbrink@calarts.edu
California Institute of the Arts[1]
Valencia, California, United States

Ajay Kapur[1,2]
akapur@calarts.edu
NYU Shanghai[2]
Shanghai, China

Abstract

In recent years, the DIY controller-building and sensor-based interactive art communities have grown significantly, driven by more accessible hardware and software infrastructures, audio and MIDI-oriented components, and new accessible creative coding and software frameworks. However, designing a custom MIDI controller or sensor-based interactive project remains non-trivial, requiring substantial engineering expertise. For musicians, artists, and makers whose primary skillsets lie outside of that domain, the friction between idea and finished product can fully halt development. The Seraph project addresses this problem by making it faster, more reliable and more accessible for educators, hobbyists and students to prototype MIDI controllers and interactive installation works without sacrificing flexibility or open source access. This paper presents the design and development of Seraph, a hardware and software platform intended to facilitate the rapid creation of USB-MIDI NIME controllers and sensor-based interactive art using the Teensy 4.1 microcontroller.

Keywords

Open Source Hardware/Software Framework, Educational Tool, PCB Breakout Board, Teensy USB-MIDI Accessory

1 Introduction

This paper presents the design and development of Seraph, an open-source hardware and software platform for rapid prototyping of USB-MIDI controllers and interactive projects using the Teensy 4.1 microcontroller. Seraph comprises a purpose-built breakout board and complementary codebase built on the USB-MIDI library. The breakout board simplifies I/O routing to peripherals and exposes the Teensy's full pin-set for expansion. The codebase provides a modular framework for mapping physical controls to MIDI messages, supporting a wide range of controller configurations. Informed by open-source culture, Seraph is designed around accessibility and modularity, and lowers the barrier to entry for experimental MIDI hardware projects without sacrificing flexibility. This paper describes the hardware and software architecture behind the project, analyzes workflow improvements for developers, musicians and artists, and presents platform case studies through Seraph driven student prototypes. The paper concludes with future work and implications for the broader DIY music technology and creative computing communities. Seraph was developed at CalArts over several years and informed by two decades of student collaboration.

2 Background and Related Work

The USB-MIDI specification has become a standard interface for computer-based musical instruments, enabling class-compliant operation and high channel and endpoint flexibility for musical controllers. The microcontroller board of choice for many DIY MIDI enthusiasts has increasingly become the Teensy 4.1, owing to its high clock speed (600 MHz Cortex-M7), large pin count, multiple serial ports, USB full-speed and host capability, strong support through the Teensyduino/Arduino ecosystem, and built in MIDI drivers. Historically, projects such as MIOS and the MIDIbox platform established foundational frameworks for DIY MIDI controllers through modular design, open schematics, and shared firmware. More recently, breakout boards tailored to the Teensy line (for example, multi-port MIDI breakout systems) have appeared, offering turnkey solutions for controller and interface development [1]. Parallel firmware efforts such as HIDUINO demonstrated early approaches to building driverless USB-MIDI devices using microcontroller platforms, further lowering barriers for artists and instrument designers [4]. Alongside these hardware and firmware

* Both authors contributed equally to this research

NIME '26, June 23–26, 2026, London, UK

efforts, a number of influential frameworks and pedagogical models have emerged from the NIME and academic music-technology communities, including CUI-style controller abstraction systems [2], early modular controller frameworks developed at CCRMA by Gurevich and Verplank [3], and educational approaches to physical computing such as those articulated by Tom Igoe at NYU's ITP [6]. Broader curricular and pedagogical reflections within the NIME community have also shaped the discourse around instrument design and experimental interface building [5][7][8]. Work such as the Bela platform and related research by Andrew McPherson has further advanced the field by emphasizing low-latency, high-resolution interaction for musical interfaces. Together these developments indicate a growing ecosystem of accessible music-oriented control infrastructures. However, there remains significant value in further streamlining workflows and increasing modularity for experimental controllers that require more open and adaptable architectural starting points.

