From Push Buttons to Notes: A Hardware/Software Ecosystem for Inclusive Music Education

Luca Andrea Ludovico[1][a], Vanessa Faschi[1][b], Federico Avanzini[1][c], Emanuele Parravicini[2] and Manuele Maestri[3]

[1]*Laboratory of Music Informatics (LIM), Dept. of Computer Science, University of Milan, via G. Celoria 18, Milan, Italy*

[2]*Audio Modeling, Via Bernardino Luini 65, Meda, Italy*

[3]*Musica Senza Confini, Italy*

Keywords: Accessible Music Education, Push-Button Interfaces, Human-Computer Interaction (HCI), Inclusive Music Learning, Assistive Technology.

Abstract: This paper explores several ways to drive a music-oriented computer system by push-button controls, with a particular focus on music education for young children and individuals with disabilities. The research investigates a range of interaction paradigms where heterogeneous push-button actions can be mapped onto musical functions, such as triggering Note-On/Note-Off events, dynamically controlling other musical parameters, or playing and stopping pre-recorded sequences. The ultimate goal is to propose a hardware/software ecosystem that utilizes button-based human-computer interfaces that are not specialized for music (e.g., joypads or colored computer keyboards). These paradigms are designed to lower the barrier to entry for engaging with music, making it accessible even to those with limited motor skills or no prior musical training. To this end, we propose an implementation where multiple push-button devices can be connected to a hub that communicates with a computer, and the role of the latter is to associate a customizable musical meaning to button events in the framework of inclusive music education.

1 INTRODUCTION

In this paper, we propose a computer-based approach to address the educational needs of pre-school-age children and people with different types of disabilities, both cognitive and physical. In particular, we focus on the potential of button-based interfaces, analyzing the range of actions that a user can perform and a system can consequently detect. After associating a musical meaning to such actions, we propose an accessible and inclusive HW/SW platform capable of combining different peripheral devices to control a music production system, e.g. a synthesizer.

Analyzing the potential of a button interface is valuable for several reasons. First and foremost, buttons are ubiquitous in everyday life, playing a crucial role in numerous contexts, such as calling an elevator or activating a pedestrian traffic light. This familiarity makes them immediately recognizable and eliminates the need for extensive explanation or instruction.

Secondly, their intuitive design is particularly advantageous in accessibility contexts. Buttons require minimal cognitive effort to understand and operate, making them suitable for individuals with cognitive disabilities. Suffice it to say that a specific type of button interface, known as "mushroom push button", is commonly used as an emergency stop or panic switch. Furthermore, simplicity and ease of use make buttons accessible to many people with physical impairments, including those with limited dexterity or strength.

Moreover, narrowing the field to music education, button interfaces are particularly relevant because they align with "traditional" musical controls. Examples include keys on keyboard and wind instruments, effect pedals, and drawknob stops for organ registers. Buttons are also typically available on the user interfaces of media players and synthesizers. These familiar controls not only simplify the transition to digital or adaptive musical tools but also make the learning process more intuitive by building on ex-

[a] https://orcid.org/0000-0002-8251-2231
[b] https://orcid.org/0000-0002-9815-1127
[c] https://orcid.org/0000-0002-1257-5878

Ludovico, L. A., Faschi, V., Avanzini, F., Parravicini, E. and Maestri, M.
From Push Buttons to Notes: A Hardware/Software Ecosystem for Inclusive Music Education.
DOI: 10.5220/0013489300003932

In *Proceedings of the 17th International Conference on Computer Supported Education (CSEDU 2025) - Volume 1*, pages 650-660
ISBN: 978-989-758-746-7; ISSN: 2184-5026

isting knowledge. This connection to established musical practices highlights the versatility and accessibility of button-based designs, making them an excellent choice for fostering engagement and inclusion in music education settings.

Lastly, buttons are widely available and relatively inexpensive, which contributes to their practicality and feasibility in various applications. Their cost-effectiveness makes them an ideal choice for projects aiming to create inclusive and affordable solutions, ensuring broader access and usability.

Push-button interfaces have been widely discussed in the scientific literature, exploring this control device's historical, anthropological, and technological implications. To mention but a few works, Plotnick traced the origins of today's push-button society by examining how buttons have been made, distributed, used, rejected, and refashioned throughout history, specifically focusing on the period between 1880 and 1925 (Plotnick, 2018). Gaspar *et al.* analyzed the design requirements of buttons and their relations with ergonomics (Gaspar et al., 2019). The relevance of this type of interface is also witnessed by research trying to reconstruct or simulate push actions in alternative ways, such as ad-hoc haptic interfaces (Allotta et al., 1999; Chen et al., 2022; Morimoto et al., 2007) or touchless interactions with public displays (Gentile et al., 2016). Of particular interest to our study, due to its proximity to sound-related aspects, is a research work that investigates the relationship between the signal properties and the perceptual attributes of everyday push-button sounds (Altinsoy, 2020).

