Connecting MIDI Interfaces to the Musical Metaverse

Gregorio Andrea Giudici, Domenico Stefani, Alberto Boem, and Luca Turchet
Department of Information Engineering and Computer Science
University of Trento
gregorio.giudici@unitn.it, domenico.stefani@unitn.it, alberto.boem@unitn.it, luca.turchet@unitn.it

***Abstract*—The integration of physical control surfaces with virtual reality environments presents significant challenges, particularly in the Musical Metaverse where users are musicians who are often encumbered by their instruments while engaging in musical practices. This paper presents MUSMET-Bridge, a standalone embedded system designed to connect USB MIDI control surfaces directly to standalone VR headsets through wireless networks, reducing the interaction overhead introduced by the use of a laptop. The system offers superior portability through its pocket-sized design powered by standard power banks, eliminates cable clutter that complicates performance setups, and provides plug-and-play operation that abstracts technical complexity from end users. The system was designed for MIDI-to-OSC message conversion for controlling Musical Metaverse applications, and was tested with the PatchWorld XR platform. We conducted a comparative evaluation study that examined the setup procedures and user experience between the proposed system and conventional laptop-based workflows using TouchOSC. Our results indicate a reduced setup complexity, with MUSMET-Bridge reducing average setup time while achieving comparable usability scores and maintaining identical responsiveness and precision for musical expression. The system demonstrates potential for enhancing accessibility to the Musical Metaverse while reducing the interaction overhead associated with current computer-mediated connection setups.**

***Index Terms*—Musical Metaverse, Open Sound Control, Embedded Computers.**

I. Introduction

Immersive social environments have rapidly become promising platforms for musical creation and performance, offering musicians access to a wide range of virtual instruments and collaborative spaces (e.g., [1]–[4]). Although the visual aspect of Virtual Reality (VR) provides new creative possibilities, physical interaction remains vital for effective music practice and expressive performance, particularly tactile feedback and the nuanced control offered by physical input interfaces [5]–[7].

Nowadays, the majority of Extended Reality (XR) headsets used to experience immersive contents and interact with virtual worlds are standalone, commonly equipped with WiFi and Bluetooth, promoting portability and connectivity with external devices [8]. However, integrating standard physical MIDI control surfaces with standalone VR technologies remains a significant challenge, and current workflows typically require the use of intermediary laptops to bridge MIDI hardware with VR applications [9]–[12]. This computer-mediated setup introduces additional complexity, increases setup time, and can encumber musicians, especially those who must physically handle their instruments with both hands during performance or practice, creating unnecessary barriers for musicians seeking seamless integration of their physical instruments with virtual environments. These technical barriers are particularly acute in the context of the Musical Metaverse [13], where seamless interaction between physical and virtual instruments is crucial for both creative exploration and live performance.

Motivated by the need to streamline the interaction and enhance the accessibility of Musical Metaverse platforms, we propose *MUSMET-Bridge*, an embedded system that enables direct wireless connectivity between USB MIDI control surfaces and standalone VR headsets, eliminating the need for a laptop intermediary (see Fig. 1).

Therefore, the primary objectives of this work are to:

- Lower the technical and practical barriers for musicians entering a Musical Metaverse by simplifying the setup process;
- Preserve the tactile expressivity of physical MIDI devices within immersive VR environments;
- Evaluate the usability and performance of the proposed system in comparison to conventional laptop-based workflows.

Fig. 1. User interacting with the MUSMET-Bridge (visible on the right).

The main contributions of this paper are as follows:

1) The design and implementation of MUSMET-Bridge, with all software and hardware designs available as open source and freely accessible online[1];
2) A comparative evaluation of setup procedures and user experience between MUSMET-Bridge and traditional laptop-based solutions;
3) Empirical insights into the potential of standalone embedded systems to enhance accessibility and reduce interaction overhead in Musical Metaverse environments.

By addressing the technical and ergonomic limitations of current MIDI-to-VR workflows, this research aims to empower musicians with more intuitive and efficient tools for creative expression in the evolving landscape of the Musical Metaverse.

The remainder of this paper is organized as follows. Section II presents background on interfaces for Musical XR, network protocols for music, encumbered musical interactions, and the PatchWorld Musical Metaverse platform. Section III describes the system architecture of MUSMET-Bridge. Section IV details our comparative evaluation methodology, while Section V presents and discusses the results of our evaluation comparing the proposed system with conventional laptop-based workflows. Finally, we draw our conclusions in Section VI.

II. Background

A. Interfaces for Musical XR

XR has emerged as a significant medium for experiencing new forms of music-making [5], [6]. In this space, researchers have explored various approaches to translate physical musical control into virtual environments.

The foundational approach for interacting with XR musical instruments is gestural input, typically using performers' hands. Examples range from immersive environments where users can literally sculpt sounds with their hands [14], to purely virtual drums and guitar driven by arms and hand gestures [15], and shared virtual worlds where hand gestures can be used to control sound generation in real-time [16].

However, the lack of direct control and tactile feedback in gestural interaction was perceived as a limitation of such interaction techniques. To address these shortcomings, researchers have developed custom physical interfaces designed specifically for XR music-making. Notable examples include sensorized gloves equipped with vibro-tactile feedback [17], a pressure-sensitive handled device combining pointing and expressive control [18], and custom-made tangible surfaces [19]. Encountered-type haptic interfaces have also been proposed as a way to bridge the gap between the intangible nature of virtual instruments and the tactile demands of musical performance [20]. More recently, Dziwis and Hadjakos [21] proposed a modular approach with interface modules for designing XR music systems, allowing for flexible combinations of physical and virtual controls tailored to the needs of performers and composers.