3 Hardware/Firmware Design

Designing a MIDI controller presents challenges across both hardware and software domains. On the hardware side, builders must either design a PCB with correct signal routing, power regulation, and input conditioning, or construct an equivalent perfboard circuit by hand, both of which present a steep learning curve for newcomers. On the software side, firmware must correctly handle USB-MIDI enumeration, translate physical control data into standardized MIDI messages, and maintain stable system states for functions like LED feedback and preset management. Seraph addresses both sides of this problem. Its PCB removes the need to understand complex circuitry upfront, its modular codebase allows users to build sophisticated controller systems without specialized programming knowledge. The result is a platform that supports both rapid prototyping and longer-term production workflows, remaining flexible and customizable without demanding an advanced engineering background.

3.1 Design Goals

The design for Seraph was driven by a variety of goals. First, we considered accessibility and ease of use, which we have accomplished through our clearly labeled PCB and our accessible and highly documented codebase. Seraph allows essentially any peripheral (buttons, encoders, LEDs, displays, analog sensors) to be easily added to projects with minimal soldering and little to no circuitry or programming experience. We also designed the project with open manufacturing in mind by providing PCB design files, BOM, and documentation lists so that makers can build, modify, or extend the board, all of which can be found on our project website[1]

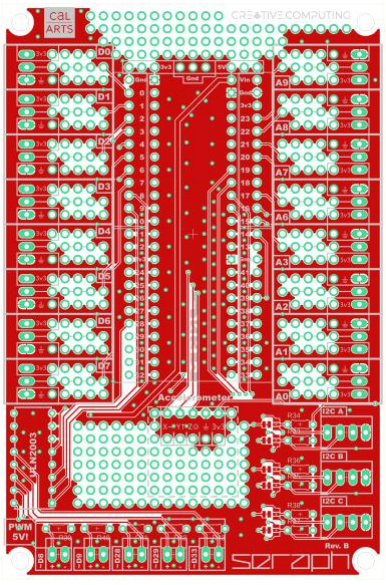
3.2 Board Architecture

The main section of the board comprises two 8-channel I/O banks, one for digital/PWM pins and the other dedicated to analog capable pins. Each pin is exposed in a modular 'channel strip' block with a 3-pin header for power, ground, and the pin itself. The block provides a small prototyping area for adding conditioning circuitry to the channel; voltage dividers, logic level shifters, pullup/pulldown resistors, and directly mounted sensors can all be accommodated. For applications beyond the 16 I/O blocks, the full Teensy pinset is also accessible from screw terminals adjacent to the MCU socket

Many interactive circuit applications require more current and/or voltage than a microcontroller I/O pin can tolerate/deliver. Seraph provides five channels of high-current open-drain outputs for solenoids, motors, and LEDs through a ULN2003A Darlington transistor. This IC was selected for its ruggedness and simplicity, reducing the likelihood of damage from beginner circuit mistakes and avoiding the design pitfalls and ESD vulnerability of MOSFETs. The ULN2003A is mounted in a DIP socket, making replacement easy in the event the chip becomes damaged.

Seraph exposes I2C connectivity through three blocks (A, B, C), each with a 4-pin header and optional pullup resistors. The T4.1 has three independent hardware I2C ports (I2C0, I2C1, I2C2), but many legacy Arduino libraries are hardcoded to I2C0, presenting an obstacle to beginners using the alternate ports. For this reason, all three blocks are jumpered to I2C0 by default. For advanced users, the alternate ports can be patched to a block using a combination of solder jumpers and cuttable traces.

The lower part of the board contains a large protoboard area, combined with the footprint of an ADXL335 analog accelerometer. A smaller protoboard strip near the Teensy's USB jack is ideal for adding external power, charging, and switches.