The rest of the paper is structured as follows: Section 2 presents a range of possibilities offered by button interfaces, Section 3 applies such a paradigm to music performance, Section 4 showcases state of the art, Section 5 describes an example of hardware and software ecosystem, Section 6 exemplifies some educational use cases, and, finally, Section 7 draws the conclusions.

2 ACTIONS AND EVENTS USING A PHYSICAL BUTTON

The interaction with a single physical button through a computer system involves various types of actions and events, each with its unique characteristics, applications, and implementation details. These actions can be categorized into basic, advanced, and derived actions, reflecting their complexity and potential use cases. Table 1 provides a synoptic overview of the button actions described in this section.

2.1 Basic Actions

Basic actions represent the fundamental interactions with a button. The most straightforward action is the *button down* event, which occurs when the button is pressed down, initiating contact. Implementation requires detecting the state change from "up" to "down" as soon as it happens, ensuring low latency and accuracy.

Conversely, the *button up* event occurs when the button is released after being pressed. This event often signals the end of an action started by a *button down* event. It is essential to detect this transition promptly to avoid inconsistencies in the user experience.

A combination of these two events forms the *button click*, a quick press and release. This is widely used for discrete actions such as selecting items in a menu or triggering a one-time function. The implementation must ensure that both the press and release occur within a predefined time interval, avoiding confusion with other actions like holding or multi-clicking.

The *button hold* action occurs when the button is pressed and held for a sustained duration. It is typically employed for continuous actions or mode switching, such as dimming a light or activating a special feature. Implementation requires measuring the duration of the press and triggering the appropriate response only when the hold time exceeds a certain threshold.

2.2 Advanced Actions

Advanced actions extend the capabilities of button interaction by introducing more complex patterns or timing-based logic.

A *double click*, for instance, involves two quick successive presses and releases. This action is commonly mapped to alternate functions, such as opening a file instead of selecting it. Detecting a double click requires timing logic to ensure that the interval between clicks falls within a predefined range.

A related but more complex action is the *multi-click*, where three or more quick successive presses are detected. This can unlock advanced features, such as resetting a device or triggering specific modes. The implementation challenge lies in accurately detecting the intended number of clicks without interference from noise or user input variations.

The *hold and release* action combines a sustained press with a release event. This allows for time-dependent actions, such as activating a special function if the button is held for more than n seconds. Accurate timing measurements and threshold checks are essential for this implementation.

In the previous case, the value of n is usually low, in the order of a few seconds. The *long hold* action is a specific variation where the button is held for an extended duration, often exceeding 5 seconds. This action is reserved for critical functions such as performing a factory reset or enabling a secure mode. Differentiating between a short and long hold requires careful calibration of timing thresholds to prevent accidental activations.

Finally, *press with modifiers* involves combining a button press with other inputs, such as simultaneous presses of additional buttons. This enables complex interactions, such as keyboard shortcuts. Implementation requires monitoring the states of multiple inputs and coordinating their timing.

2.3 Derived Actions

Derived actions leverage additional hardware capabilities or sophisticated input patterns to provide enhanced functionality. One example is *press and rotate*, which occurs when a button integrated with a rotary encoder is pressed while being turned. This action is ideal for precise adjustments, such as navigating menus or fine-tuning settings. Synchronizing the signals from the button press and the rotary encoder is crucial for effective implementation.

Another derived action is *pressure-sensitive interaction*, where varying levels of pressure applied to the button are detected. This is particularly useful in contexts requiring dynamic control. However, implementing this action requires specialized hardware capable of detecting pressure levels.

The *sequential presses* action involves a defined sequence of presses, such as press-release-press. This is often used for unlocking specific actions, such as entering a password or toggling modes. Accurate tracking of the input sequence and timing is necessary to differentiate valid patterns from unintended presses.

2.4 Implementation Considerations

Across all these actions, certain considerations are universal. First, *debouncing* is critical to mitigate false detections caused by electrical noise or mechanical imperfections in the button. This activity can be performed by either hardware or software components in order to filter out unintended state changes, thus ensuring reliable input recognition.

Second, focusing on user experience, the balance between simplicity and complexity is vital. While basic actions are intuitive and straightforward to implement, understand, and perform, advanced and derived actions provide greater versatility at the cost of additional implementation complexity, user learning, and concentration.

The actions we have described so far focus on the use of a single button. The availability of multiple buttons would enable a wide range of applications. An example is provided by the already-mentioned action called *press with modifier*. Even the interfaces of traditional musical instruments, such as keyboard instruments, can be seen as combinations of buttons: in the case of a piano, the keys are sensitive to activation (with variations in pressure) and release, and the pedals can be considered switch controllers.

Finally, the application context greatly influences the design of these interactions. For instance, accessibility-focused systems benefit from simple and easily distinguishable actions, while musical performance systems require precise timing and responsiveness. By tailoring the implementation to the specific use case, physical button interactions can be optimized for both functionality and usability. The adaptation of the button paradigm to inclusive music education will be addressed in the following section.