[1]https://github.com/CIMIL/musmet-bridge

Moreover, beyond custom input devices and hardware solutions, researchers also explored the use of standard MIDI devices as controllers for XR musical instruments. Bravo and Fasciani [22] presented a human-agents performance system where conventional MIDI controllers are used alongside virtual agents in an XR setting. Desnoyers-Stewart et al. [11] explored augmenting a MIDI keyboard with virtual interfaces, enabling new forms of interaction and feedback by overlaying virtual controls and information onto a familiar physical device. Johnson et al. [12] evaluated the effectiveness of Mixed Reality (MR) instrument learning using the theremin, a classic electronic instrument, demonstrating how XR can enhance traditional instrument pedagogy by providing real-time visual and spatial cues.

Despite these achievements, most current studies that integrate MIDI devices into XR environments, such as those discussed above, rely on a laptop as an intermediary for physical MIDI integration. This approach does not fully leverage the portability and immediacy that modern standalone VR systems offer. As a result, these solutions fall short of realizing the seamless, self-contained experiences envisioned in the broader context of the Musical Metaverse, where direct and untethered interaction is a core principle. This gap highlights the motivation for the present work by removing the laptop dependency and embracing the native capabilities of standalone XR platforms.

B. Network Protocols for Sound and Music

The Musical Instrument Digital Interface (MIDI) protocol has been the standard for connecting physical controllers to sound synthesis and processing systems since the 1980s. Its most notable network extensions are the RTP-MIDI standard[2] for sending/receiving MIDI 1.0 through local networks, and the newer Network MIDI 2.0[3].

However, network MIDI protocols are not widely supported across musical control surfaces and VR metaverse software, which instead often support Open Sound Control (OSC) [23], due to its simplicity and flexibility. OSC's primary advantage lies in its generic and flexible structure that allows sending multiple values of different data types within a single message. Unlike MIDI's fixed message format, OSC provides a dynamic, open-ended, URL-style naming scheme that can accommodate arbitrary data structures. This flexibility enables OSC to handle complex parameter sets and multi-dimensional control data [24].

In response to the coexistence of MIDI and OSC protocols in creative workflows, several software solutions have emerged to bridge MIDI-to-OSC communication. These tools convert MIDI messages into OSC format or vice versa, allowing MIDI controllers and devices to interface with OSC-enabled applications and environments. A notable example is the TouchOSC software [25], which is a modular control surface toolkit for creating custom controllers.

[2]https://midi.org/rtp-midi
[3]https://midi.org/network-midi-2-0

C. Encumbered Musical Interactions

Musicians engaging in musical practices with two-handed musical instruments such as guitars have been shown to be typically encumbered by having their instrument in their hands or by playing [26]. In this context, interacting with physical or digital tools (e.g., DSP effects or a laptop) can be in the way of playing, resulting in secondary mental demands that can render the performance with the instrument less accurate [27]. In their work, Avila *et al.* [26] adopted an ethnographic approach to analyze several musicians engaging in individual and collaborative practices related to music making. In their analysis, the authors found multiple opportunities for augmenting guitars to improve the musician-instrument relationship, which converged on the need to support encumbered interactions. In particular, the authors proposed to reduce the overhead of transitioning between playing and interacting with other resources.

In this spirit, the Section III describes a device designed to connect physical control surfaces to a Musical Metaverse application and aimed at reducing the overhead added by the use of a laptop as a connecting device.

Fig. 2. MR environment in PatchWorld with a full see-through configuration (top) or a passthrough window on a VR configuration (bottom).

D. PatchWorld

A notable example of Musical Metaverse application is PatchWorld[4] by PatchXR [28]. PatchWorld is an XR platform for immersive music creation, experimentation, and collaboration within virtual environments. It enables multiple users to build virtual musical instruments from scratch and play them in shared immersive environments. Patchworld features a library of devices and tools to be combined and interconnected in using a visual patching paradigm (see Fig. 2), similar to those used in music software such as Pure Data and Max/MSP without requiring coding skills.

Additionally, PatchWorld integrates MR capabilities, blending virtual and real-world elements to expand musical interaction possibilities. Features such as passthrough camera views allow users to combine real instruments with virtual enhancements. PatchWorld supports OSC messages, enabling bidirectional exchange of control data for sound synthesis and effects. This has been used for integration with professional music software such as Ableton Live, which gives users the possibility of controlling sound processing tools from a virtual environment.

Fig. 3. PatchWorld OSC-MIDI receiver objects.

However, more interesting for Musical Metaverse applications is the possibility of receiving MIDI-like OSC messages in a virtual environment. For this, the PatchWorld community has developed a control surface for the aforementioned TouchOSC software and a matching virtual receiver (see Fig. 3), integrating both screen controls and conversion of incoming MIDI messages by using a laptop as intermediary[5]. This enables the use of physical musical interfaces for controlling sound synthesis and manipulation processes that happen entirely in the virtual environment and can be heard by other users, supporting the metaverse aspect of the application.

III. System Architecture

The MUSMET-Bridge is designed to wirelessly connect physical MIDI control surfaces to standalone head-mounted displays (HMDs) within a standard WiFi network. This enables controlling virtual devices in Musical Metaverse environments with physical controllers without the intermediary of a personal computer (see Fig. 4).