[1] https://creativecomputing.calarts.edu/seraph/

Figure 1: Front face of Seraph PCB

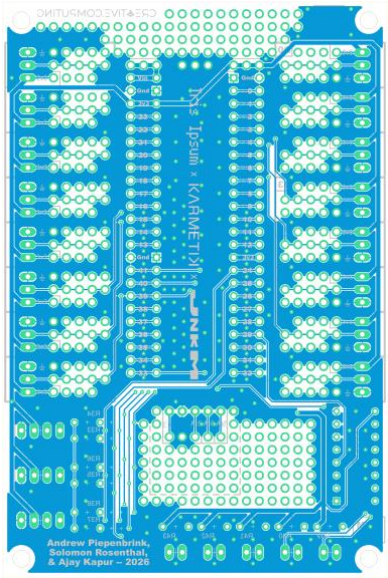


Figure 2: Back face of Seraph PCB

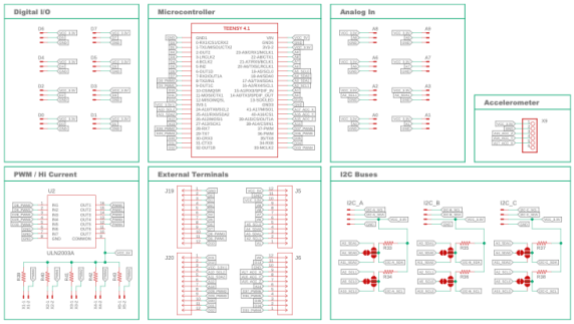


Figure 3: Seraph schematic

3.3 Fabrication and Assembly

The Seraph breakout board is fabricated using standard 2-layer PCB manufacturing processes to keep cost low for hobbyists and small batches. The BOM lists compatible through-hole components for easy hand-soldering (headers and accelerometer), and assembly instructions and revision history are provided via the Seraph GitHub.[2]

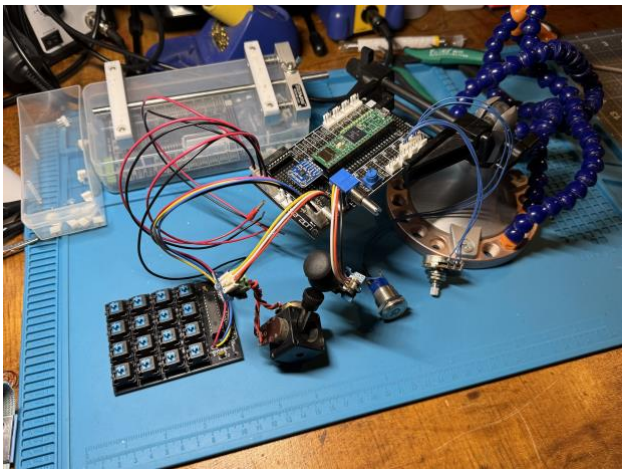


Figure 4: Seraph assembled

3.4 Firmware / Software Architecture

Seraph's demo firmware is organized as a set of small, single-file example sketches that follow a consistent sensor-to-MIDI pipeline. Each sketch is intended to be readable for beginners while still reflecting good controller-building practices including basic signal conditioning and event based MIDI transmission. The demos serve as working templates that can be copied and extended directly to custom controllers, and can be be easily modified to suit the needs of individual projects. The codebase is built on the Teensyduino USB-MIDI library, which allows the Teensy 4.1 to act as a USB MIDI class-compliant device. Users can load any demo file, connect their Teensy board mounted to Seraph, edit the file to suit their specific circuit, and compile/upload via the Arduino IDE to create their custom interface. The codebase abstracts much of the USB-MIDI enumeration and state-handling logic, letting makers focus on musical mapping and UI behavior instead of low-level USB complexities. Demo sketches and circuit diagrams such as the one shown in figure 5 can be found on the Seraph Github repository.[3]

[2] https://github.com/Calarts-Creative-Computing/SERAPH
[3] https://github.com/Calarts-Creative-Computing/SERAPH

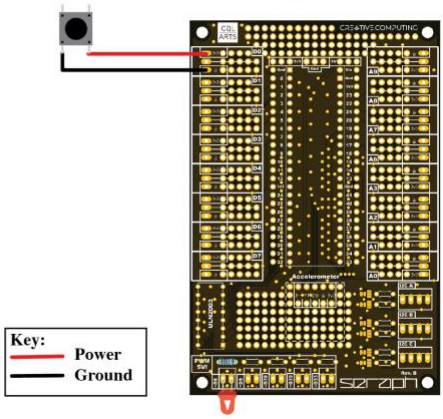


Figure 5: Instructional Seraph wiring diagram for a simple button and led circuit

4 Case Studies

In educational contexts, Seraph offers students an intuitive way to easily explore the development of their very own complex and novel control interfaces with minimal experience. The ability to jump right into a project without the need for an overly complicated perf or breadboard circuit supports our goals of allowing users with minimal experience to explore complex projects.