3 BUTTON INTERFACES FOR INCLUSIVE MUSIC EDUCATION

The button actions described in Section 2, forming the core of user interaction in music production environments, can be employed in educational settings to support music learning. These actions offer a wide range of functionalities that can be adapted to suit the needs of learners. Understanding how these button actions can be used, along with considerations for accessibility and inclusivity, is crucial for ensuring that all users can engage with music technology effectively. In the remainder of this section, we will begin by examining single-button actions and then explore the possibilities enabled by the simultaneous use of multiple buttons.

3.1 Single Button

When considering music production, *button down* can trigger musical events, such as playing a note on a traditional instrument or a MIDI keyboard. When a student presses a key, they may hear a corresponding sound, which allows them to experiment with pitch, timbre, and harmony. This basic interaction is an

Table 1: Actions and Events Using a Physical Button. The Categories (from Top to Bottom) Refer to Basic Actions, Advanced Actions, and Derived Actions, Respectively.

Action	Description
Button Down (Press)	The button is pressed down, initiating contact.
Button Up (Release)	The button is released after being pressed.
Button Click	A quick press and release of the button.
Button Hold	The button is pressed and held for a sustained period.

Action	Description
Double Click	Two quick successive presses and releases of the button.
Multi-Click	Three or more successive quick presses and releases.
Hold and Release	The button is held down for a set duration before release.
Long Hold	The button is held for an extended duration (e.g., more than 5 s).
Press with Modifiers	Combining a button press with other inputs (e.g., simultaneous button presses).

Action	Description
Press and Rotate (with Encoders)	Pressing a button integrated with a rotary encoder while turning it.
Pressure-Sensitive Actions	Detecting varying levels of pressure applied to a button.
Sequential Presses	A defined sequence of presses (e.g., press-release-press).

intuitive way for beginners to get started with creating music and experimenting with sounds. In music education, *button down* actions can also be used to trigger pre-recorded sequences or exercises in music software. For example, pressing a button could start a metronome or initiate an exercise designed to practice rhythm or note recognition. For students with physical disabilities, particularly those with limited dexterity, the button press should be designed to require minimal effort, possibly utilizing larger or softer buttons with customizable sensitivity. This ensures that learners with motor impairments can still engage with the content without feeling overwhelmed by the physical demands of the interaction. Such a consideration can be extended to all button actions described below.

The *button up* action is usually associated with the end or completion of an action, such as releasing a key on a keyboard controller. When applied to sound production, this action is essential for defining the duration of notes (even if a delay effect or controls such as the sustain pedal can prevent the sudden release of sound). In this sense, button release can be used to teach concepts like note length and rhythm. Considering *button up* as the counterpart of an action started by a *button down* event, such an action could also stop the reproduction of an audio file or the synthesis of a note sequence previously started. Concerning accessibility, software or hardware could offer adjustable settings for note release time to assist students with motor impairments, helping users to adhere to a predefined time grid or release the button in a controlled manner.

The *button click* action is used extensively in user interfaces, and, consequently, also in music production and education software, often performed through a mouse button or a keystroke. In this sense, use cases are endless: for example, in a digital audio workstation (DAW), clicking a button might trigger a play/pause event or allow the user to select an instrument or change a sound preset; in educational applications, such an action could trigger interactive lessons, quizzes, or feedback on musical tasks, such as identifying scales or intervals. But the scenarios described so far do not refer to the musical meaning of the *click* action, rather they showcase educational environments that can be controlled, like almost any software, by keyboard and mouse. Rather, focusing on the musical meaning of a *button click*, this action could be associated with the production of impulsive sounds and rhythmical patterns or the response to a stimulus through a prompt reaction. A sequence of *button clicks* can also be employed as an alternative to *button down*/*button up* or *button hold*: for example, instead of triggering a note attack via *button down* and releasing the sound via *button up*, the user could perform two distinct *button clicks*. In terms of accessibility, buttons should be clearly labeled, with visual or auditory cues to confirm the action, so that students with visual impairments or learning difficulties can easily navigate the interface. The use of *button clicks* as an alternative to *button down*/*button up* could be helpful when a hold action is too demanding.

The *button hold* action involves pressing and holding a button for a longer period. In music production, this can be used, e.g., to sustain a note, much like a sustain pedal on a traditional keyboard, allowing for expressive control over musical performance. In music education, a button hold could be employed to teach students about the concept of sustained notes or to control the duration of a note in an exercise. This action is particularly beneficial in the context of limited fine motor control, as it may be easier to sustain a button press rather than perform quick taps.

In general terms, the family of advanced actions (see Section 2) can be invoked in a more advanced training phase. Some actions, such as *double click*, *long hold* and *press with modifiers*, can be physically or cognitively demanding. On the other hand, one of the advantages of adopting simple interfaces such as buttons lies in the possibility to tailor activities based on the student's current abilities and the expected skills; in this sense, advanced actions, if properly calibrated, can represent engaging challenges. Moreover, for users with physical disabilities, there are customization options allowing for slower or alternative input methods such as voice commands.