[4]PatchWorld: https://patchxr.com/

[5]https://youtu.be/yNEAcQKFzz0

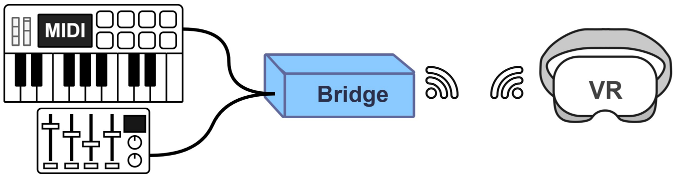


Fig. 4. Usage scenario for the Bridge.

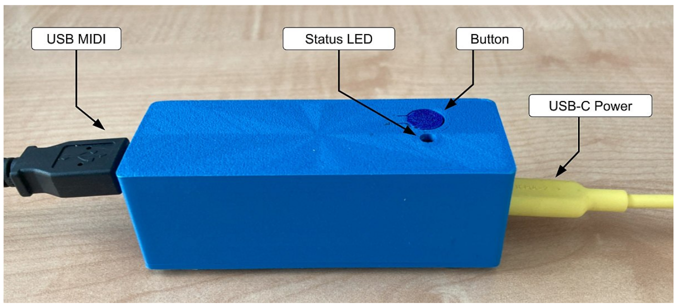


Fig. 5. Assembled MUSMET-Bridge. MIDI devices are connected to the USB-A port on the right, while the yellow cable on the right supplies power through a USB-C connector.

A. Bridge Operation

The Bridge presents itself as a small[6] plastic box with a USB-A port, a USB-C port, a push button, and a status LED (see Fig. 5). Powering on the device is done by connecting a power supply, such as an AC adapter or power bank, to the USB-C port. Power is confirmed by a series of fast blinks of the status LED, followed by two slow blinks.

By default, the Bridge powers on in "Play-mode", meaning that it will directly attempt to connect to a preconfigured WiFi network. Upon successful connection, the Bridge waits for one or more[7] MIDI devices to be connected to its USB-A port. Every MIDI message from connected control surfaces can then be forwarded to a pre-configured IP address and Port combination as OSC messages. The code provided in the online repository is set up to forward all note, control change, aftertouch, and pitch bend messages, and uses an OSC scheme that is compatible with PatchWorld (see Section III-C).

The Bridge is configured with the use of a smartphone, tablet, or computer, and require booting the device in "Configuration-mode". Configuration-mode is enabled by re-booting the Bridge using the push button, waiting for the fast blinks to be over, and then rebooting again after the first slow blink. In configuration mode, the Bridge will appear to any WiFi capable device as an access point and will accept connections. Once connected to the Bridge, users can access the configuration page with a browser, with the URL `http://bridge.conf/`. The configuration page (see Fig. 6) allows setting (i) the WiFi Network SSID, (ii) the WiFi Password, (iii) the IP address of the OSC target, and (iv) the relative port.

[6]Width: 33 mm, height: 90 mm, depth: 30 mm
[7]Up to four MIDI devices can be connected with a USB hub.

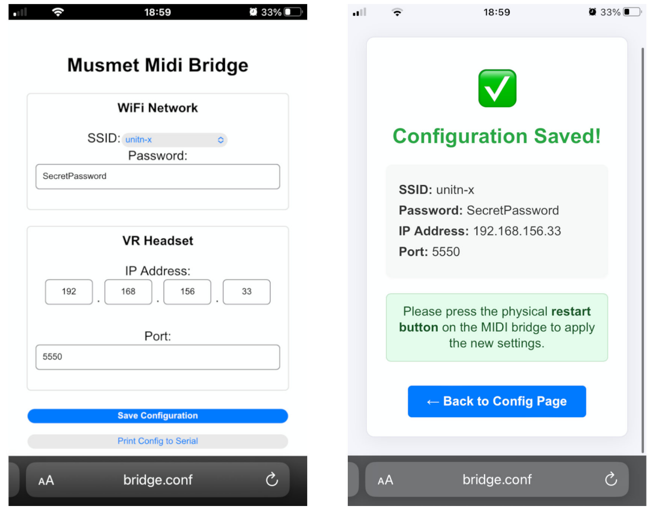


Fig. 6. MUSMET-Bridge configuration page.

Once the configuration is saved, the Bridge can be rebooted with the push-button. Configurations are stored in the flash memory of the microcontroller and persist after rebooting.

B. Hardware

The Bridge is based on the Raspberry Pi Pico W micro-controller. The Pico W is mounted on a circuit board (see Figure 7) that hosts the reboot pushbutton, the status LED, and routes power from a generic USB-C power breakout board. Power is supplied via a USB Type-C port to ensure compatibility with a wide range of modern power supplies and power banks. The Bridge is characterized by an exceptionally low power consumption[8], making it highly suitable for mobile and battery-powered applications.

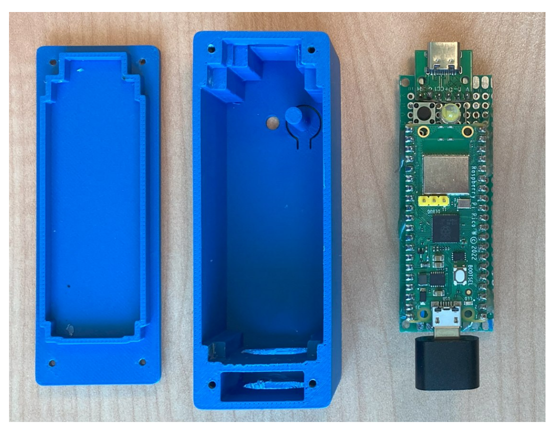


Fig. 7. Disassembled MUSMET-Bridge, with the 3D printed lid and enclosure on the left, and the circuitboard with Raspberry Pi Pico W and adapters on the right.