Through its use in the Interface Design Class at Calarts, our community has conducted a variety of prototype builds using the board, detailed below. The following project descriptions were written by each builder and are presented here with their permission.

4.1 H.A.I (Head as Interface) by Zelia ZZ Tan, Paolo Sandejas and Fuyu Wang

Figure 6: Head As Interface

H.A.I. (Head as an Interface), shown in figure 6, is a wearable sonic instrument that converts bodily motion into sound, positioning the performer as an active component of the signal chain. Distance sensors mounted on the head capture spatial gestures and translate them into real-time sonic modulation, fostering an improvisational and physically engaged mode of interaction. Emphasizing embodiment, play, and performative presence, the work interrogates conventional distinctions between human and technological systems, instead presenting an adaptive interface in which movement, perception, and sound operate as a unified expressive continuum.[4]

4.2 Lumophone by Benjamin Norbrook

Figure 7: Lumophone

The Lumaphone, shown in figure 7, is a light-responsive delay processor whose behavior evolves dynamically in response to environmental illumination. At its core is a feedback system linking an LED and a light sensor, in which the LED's brightness is inversely coupled to the sensor's input: increased ambient light dims the LED, while darkness intensifies it. When this interaction is accelerated, it produces a nonlinear feedback network whose sonic character shifts continuously according to both user input and surrounding light conditions. The system is driven by a Teensy 4.1 microcontroller communicating with Max/MSP via serial connection, with additional performance control provided by four potentiometers and two force-sensitive resistors, all of which are connected via the Seraph platform. Two potentiometers calibrate the sensor to ambient lighting, while the remaining controls adjust feedback rate and maximum delay time. The force-sensitive resistors introduce expressive modulation, with one influencing refresh rate and the other functioning as a wet/dry mix control. Directly illuminating or occluding the light sensor further alters the system's response, allowing performers to mute, intensify, or destabilize the delay. Through these interacting parameters, the Lumaphone produces a highly responsive sonic output ranging from stable rhythmic oscillations to dense, chaotic textures.[5]

[4] youtube.com/watch?v=rxaJTSi7KDY&feature=youtu.be

4.3 VineiTone by Jason Zeng, Eric Wang, and Soohyun Suzy Park

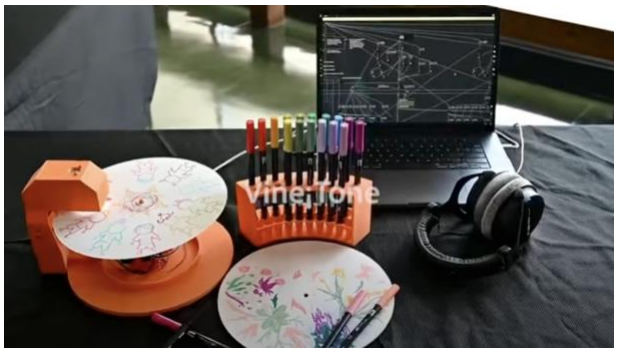


Figure 8: Vine¡Tone

Vine¡Tone, shown in figure 8, is a color-sensing turntable instrument that translates chromatic markings on a vinyl disc into generative sound. Users inscribe the surface of the record with colors representing emotions, memories, or temporal states, and as the disc rotates, an embedded color sensor reads these markings and converts them into musical output, effectively reanimating recorded visual traces as audio. The installation integrates a motorized platter, distance sensing for presence detection, and automated playback logic: the record spins upon activation and halts when the disc is removed, at which point audio output is muted, mirroring the operational behavior of a conventional turntable. Through this system, Vine¡Tone frames the vinyl medium not as a storage device for fixed recordings but as a dynamic interface in which color, motion, and sound form a continuous expressive loop.