The *hold and release* action is an advanced type of interaction that can be used to control continuous parameters, such as modulation depth or volume. For instance, holding a button might gradually increase the volume of an audio track until the button is released, returning the volume to its original state. In music education, this kind of interaction could be used to teach students about dynamic changes in music, such as crescendo and decrescendo. Accessibility features should include adjustable duration thresholds for holding the button, ensuring that users with motor impairments can control the action effectively.

As mentioned in Section 2, derived actions are more specialized interactions that combine multiple input types. Even if potentially powerful in controlling multiple dimensions simultaneously, these actions can be perceived as challenging to perform from a motor perspective (consider scenarios involving precise control of pressure or rotation) and complex to learn and associate from a cognitive perspective. One example is *press and rotate*, where a button is pressed while being rotated. This is commonly used to adjust parameters like volume or filter cutoff in music production. In educational contexts, press-and-rotate controls can help students engage with interactive exercises that involve adjusting multiple parameters simultaneously, such as tuning a virtual instrument or adjusting sound properties in real time.

Pressure-sensitive buttons are another example of derived actions, where the pressure applied to a button affects the outcome. In music production, pressure-sensitive pads are commonly used to control the velocity of notes played on a MIDI controller, allowing for dynamic control over the intensity of sound. This provides musicians with the ability to add expression to their performance. In music education, pressure-sensitive buttons could be used to help students understand concepts like dynamics (soft and loud) and articulation. To ensure accessibility, pressure-sensitive buttons should allow for customization in terms of sensitivity, making it easier for students with different levels of motor control to interact with the system.

In general, sensory limitations can play a role. Users with reduced tactile sensitivity might struggle to distinguish between buttons or confirm if they have been pressed correctly, while those with visual impairments may face difficulties if the buttons or their states are not clearly visible.

3.2 Multiple Buttons

In an accessibility context, requiring the use of multiple buttons for performing actions can present various challenges for users with disabilities. Motor coordination can be a significant issue, as some users may lack the fine motor skills needed to press multiple buttons simultaneously or in quick succession. Those with limited mobility, such as individuals with conditions like cerebral palsy or paralysis, may find it particularly difficult to operate multiple buttons at once. Additionally, reduced finger strength or dexterity can make pressing or holding buttons challenging.

From a cognitive perspective, the complexity of remembering sequences or combinations of button presses can be overwhelming for users with memory impairments or cognitive disabilities. This increased complexity also raises the likelihood of errors, which may lead to frustration or even task abandonment. Environmental factors can exacerbate these difficulties. For instance, buttons requiring simultaneous use may be physically out of reach for some users or may demand excessive or uneven force to operate, adding to the challenge. Usability concerns, such as the potential for fatigue from repetitive or prolonged multi-button actions, are also significant. Precision requirements for pressing multiple buttons correctly can further diminish accessibility and lead to frustration.

The challenges described here, if appropriately adapted under the guidance of an expert, can be transformed into opportunities for learning, rehabilitation, and inclusive music-making. It is essential that the goals are tailored to the individual's abilities and that the tasks do not lead to frustration.

Some advanced actions mentioned above fall under the umbrella of multiple-button availability. For example, we can trace back *press with modifiers* to the combined use of independent buttons. In a traditional setting, playing a chord on a piano can be seen as a press (the root note) with modifiers (the other notes in the chord) action; still, the mentioned elements are independent keys whose role (either the main button or a modifier) can change during the performance. In music education, this approach can be used, e.g., to learn the principles of harmony, but, similarly to other advanced actions, it requires dexterity and can be cognitively challenging.

Sequential presses involve pressing multiple buttons in a specific order to trigger an event. In music production, this could be used to activate a series of effects or sequences, such as arpeggios or drum patterns. In music education, sequential presses can be used in exercises that teach students about musical structure and sequence, such as arranging notes or completing rhythm patterns. Accessibility considerations include allowing for flexible timing between presses and providing feedback to indicate whether the sequence was completed correctly.

Finally, let us mention the scenario where multiple buttons are used by independent players to perform basic, advanced, or even derived actions. This approach can mitigate the usability issues discussed above. Furthermore, collaborative performances foster inclusivity by encouraging social interaction, teamwork, and mutual understanding among participants of diverse abilities and backgrounds. They empower individuals to express themselves, contribute meaningfully, and feel valued within a group.

4 STATE OF THE ART

As mentioned above, the interfaces of many "traditional" musical instruments and sound equipment implement forms of interaction that may recall button actions. But, in this section, we want to address computer-based approaches that explicitly rely on the button paradigm. Due to their interface, such tools fall under the wider umbrella of tangible user interfaces (TUIs), which provide innovative ways for users to interact with digital systems through physical objects (Ishii, 2007). In the context of music production and learning, music TUIs enable more intuitive and creative musical experiences (Baratè and Ludovico, 2024). By bridging the gap between tactile interaction and digital sound processing, music TUIs foster a deeper connection between the performer and the music, often with significant benefits for accessibility and education.