Currently, the Bridge is built on a perforated circuit board, but PCB design files are available in the online repository.

[8]Raspberry Pi Pico W's power consumption is roughly 50/60mA during active usage.

To allow connecting USB MIDI control surfaces, which often come with cables with a USB-A plug, the Bridge used an USB-A-Socket to USB-B-micro plug adapter. The adapter allows connecting to the USB-B micro socket on the Raspberry Pi Pico W. The entire hardware assembly is enclosed within a custom 3D printed PLA case (see Figure 7).

C. Software

The firmware of the Bridge was developed using the Arduino framework, leveraging the following open-source libraries:

1) **Arduino-Pico**[9]: a port of Arduino for Pico microcontrollers, providing support for the Raspberry Pi Pico W and enabling integration with existing Arduino libraries (e.g., WiFi and LittleFS libraries).
2) **EZ_USB_MIDI_HOST**[10]: Manages USB MIDI host functionality, allowing the Bridge to enumerate and communicate with class-compliant MIDI devices.
3) **OSC**[11]: Implements the OSC protocol for Arduino, facilitating the construction and transmission of OSC messages over WiFi.

A core part of the Bridge operation is the "Configuration-mode", which consists of a simple web-page hosted on the microcontroller itself. The microcontroller exposes a WiFi access point, and presents the HTML webpage on port 80 of the gateway. Form responses are fed to an Arduino callback, where a configuration file is saved in the flash memory with the LittleFS library.

In "Play-mode", the Raspberry Pico reverts back to being a WiFi device and attempts to connect to the network with the configured credentials, showing failure and success with different blink patterns.

***MIDI-OSC translation**:* OSC is a flexible network protocol with an open-ended message format. OSC messages consist of an address pattern string, a list of argument types, and the actual arguments. The code of the Bridge can therefore be adapted to translate incoming MIDI messages following the requirements of the OSC recipient. In this case, the Bridge was prepared to follow the OSC address scheme used in the available PatchWorld MIDI-OSC blocks (Fig. 3). MIDI messages with multiple arguments are split into multiple OSC messages, as default PatchWorld OSC receiver objects output a single value.

TABLE I
MIDI-OSC TRANSLATION FOR PATCHWORLD

MIDI		**OSC**	
Message	**Arguments**	**Address**	**Arguments**
NoteOn	pitch velocity	/ch{channel}note /ch{channel}nvalue	pitch velocity
NoteOff	pitch velocity	/ch{channel}noteoff /ch{channel}noteoffvalue	pitch velocity
ControlChange	controller value	/ch{channel}cc /ch{channel}ccvalue	controller value
PitchBend	value	/ch{channel}pitch	value
AfterTouch	value	/ch{channel}pressing	value

IV. Evaluation

We conducted an evaluation study aimed at comparing two setup conditions for interfacing with PatchWorld with physical MIDI devices:

1) *Laptop-based* setup with TouchOSC (default procedure, as described in PatchWorld tutorials);
2) The proposed *MUSMET-Bridge*.

For each participant, the study was composed of an acclimatization stage, a setup with Condition 1, a setup with Condition 2, and a repetition of each setup condition in a different room with a different WiFi network. The order of the setup stages with the two conditions was reversed for half of the participants. The following sections describe in detail the demographics of the participants, the evaluation procedure, and both the qualitative and quantitative measures conducted.

A. Participants

Ten musicians took part in the experiment (8 Italians, 1 Sri Lankan, 1 Vietnamese; 1 female, 7 males, 2 who prefer not to say), aged between 21 and 39 (mean 27.7, SD 6.1). They reported having an average of 13.6 years of musical experience (SD 7.2), and on average started learning to play music at the age of 14.1. Participants were selected for their past experience with MIDI controllers and computers for music performance or production. Four participants used a MIDI keyboard for the experiment, three used a 4×4 grid of velocity-sensitive pads, and the remaining three used an electric guitar with the Fishman TriplePlay Connect MIDI pickup[12]. All participants were provided with an additional slider-based MIDI interface to control sound parameters, sequencers, and a looper.

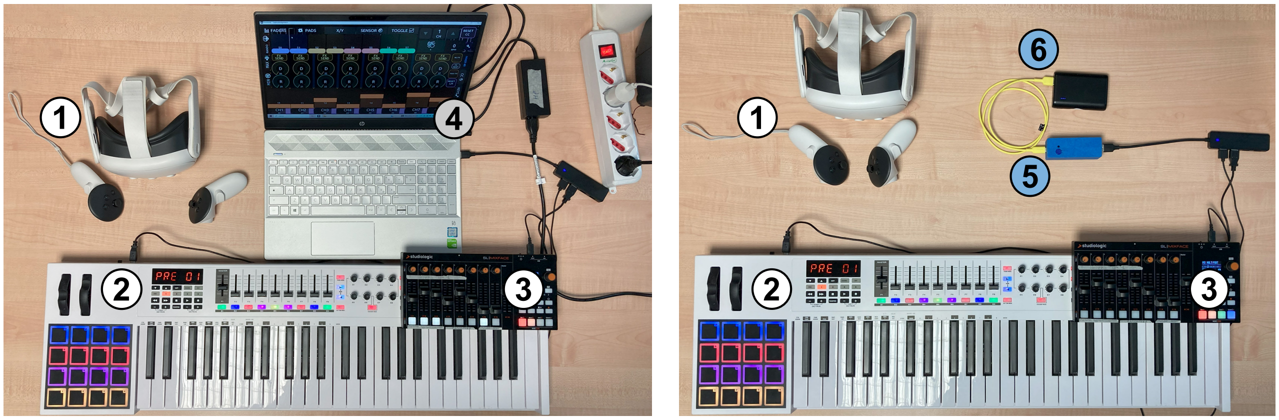
Participants took on average 80 minutes to complete the whole experiment. The procedure, approved by the local ethical committee, was in accordance with the ethical standards of the 1964 Declaration of Helsinki.