4.4 Results

As demonstrated in the examples shown above, the Seraph platform has reliably facilitated the creation of novel interactive projects in our community. We believe the platform is a valuable tool for any educator, student or enthusiast looking to simplify the process of interactive electronics development.

7 Conclusion and Future Work

Seraph is an open-source hardware and firmware platform that simplifies the process of building an interactive sensor based project. With its breakout board and modular set of demo programs, the project reduces development time, increases flexibility, and lowers the technical barrier for makers, musicians, and students. Evaluation via prototype builds demonstrates the effectiveness of this approach. We invite the community to adopt, remix, and extend the platform, and envision a future where custom musical controllers can be designed and built with ease. As DIY and computer-music practices expand, tools like Seraph help bridge the gap between concept and instrument, even for individuals who have less of a technical background.

Several future improvements are planned for the project. This includes adding support for audio I/O, designing enclosure modules and front-panel templates to streamline mechanical integration, more extensive user documentation, tutorial videos, and community sharing of controller templates, and more extensive evaluation in educational settings to assess learning outcomes and community adoption.

Ethical Standards

This work was conducted as part of an academic research and creative practice project within the Creative Computing Program at the California Institute of the Arts. No external commercial funding, corporate sponsorship, or industry partnership supported the development of this project. All design, fabrication, documentation, and evaluation activities were carried out independently for educational and research purposes. The author declares no financial or non-financial conflicts of interest related to this work. No proprietary restrictions, licensing agreements, or commercial obligations influenced its design, implementation, or presentation.
This research did not involve human subjects in a manner requiring institutional review, nor were any formal user studies, behavioral experiments, or identifiable participant data collected. Informal testing and demonstration of prototypes were conducted solely for technical validation and did not involve personal data collection, evaluation of individuals, or experimental protocols. No animals were involved in this research. The project is intended as an open, educational tool for artists, designers, and engineers.

Acknowledgments

This work was completed within the Creative Computing Program at the California Institute of the Arts, whose facilities, resources, and interdisciplinary environment supported the development of this project. The authors gratefully acknowledge the faculty, staff, and student community whose feedback, discussions, technical guidance and prototype builds contributed to the refinement of both the hardware and software components. The authors also thank members of the broader physical computing and electronic instrument design communities whose open-source tools, published research, and shared knowledge informed this work. Additional appreciation is extended to the developers and maintainers of the Teensy platform and its USB-MIDI libraries for providing the technical foundation that made this project possible.

[5] https://creativecomputing.calarts.edu/lumaphone/

References

[1] [author not available]. [date not available]. Teensy 41 Midi Breakout Board 8in 8out Usb Host. www.tindie.com. Retrieved February 12, 2026 from https://www.tindie.com/products/deftaudio/teensy-41-midi-breakout-board-8in-8out-usb-host/

[2] Dan Overholt. Musical Interaction Design with the CREATE USB Interface: Teaching HCI with CUIs instead of GUIs. In Proceedings of the ACM Conference, 2005.

[3] Michael Gurevich, Bill Verplank, and Scott Wilson. Physical Interaction Design for Music. CCRMA, Department of Music, Stanford University, Stanford, California, USA.

[4] Dimitri Diakopoulos and Ajay Kapur. HIDUINO: A firmware for building driverless USB-MIDI devices using the Arduino microcontroller. California Institute of the Arts and New Zealand School of Music.

[5] E. Tomás. 2020. A Playful Approach to Teaching NIME: Pedagogical Methods from a Practice-Based Perspective. In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME '20), 143–148.

[6] Dan O'Sullivan and Tom Igoe. 2004. *Physical Computing: Sensing and Controlling the Physical World with Computers*. Course Technology, Boston, MA. ISBN: 159200346X.

[7] G. D'Arcangelo. 2002. Creating a Context for Musical Innovation: A NIME Curriculum. In Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME '02), 1–4.