One example of a music-focused TUI is the *Kibo* by Kodaly,[1] a MIDI Bluetooth instrument crafted from solid maple wood (Amico and Ludovico, 2020). The system includes a modular keyboard composed of eight distinct magnetic shapes, each representing a different musical note or control parameter (see Figure 1). Users can rearrange these shapes to create custom musical phrases. Through its dedicated app, the *Kibo* translates these physical interactions into musical output, not only triggering notes but also offering features such as tone adjustments, scale selection, and octave control. This integration of tangible and digital elements makes the *Kibo* particularly effective in music education, especially for young children and individuals with disabilities (Baratè et al., 2023). The *Kibo* implements several button actions: its eight tangibles are pressure-sensitive, thus supporting *button down*, *button up*, *button click*, *button hold*, and *pressure-sensitive interaction*. Moreover, the board has a knob control to perform *press and rotate* action. All these controls produce standard messages via the MIDI protocol and can be associated with different musical parameters.

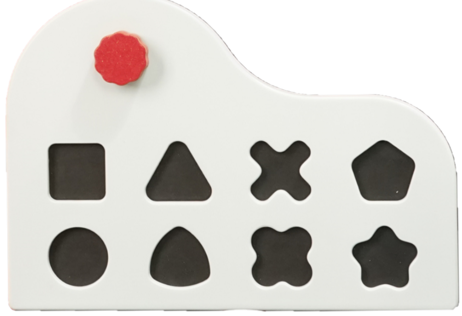

Figure 1: The *Kibo* by Kodaly.

The *Skoog* by Playable Technology is a tactile, cube-shaped musical interface designed for accessibility (see Figure 2). Users can press, squeeze, or tap the *Skoog* to produce sound, which is mapped to various instruments or MIDI controls. The device connects to apps that allow users to select scales, keys, and instruments, making it a versatile tool for inclusive music education (Rinta, 2019). The *Skoog* can also be used as a simple 5-button communicator when paired with an iPad. Its soft, responsive surface makes it particularly suitable for children, individuals with motor impairments, and those new to music-making. The *Skoog* implements *button down*, *button up*, *button click*, and *button hold* actions. Unfortunately, this project has recently been discontinued.[2]

AudioCubes by Percussa[3] are wireless cubes that use light sensors to control sound and music. When cubes are moved, rotated, or placed near each other, they send MIDI or OSC signals to music software, enabling dynamic and visually engaging performances. All the faces are equipped with proximity sensors capable of detecting other objects, hands included; in

[1] https://www.kodaly.app/

[2] https://www.playable.tech/

[3] https://www.percussa.com/what-are-audiocubes/

Figure 2: The *Skoog*.

this way, each *AudioCube* can turn into a multiple-button interface, also sensitive to pressure. This modular system provides an open-ended platform for exploring composition and live performance, encouraging experimentation through tactile interaction.

The *Soundplane Model A* by Madrona Labs[4] is a tactile music interface that combines the expressiveness of a stringed instrument with the capabilities of a MIDI controller. It can detect a wide range of touches on its playing surface, from a light tickle to a very firm press. While it is not a traditional button-based interface, it has been chosen to exemplify the tactile qualities that define TUIs in music.

The music-focused TUIs mentioned above demonstrate the potential of tangible interaction to enhance creativity, accessibility, and engagement in music-making. These systems encourage exploration and learning by offering immediate feedback and a hands-on approach to sound manipulation. TUIs break down barriers to music education, making it accessible to people of all ages and abilities. Furthermore, these devices can form an ensemble, operating with other identical or similar tools, electronic equipment, or traditional instruments. Music-making's social and connective functions are fundamental to fostering interaction, integration, and cooperation among learners (Baratè et al., 2021; Frid, 2019; Samuels and Schroeder, 2019).

5 A PROPOSAL FOR A HW/SW ECOSYSTEM

In the previous sections, we highlighted the various actions that can be performed and detected, as well as the importance of push-button devices as interfaces for music creation and learning. We now propose a hardware and software ecosystem designed to implement the control of musical parameters using this approach and leveraging readily available devices, such as simplified keyboards and gamepads. Clearly, the platform could also include custom-made devices, provided they can communicate in a standard manner, for instance, via USB ports.

Figure 3 depicts the ecosystem designed for inclusive music education. Its most relevant component is on the right: a central hub that connects various push-button interfaces (or similar devices). These interfaces are easily accessible and controllable, making them suitable for individuals with physical or cognitive impairments. Moreover, it is highly probable that such interfaces, e.g. joypads, are already available to users and commonly adopted in everyday life; this implies that their functions and interfaces are already known. If not available, these devices can be purchased at a low cost. The communication between peripheral devices and the hub can occur via USB, MIDI, or other commonly-accepted protocols.

In our proposal, a variety of devices can play the role of the hub. An example is provided by the *Microsoft Xbox Adaptive Controller*, an innovative gaming device designed to make gaming accessible for individuals with disabilities. Released in 2018, it features a flat rectangular design with numerous ports for connecting external devices (see Figure 4). This modular system allows users to customize their experience by attaching a wide range of compatible switches, buttons, joysticks, and other assistive devices. The controller supports connectivity with Xbox consoles and Windows PCs via USB-C or Bluetooth, enabling seamless integration into various setups.