B. Procedure

The evaluation study consisted of single-participant sessions and was performed at the University of Trento.

Participants were lead to Room A and presented with three MIDI instruments (i.e., a semi-weighted keyboard, a 4×4 grid of velocity sensitive pads, and a guitar equipped with a MIDI pickup) and asked to choose one that they felt more comfortable with. Then, participants underwent an acclimatization stage, where they were asked to wear an HMD with PatchWorld, while the experimenter configured the

[9]https://github.com/earlephilhower/arduino-pico
[10]https://github.com/rppicomidi/EZ_USB_MIDI_HOST
[11]https://github.com/CNMAT/OSC
[12]https://www.fishman.com/portfolio/tripleplay-connect-midi-guitar-controller/



(a) (b)

Fig. 8. Two full setups with (a) laptop and TouchOSC, and (b) the MUSMET-Bridge. (1) Meta Quest 3 HMD, (2) MIDI Keyboard, (3) MIDI Slider controller, (4) Laptop with TouchOSC, (5) the MUSMET-Bridge, (6) power bank.

connection between the selected MIDI device and PatchWorld. Participants were then allowed to play and explore a prepared world for a minimum of three minutes.

After acclimatization, the MIDI device was disconnected, and half of the participants were given written instructions and asked to configure the communication between the MIDI controller and PatchWorld using a Laptop with TouchOSC (Condition 1), while the remaining half started with the proposed MUSMET-Bridge (Condition 2). The process for Condition 1 consisted of the following steps:

1) Taking the laptop out of its case,
2) Powering on the laptop,
3) Connecting the laptop to a WiFi Network,
4) Opening PatchWorld's TouchOSC patch,
5) Configuring MIDI inputs in TouchOSC,
6) Configuring OSC parameters (IP address, ports),

The steps for Condition 2 were the following:

1) Taking the Bridge out of its case,
2) Powering on the Bridge,
3) Connecting the Bridge to a WiFi Network,
4) Configuring OSC parameters (IP address, ports),
5) Connecting the MIDI device,

After the setup, participants were allowed to play with the system for a minimum of three minutes. After setting up the connection under one of the conditions and playing freely with PatchWorld, participants were asked to fill a System Usability Scale (SUS) questionnaire and an ad-hoc questionnaire (see Section IV-D). Then, participants were asked to set up the connection with the remaining condition, and were allowed to play freely once again. Setup with the remaining condition was followed by the SUS and ad-hoc questionnaires.

Once the two setup conditions were completed in Room A, the participant was conducted to a second room (Room B), where the two setup conditions were repeated, each followed by the same two questionnaires. At last, participants underwent an interview and demographics survey.

C. *Quantitative Measures*

Evaluation sessions were videotaped and analyzed to measure the time taken for the following three phases of the setup with each system:

1) **Bootstrap:** Extracting the connecting device (Laptop or Bridge) from its case and powering it on. In the case of the Laptop, this included opening PatchWorld's TouchOSC patch,
2) **WiFi and OSC Configuration:** Connecting the device to an indicated WiFi network and configuring the IP address and port of the HMD,
3) **MIDI Connection:** Connecting the MIDI device and verifying the connection (either in TouchOSC or with the status LED on the Bridge).

D. *Qualitative Measures*

Each setup condition, including repetitions, was followed by a SUS and an ad-hoc questionnaire, resulting in four of each per participant. The SUS questionnaire was administered either in its original English version, or in its Italian translation[13] [29] depending on user preference.

The ad-hoc questionnaire aimed at evaluating the perceived responsiveness and precision of the system, and included the following two questions with a 7-point Likert scale answer:

1) How responsive was the system? (1 = Not at all, 7 = Very Responsive);
2) How precise was the system? (1 = Not at all, 7 = Very Precise).

Participants were informed that responsiveness referred to the reaction speed of the system (i.e., the degree to which latency between a key press or control actuation and the corresponding sound effect was minimized), while precision referred to the accuracy with which control gestures were registered (i.e., absence of missed notes or control actions).

[13] https://pdf.sus.tools/

We conducted a final semi-structured interview to highlight similarities and differences between the two systems, and identify the aspects of each system that were considered the most interesting and helpful, but also the most annoying and problematic.

V. Results And Discussion

In this section, we summarize and discuss the results obtained from our comparative evaluation. Our analysis combines quantitative measurements derived from video analysis and questionnaires, with qualitative insights gathered through questionnaires and final interviews. These complementary methodologies provide a comprehensive assessment of both systems' performance and user experience, as well as an understanding of the strengths and limitations of each approach. The analysis is divided into four main areas: time efficiency, system usability, perceived responsiveness and precision, and participant feedback.

A. Time Efficiency

The average time results for each phase can be found in Fig. 9, along with the total setup time on the right.

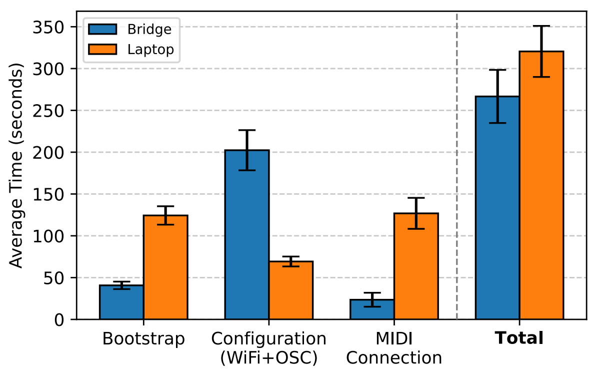


Fig. 9. Average setup time and standard error for each phase.

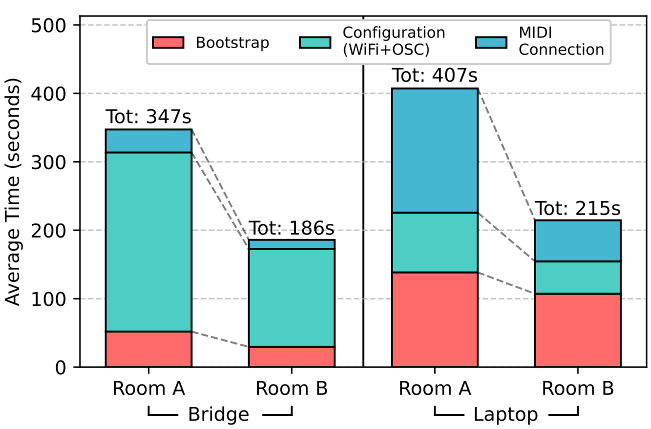


Fig. 10. Per-phase average setup time in Room A and Room B for both Laptop and Bridge.

The Bridge took on average close to one minute (53.9 s) less than the Laptop to set up. The bootstrap and MIDI connection phases were drastically faster with the Bridge (40.8 ± 4.5 s and 23.6 ± 8.4 s) than with the Laptop (124.4 ± 11.0 s and 126.9 ± 18.5 s). On the other hand, configuring the WiFi connection and OSC parameters was considerably slower on the Bridge (202.4 ± 24.0 s) than with the Laptop (69.3 ± 5.9 s).

Observation of the video recordings revealed that the longer Bridge configuration time is to be attributed to (i) understanding the mode-switch mechanism and status LED, and (ii) connecting a smartphone to the configuration webapp.

The decrease in setup time for each phase between Room A and Room B is shown in Fig. 10. For the Laptop, the most affected phase was the MIDI connection, which was reduced due to TouchOSC remembering the MIDI configuration and the evaluation procedure not requiring changing MIDI devices between rooms. For the Bridge, the configuration phase in Room B took on average half the time of Room A, despite the procedure requiring changing both the WiFi network and OSC configuration. This highlights a degree of learning required with the Bridge, which affects the first setup more than the more familiar Laptop setup.

Finally, since configuration is the only phase that is more time-consuming on the Bridge, this indicates that repeated use with the same network (e.g., home use) would further reduce the setup time with the Bridge, while the setup time on the Laptop would still be partially affected by the recurring bootstrap and MIDI connection phases.

TABLE II
System Usability Scale (SUS) average responses with standard error (SE) and average of the final SUS Scores.

Question	Bridge (Mean ± SE)	Laptop (Mean ± SE)
I think that I would like to use this system frequently.	3.5 ± 0.2	3.4 ± 0.3
I found the system unnecessarily complex.	2.5 ± 0.2	2.1 ± 0.2
I thought this system was easy to use.	3.5 ± 0.2	4.0 ± 0.2
I think that I would need the support of a technical person to be able to use this system.	2.1 ± 0.3	2.1 ± 0.2
I found the various functions in this system were well integrated.	3.5 ± 0.2	3.8 ± 0.2
I thought there was too much inconsistency in this system.	1.9 ± 0.2	1.6 ± 0.2
I would imagine that most people would learn to use this system very quickly.	3.5 ± 0.3	3.8 ± 0.3
I found this system very cumbersome to use.	2.4 ± 0.2	2.2 ± 0.2
I felt very confident using this system.	3.2 ± 0.2	3.6 ± 0.1
I needed to learn a lot of things before I could get going with this system.	1.9 ± 0.2	1.8 ± 0.2
SUS Score	66.2 ± 4.3	71.4 ± 3.4

B. System Usability

The responses to the SUS questionnaire are presented along with the average SUS scores in Table II. The average SUS

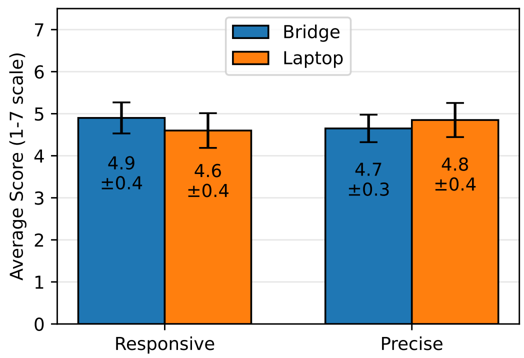


Fig. 11. Results of the ad-hoc questionnaire.

scores between Bridge and Laptop are comparable (66.2 ± 4.3 and 71.4 ± 3.4, respectively), with a slight preference for the Laptop. Despite being willing to use the system frequently, which was considered portable and convenient to place, users generally found it more complex to set up. The Laptop received marginally higher ratings for ease of use and user confidence, although it was a system with a familiar interface for everyone, with average scores that were 0.4-0.5 points higher than the Bridge.