The computer system that receives messages from the hub runs software capable of giving them a customizable musical meaning and routing them to a DAW, a synthesizer, or another sound system using standard protocols like MIDI or OSC (Wright, 2005).

The gray section on the left represents optional devices for additional inclusivity, such as eye trackers and wireless motion detectors. However, the primary focus of the current work remains on the push-button interfaces.

Please note that such an ecosystem has already been implemented, released under the name *Inclusive MIDI Controller*, and tested in an observational study presented in (Faschi et al., 2024). In that work, the focus was on the accessibility of the central part of the diagram, namely the software interface gathering messages from peripheral devices, rather than the adoption of push buttons to foster inclusivity. Moreover, the aim was to enable participation in creative processes and musical performance, while here we are mainly addressing music education.

[4] https://www.madronalabs.com/soundplane

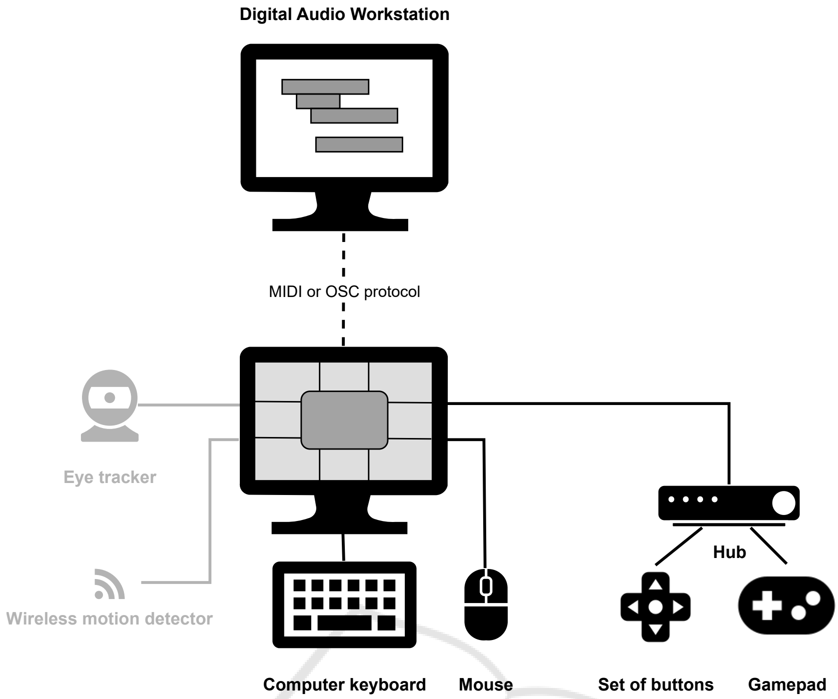

Figure 3: Schema of the HW/SW ecosystem.

6 USE CASES

In this section, we will propose some educational activities aimed at different types of disabilities and fully relying on button interfaces. By choice, we will not present specially designed hardware, in order to make these experiences as replicable as possible in educational contexts and at relatively low costs. Clearly, the ability to use custom interfaces would expand the possibilities: consider, for example, 3D-printed buttons featuring Braille coding of functions, designed for Blind and Visually Impaired (BVI) users. Rather, we will refer to simplified button interfaces connected to the hub in Figure 3.

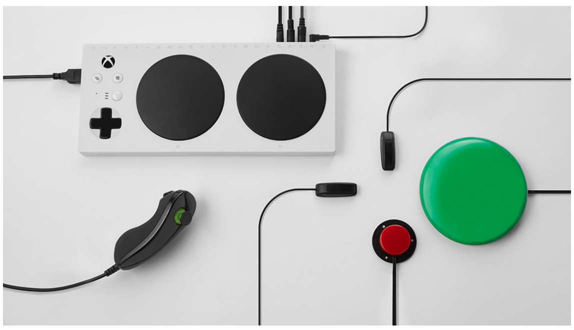

Figure 4: The *Microsoft Xbox Adaptive Controller*.

6.1 Triggering Simple Events

In this use case, the type of detected events is *button click*, eventually prolonged. As mentioned above, it is the simplest form of interaction; due to its widespread use in everyday life, we can expect that all users know this action, even if they may experience difficulties in reproducing it or understanding what its musical meaning is.

Button clicks can be detected by gamepads, colored keyboards, or other accessible devices. An old joystick had two buttons, while a modern gamepad contains at least 4 buttons plus navigation controls. To mention a more advanced device, the *Microsoft Xbox 360 Controller* features:

- Directional Pad, or D-Pad – a four-directional pad to control movement via *Up*, *Down*, *Left*, and *Right* buttons;
- Action buttons – 4 buttons, named *A*, *B*, *X*, and *Y*, positioned on the upper side of the controller;
- Shoulder and Trigger Buttons – *Left Bumper* (LB) and *Right Bumper* (RB), located on the top left and top right of the controller, plus *Left Trigger* (LT) and *Right Trigger* (RT), located below;
- System and Menu Buttons – *Start*, *Back*, and *Xbox Guide* buttons.