C. Perceived Responsiveness and Precision

The ad-hoc questionnaire administered after each setup stage revealed generally low perceived responsiveness and precision. However, assigned scores (in a 7-point Likert scale) were virtually identical for the Laptop and the Bridge: as can be seen in Fig. 11, the Bridge received 4.9 ± 0.4 for responsiveness and 4.7 ± 0.3 for precision, while the Laptop 4.6 ± 0.4 and 4.8 ± 0.4, respectively.

D. Participant Feedback

Participants' interviews were analyzed using inductive thematic analysis [30]. After a process of open coding performed by three of the authors, we identified three main themes based on the most common shared patterns and commonalities.

*1) **Setup and Connectivity**:* All participants considered the Laptop's main advantage to be its general-purpose and familiar nature, since it is often already part of a musician's workflow, not only for managing music production software (e.g., Pure Data, Reaper, Ableton Live), but also for other activities such as checking emails, writing, and browsing the Internet. Initial WiFi setup for the Laptop was generally perceived as more "*familiar*", as users were accustomed to standard computer interfaces, since "*I know where the menu is, I know how to do it*". In contrast, the process of configuration for the Bridge was identified by seven participants as the weakest point of the system and a source of frustration. This was especially evident in how users had to switch the device to its "Configuration-mode": *"the little box has so many steps between powering on and being able to select a network and enter the WiFi password"*.

Despite these issues, the webapp was considered intuitive and simple to use by six participants because all necessary configuration fields (e.g., IP, port) are presented on a single page, without the need to navigate through different menus, as is the case of TouchOSC. As a participant noticed *"the Bridge is nice because you make a unified configuration for everything. A very simple interface (webapp, A/N), and then you don't worry about it anymore."*

After configuring the networking component of the Bridge, connecting MIDI devices became simple and "*plug-and-play*" for most participants, abstracting technical details and providing a unified configuration experience: *"The Bridge is very comfortable, you plug it in and everything works"*. Conversely, configuring MIDI on the Laptop and especially in TouchOSC required manually navigating menus and explicitly selecting ports.

*2) **Design and Portability**:* The Bridge's compact size and portability were considered significant advantages over the Laptop by the majority of participants. Especially, its pocket-sized design eliminates cable clutter, making it easier to position on a table. As a participant mentioned *"in fact you can literally carry it in your pocket."*

Half of the participants have especially appreciated that the Bridge could run on a power bank, making it much more mobile than the Laptop, which is larger, heavier, and typically needs to be plugged into an outlet. However, three participants expressed a desire for the Bridge to have an integrated internal battery, eliminating the need for a separate power supply.

*3) **Feedback and Control**:* Half of the participants appreciated the visual feedback provided by the screen of the Laptop and the interface of TouchOSC, allowing users to confirm actions quickly and have more control in different scenarios. As a participant noticed, with the Laptop "*I am in control of all things and if something happens I easily know where to put my hands"*, while another participant pointed out that with "*the little box, I think, is a little bit more difficult to modify on the fly.*" However, the Bridge's single LED indicator was a significant drawback, frequently confusing users who struggled to understand whether the device was in "Configuration-mode" or "Play-mode", or what the connection status was.

Therefore, seven participants expressed a desire for better visual feedback, possibly a small screen or at least clearer indicators, such as *"a second light or something to indicate the mode.[...] Because the blinking happens quite slowly."*

VI. Conclusions

This paper presented MUSMET-Bridge, a standalone embedded system designed to streamline the integration of physical MIDI control surfaces with Musical Metaverse environments.

The evaluation showed that the Bridge offers significant practical advantages over conventional laptop-based systems. With the Bridge the setup time for bootstrap and MIDI connection phases was substantially reduced, while networking configuration required more time due to the learning curve associated with the device's LED-based mode-switching mechanism. Despite this, repeated use showed substantial time reductions, suggesting that the Bridge becomes more efficient with familiarity.

However, the mechanism used for mode-switching creates a usability bottleneck that could benefit from a better design. By adding clearer mode indicators or small display screens to streamline the process of mode-switching, it will be possible to eliminate the configuration time advantage that laptop-based solutions currently maintain.

Especially, participants consistently highlighted that portability is one of the main advantages of MUSMET-Bridge. The small size of the device and the possibility of running on standard power banks were considered particularly valuable for musicians who need to transition between different spaces, such as the studio and stage.

Through the work presented in this paper, we showed how embedded devices, such as MUSMET-Bridge, could represent a significant step toward the integration of common practices and workflows of electronic musicians (e.g., using MIDI and DAWs) with immersive virtual environments. As standalone XR headsets become increasingly widespread, the need for similarly standalone peripheral connectivity solutions for musicians becomes more pressing.

Future enhancement of the MUSMET-Bridge should focus on the integration of improved user interfaces, enhanced wireless protocols, and advanced power management to better refine the balance between portability, functionality, and ease of use.

Acknowledgment

This work has been supported by the MUSMET project funded by the European Innovation Council (grant n. 101184379). G. A. Giudici's contributions are funded by the European Union (EU) under NextGenerationEU (M.D. 118/2023). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the EU or the European Research Executive Agency. Neither the EU nor the granting authority can be held responsible for them.

References

[1] R. Hamilton, "Collaborative and competitive futures for virtual reality music and sound," in *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2019, pp. 1510–1512.