By connecting a gaming peripheral to the hub, you have access to a minimum of 2 and easily 12 or more buttons. Please note that having a more complex controller expands the possibilities but, from an inclusive perspective, complicates the task of performing a specific action. In the case of motor impairment, it might be challenging to reach a certain button or avoid accidentally pressing another (Kwan et al., 2011). In the case of cognitive impairment, it might be difficult to memorize or recall an excessive number of actions.

Now, let us analyze the musical functions that can be linked to such actions. First of all, it would theoretically be possible to differentiate the musical parameters controlled by different buttons, or sets of buttons. For example, the four action buttons could play different pitches, while the four shoulder buttons could change the current instrument. However, this approach might be cognitively demanding and its feasibility should be evaluated by a therapist or caregiver to challenge the user without causing a sense of frustration.

Since the range of disabilities is wide and user-tailored approaches are often required, in our proposal the musical parameters linked to each button can be customized via software (see the central part in Figure 3. In this way, a user capable of clicking a single button could, e.g., trigger an impulsive drum sound, like a triangle or a snare drum; a user capable of using two buttons could play two unpitched instruments to reproduce a more complex rhythmic pattern or alternate between two pitches, e.g., the tonic and the dominant in a musical scale; a user capable of triggering eight buttons could play a sort of toy piano or xylophone, as well as produce a range of different sound effects; and so on.

Educational applications range from the awareness of musical parameters, such as melody, rhythm, harmony, and timbre, to the development of skills in playing together and being involved in a collaborative performance.

To the aims of music education, we give key importance to the possibility of translating user actions into customizable musical parameters under the control of the therapists or the users themselves. The production of MIDI or OSC messages operated by the software guarantees generalizability and compatibility with external sound and music systems.

6.2 Triggering Sequences of Events

Considering once again simple actions such as button clicks, it is possible not only to control simple events, such as "play an impulsive sound," "trigger the attack of a note," and "change the current instrument," but also to initiate more complex, pre-defined sequences, in the form of music or sound events.

An example involving musical sequences consists of assigning different melodic-rhythmic patterns to the pressing of different buttons. Educational experiences can be imagined where these sequences must be reproduced in a given order (for example, after the first one ends, the second one must begin, and so on), or giving the user the freedom to explore the sequences in any order, either mutually exclusive or not. The adoption of colored buttons, together with the production of sound, can be a reinforcement technique both to explain and to memorize actions.

Furthermore, in relation to the execution state of the sequences, it is possible to assign different functions to the clicks: a complete *button click* could start the sequence without allowing it to be stopped, or a second *button click* could act as a stop button, or *button down* and *button up* could perform the functions of play and pause, respectively. The sequence could be rewound or paused after a stop action.

Now, let us consider a similar approach in a multitrack context involving an external DAW controlled by the software. Ad hoc buttons could start, stop, and rewind the overall playback. A group of specialized buttons could be associated with a set of tracks, performing a selective muting/unmuting action. The vertical directional arrows could control the volume. The horizontal directional arrows could change the context, for example by loading the previous or the next song within the library.

Also in this scenario, the software's function is to allow customization of all associations between events and actions within the DAW, thus supporting a highly customized educational experience. To mention but a few options, a therapist can arrange an ad hoc song library in the DAW, prepare the number of independent tracks depending on the user's skills, choose the most suitable input device to be connected to the hub, configure the number and position of the buttons to be activated, set the user's actions that trigger sound events, and so on.

7 CONCLUSION

This paper presented a hardware/software ecosystem designed to make music education more accessible and inclusive through the use of push-button interfaces. The proposed system aims to bridge the gap between physical interaction and musical expression by leveraging affordable, widely available hardware and customizable software. By focusing on simplicity and adaptability, the system is particularly wellsuited for young children and individuals with cognitive or physical disabilities, enabling meaningful engagement with music.

The research highlights the versatility of push-button actions, ranging from basic interactions like clicks and holds to more advanced patterns such as sequential presses and pressure-sensitive input. These actions are mapped to diverse musical functions, ensuring that users with varying motor skills and cognitive abilities can actively participate in music-making activities. The ecosystem's flexibility is further demonstrated through its compatibility with standard protocols like MIDI and OSC, facilitating seamless integration with external sound systems and DAWs.

The observational study conducted with the *Inclusive MIDI Controller* implementation underscores the potential of this approach to foster creativity and collaboration. Educational use cases, such as triggering simple events, playing predefined sequences, and controlling complex musical parameters, showcase the system's ability to adapt to individual needs. This adaptability not only lowers the barriers to music education but also empowers users to develop a deeper appreciation and understanding of music.

While the results are promising, further research and development are necessary to expand the system's capabilities. Future work will include integrating advanced sensing technologies, exploring new interaction paradigms, and planning extensive user studies to refine the platform's design. Furthermore, we need to conduct experiments on the field with users characterized by different types of impairments. In this sense, we want to adopt standard assessment tools, such as the *System Usability Scale* (SUS) (Brooke et al., 1996), the Accessible Usability Scale (AUS),[5] and the *Quebec User Evaluation of Satisfaction with Assistive Technology* (QUEST) (Demers et al., 1996).