[2] L. Men and N. Bryan-Kinns, "LeMo: Supporting Collaborative Music Making in Virtual Reality," in *Proceedings of the IEEE VR Workshop on Sonic Interactions for Virtual Environments (SIVE)*, 2018, pp. 1–6.

[3] A. Boem, D. Dziwis, M. Tomasetti, S. Etezazi, and L. Turchet, "It takes two - shared and collaborative virtual musical instruments in the musical metaverse," in *Proceedings of the IEEE International Symposium on the Internet of Sounds (IS2)*, 2024, pp. 1–10.

[4] M. Buffa, D. Girard, and A. Hofr, "Using web audio modules for immersive audio collaboration in the musical metaverse," in *Proceedings of the IEEE International Symposium on the Internet of Sounds (IS2)*. IEEE, 2024, pp. 1–10.

[5] S. Serafin, C. Erkut, J. Kojs, N. C. Nilsson, and R. Nordahl, "Virtual reality musical instruments: State of the art, design principles, and future directions," *Computer Music Journal*, vol. 40, no. 3, pp. 22–40, 2016.

[6] L. Turchet, R. Hamilton, and A. Çamci, "Music in extended realities," *IEEE Access*, vol. 9, pp. 15 810–15 832, 2021.

[7] F. Berthaut, "3d interaction techniques for musical expression," *Journal of New Music Research*, vol. 49, no. 1, pp. 60–72, 2020.

[8] A. Boem, M. Tomasetti, and L. Turchet, "Issues and challenges of audio technologies for the musical metaverse," *J. Audio Eng. Soc*, vol. 73, no. 3, pp. 94–114, 2025.

[9] J. Desnoyers-Stewart, D. Gerhard, and M. Smith, "Mixed reality midi keyboard," in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, Sep 2017.

[10] D. Johnson and G. Tzanetakis, "VRMin: Using Mixed Reality to Augment the Theremin for Musical Tutoring," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2017, pp. 151–156.

[11] J. Desnoyers-Stewart, D. Gerhard, and M. Smith, "Augmenting a midi keyboard using virtual interfaces," *Journal of the Audio Engineering Society*, vol. 66, pp. 439–447, Jun 2018.

[12] D. Johnson, D. Damian, and G. Tzanetakis, "Evaluating the effectiveness of mixed reality music instrument learning with the theremin," *Virtual Reality*, vol. 24, no. 2, pp. 303–317, 2020.

[13] L. Turchet, "Musical metaverse: vision, opportunities, and challenges," *Personal and Ubiquitous Computing*, vol. 27, no. 5, pp. 1811–1827, 2023.

[14] A. Mulder and S. Fels, "Sound sculpting: Manipulating sound through virtual sculpting," in *Proceedings of the Western Computer Graphics Symposium*. Citeseer, 1998, pp. 15–23.

[15] T. Mäki-Patola, J. Laitinen, A. Kanerva, and T. Takala, "Experiments with virtual reality instruments," in *Proceedings of the Conference on New interfaces for Musical Expression (NIME)*, 2005, pp. 11–16.

[16] R. Ishino and N. Tokui, "Miami: A Mixed Reality Interface for AI-based Music Improvisation," *International Conference on New Interfaces for Musical Expression (NIME)*, jun 2022.

[17] V. Zappi, M. Gaudina, A. Brogni, and D. Caldwell, "Virtual sequencing with a tactile feedback device," in *Proceedings of the International Workshop of Haptic and Audio Interaction Design (HAID)*. Springer, 2010, pp. 149–159.

[18] F. Berthaut and M. Hachet, "Spatial interfaces and interactive 3d environments for immersive musical performances," *IEEE Computer Graphics and Applications*, vol. 36, no. 5, pp. 82–87, 2016.

[19] A. Çamcı and J. Granzow, "Hyperreal instruments: Bridging vr and digital fabrication to facilitate new forms of musical expression," *Leonardo Music Journal*, vol. 29, pp. 14–18, 2019.

[20] A. Boem and H. Iwata, "Encounter-type haptic interfaces for virtual reality musical instruments," in *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces*, 2018, pp. 1–2.

[21] D. Dziwis and A. Hadjakos, "Interface modules for extended reality in music," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2024, pp. 439–446.

[22] P. P. Lucas Bravo and S. Fasciani, "A human-agents music performance system in an extended reality environment," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2023, pp. 10–20.

[23] M. Wright, A. Freed *et al.*, "OpenSound Control: A new protocol for communicating with sound synthesizers," in *Proceedings of the International Computer Music Conference (ICMC)*, 1997.

[24] A. Fraietta, "Open sound control : Constraints and limitations," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2008, pp. 19–23.

[25] Hexler, "TouchOSC," 2025, accessed: 2025-06-23. [Online]. Available: https://hexler.net/touchosc

[26] J. P. M. Avila, C. Greenhalgh, A. Hazzard, S. Benford, and A. Chamberlain, "Encumbered Interaction: a Study of Musicians Preparing to Perform," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. ACM, 2019, p. 1–13.

[27] A. Ng, S. A. Brewster, and J. H. Williamson, "Investigating the effects of encumbrance on one-and two-handed interactions with mobile devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014, pp. 1981–1990.

[28] PatchXR, "Patchworld," 2025, accessed: 2025-06-23. [Online]. Available: https://patchxr.com

[29] S. Borsci, S. Federici, and M. Lauriola, "On the dimensionality of the System Usability Scale: a test of alternative measurement models," *Cognitive Processing*, vol. 10, no. 3, pp. 193–197, Aug 2009.

[30] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, pp. 77–101, 01 2006.