By addressing both educational and accessibility challenges, this ecosystem represents a significant step toward democratizing music education. Its ability to cater to diverse user needs highlights the transformative potential of inclusive design in fostering engagement, creativity, and collaboration in music-making.

REFERENCES

Allotta, B., Colla, V., Bioli, G., and Conticelli, F. (1999). Haptic interface for simulating push-buttons. In *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No. 99TH8399)*, pages 878–883. IEEE.

Altinsoy, M. E. (2020). Perceptual features of everyday push button sounds and audiotactile interaction. *Acoustical Science and Technology*, 41(1):173–181.

Amico, M. D. and Ludovico, L. A. (2020). Kibo: A midi controller with a tangible user interface for music education. In Lane, H., Uhomoibhi, J., and Zvacek, S., editors, *Proceedings of the 12th International Conference on Computer Supported Education (CSEDU 2020) - Volume 1*, CSEDU, pages 613–619, Setúbal. SCITEPRESS - Science and Technology Publications, Lda.

Baratè, A., Korsten, H., Ludovico, L. A., and Oriolo, E. (2023). Music tangible user interfaces and vulnerable users: State of the art and experimentation. In Constantine, L., Holzinger, A., Silva, H. P., and Vanderdonckt, J., editors, *Computer-Human Interaction Research and Applications. 5th International Conference, CHIRA 2021, Virtual Event, October 28–29, 2021, and 6th International Conference, CHIRA 2022, Valletta, Malta, October 27–28, 2022, Revised Selected Papers*, volume 1882 of *Communications in Computer and Information Science*, pages 1–25. Springer Science and Business Media Deutschland GmbH.

Baratè, A. and Ludovico, L. A. (2024). A multidimensional taxonomy model for music tangible user interfaces. In *MultiMedia Modeling. 30th International Conference, MMM 2024, Amsterdam, The Netherlands, January 29 – February 2, 2024, Proceedings, Part III*, volume 14556 of *Lecture Notes in Computer Science*, pages 517–531. Springer.

Baratè, A., Ludovico, L. A., and Oriolo, E. (2021). An ensemble of tangible user interfaces to foster music awareness and interaction in vulnerable learners. In Constantine, L., Holzinger, A., and Silva, H. P., editors, *Proceedings of the 5th International Conference on Computer-Human Interaction Research and Applications (CHIRA 2021)*, pages 48–57. SCITEPRESS - Science and Technology Publications, Lda.

Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.

Chen, Y., Liang, X., Chen, S., Chen, Y., Lin, H., Zhang, H., Jiang, C., Tian, F., Zhang, Y., Yao, S., et al. (2022). Haptag: a compact actuator for rendering push-button tactility on soft surfaces. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pages 1–11.

Demers, L., Weiss-Lambrou, R., and Ska, B. (1996). Development of the quebec user evaluation of satisfaction with assistive technology (quest). *Assistive technology*, 8(1):3–13.

Faschi, V., Ludovico, L. A., and Avanzini, F. (2024). An accessible software interface for collaborative music performance. In *Proceedings of the 21st Sound and Music Computing Conference*, pages 150–157. SMC.

Frid, E. (2019). Accessible digital musical instruments—a review of musical interfaces in inclusive music practice. *Multimodal Technologies and Interaction*, 3(3):57.

Gaspar, J., Fontul, M., Henriques, E., and Silva, A. (2019). Push button design requirements and relations to but-

[5] https://makeitfable.com/accessible-usability-scale/

ton architecture elements. *International Journal of Industrial Ergonomics*, 70:92–106.

Gentile, V., Sorce, S., Malizia, A., Pirrello, D., and Gentile, A. (2016). Touchless interfaces for public displays: can we deliver interface designers from introducing artificial push button gestures? In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 40–43.

Ishii, H. (2007). Tangible user interfaces. In *The human-computer interaction handbook*, pages 495–514. CRC Press.

Kwan, C., Paquette, I., Magee, J. J., Lee, P. Y., and Betke, M. (2011). Click control: improving mouse interaction for people with motor impairments. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '11, page 231–232, New York, NY, USA. Association for Computing Machinery.

Morimoto, K., Miyajima, C., Itou, K., and Takeda, K. (2007). A virtual button interface using fingertip movements. In *2007 International Conference on Machine Learning and Cybernetics*, volume 4, pages 2089–2093. IEEE.

Plotnick, R. (2018). *Power button: A history of pleasure, panic, and the politics of pushing*. MIT Press.

Rinta, T. (2019). A casestudy on the use of an innovative, technical, musical instrument, skoog, in a special needs education setting with a child with autism and its effects on social skills. *Journal of Music, Technology & Education*, 12(2):179–200.

Samuels, K. and Schroeder, F. (2019). Performance without barriers: Improvising with inclusive and accessible digital musical instruments. *Contemporary Music Review*, 38(5):476–489.

Wright, M. (2005). Open Sound Control: an enabling technology for musical networking. *Organised Sound*, 10(3):193–200.