DOI: 10.61782/fa.2025.0558

MELOPAD, A PORTABLE DEVICE FOR MUSIC COMPOSITION BASED ON THE BELA AUDIO AND INTERACTION PLATFORM

Jasmin Thieme [1*] **Luis Molina-Tanco** [2]

[1] Shure DACH, Eppingen, Germany

[2] Telecommunication Research Institute (TELMA), Universidad de Málaga, Málaga, Spain

ABSTRACT

MeloPad is a prototype of a portable device designed for recording short musical annotations. Portability is key, allowing users to create music in any situation without requiring an external power source. The device includes interchangeable physical covers that simulate different instrument families. *MeloPad* detects the attached cover and dynamically adjusts its menu to offer sounds corresponding to the selected instrument family. It features an OLED display and buttons for selecting options. Users can practice a melody, record it, listen to it through headphones, and store it in the device's memory. When no cover is attached, previously recorded melodies can be accessed. The interaction and audio processing are managed using Bela, an open-source embedded computing platform designed for real-time audio and sensor-based applications. MeloPad is inspired by a MIDI peripheral developed by the French company Joué Play.

Keywords: *portable synth, Bela audio, i2c interface, 3d prototyping*

1. INTRODUCTION

The primary goal of this project, part of a bachelor's degree in Sound and Image Engineering (a four-year program), is to develop *MeloPad*, a fully functional prototype of an interactive electronic device that integrates knowledge in audio signal processing, digital circuit design, programming, and 3D modelling.

MeloPad is designed to capture spontaneous musical ideas anytime, anywhere, without relying on a mobile phone or the limitations of voice recording. Musical inspiration often strikes at inconvenient moments -on a train, while walking, or travelling- when there is no easy way to record a melody. While a phone's notepad or voice recorder can help, humming a tune rarely does justice to how it sounds in one's mind. *MeloPad* provides an intuitive alternative, offering a selection of virtual instruments and adjustable frequency ranges to accurately capture musical ideas. It features interchangeable physical covers called "*sábanas*" (sheets) which mimic different instrument interfaces. The device detects the placed sheet and adjusts its sound accordingly. Users can record, play back, and save their melodies for later use.

MeloPad draws inspiration from the *Joué Play* MIDI controller by the French company *Joué Music Instruments* [1]. Similar to Joué Play, which uses swappable, velocity-sensitive silicone pads to represent different instrument layouts, *MeloPad* employs interchangeable sheets to simulate different instrument interfaces such as a piano, string instrument, or percussion set. This physical interaction enhances the user experience, making the device more intuitive. Unlike Joué Play, which requires external power and must be connected to a tablet or similar device, *MeloPad* is fully autonomous, running on a rechargeable battery. A small OLED screen and three buttons allow for intuitive navigation and selection of options.

MeloPad is built around *Bela* [2], a real-time audio processing platform with low-latency and a user-friendly development environment, designed for musicians and

[*] ***Corresponding author***: Jasmin Thieme, jasminth168@gmail.com

developers with intermediate programming skills. To keep the device compact and portable, we chose *Bela Mini* [3], a smaller version that retains all essential functionality while reducing the overall size.

2. USE CASE ANALYSIS

To define the functionality of *MeloPad*, we have identified three main use cases that ensure a complete user experience: recording melodies, accessing stored recordings, and transferring them to an external device. Additionally, we analyse the operational context in which users interact with the system, detailing the different modes of operation and the interfaces involved. Finally, we outline the key functional requirements that *MeloPad* must meet to ensure seamless recording, playback, and file transfer, as well as an intuitive and portable user experience.

2.1 Use cases

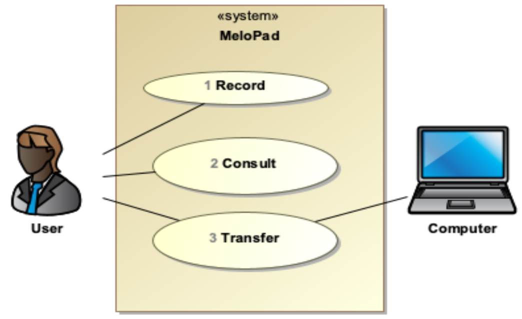

Figure 1. *MeloPad* Use Cases.

The system supports the core use cases shown in Figure 1 and explained as follows:
1. **Record:** The user selects an instrument and an octave range, practices the melody, and starts recording. The system captures the played notes and stores them as a .wav file in *MeloPad*'s memory.
2. **Consult:** Users can access previously stored melodies, play them back through headphones, and delete them if desired.
3. **Transfer:** By connecting *MeloPad* to a computer via USB, users can access the stored .wav files, copy them, and delete them using a file explorer interface.

Each use case has specific preconditions. For recording, a capacitive sheet must be correctly placed, and headphones must be connected. For accessing stored recordings, the sheet must be removed. Transferring files requires a USB connection to a computer.

2.2 Operational Context

MeloPad interacts with the user through various interfaces. The capacitive sheet detects touch input, while buttons allow navigation through menus and function selection. The system provides feedback through an integrated display and audio output via a 3.5mm headphone jack.
Key system interactions include:
• **User inputs**: Touching the capacitive sheet (note input), pressing buttons (navigation and controls), and adjusting the volume potentiometer.
• **System outputs:** Displaying information on the screen, playing audio through headphones, and allowing USB file transfer.
• **External connections:** The system connects to a computer for transferring recordings and supports battery charging via micro-USB.

MeloPad automatically detects whether a capacitive fabric is installed or not. If present, it enters Record Mode, allowing users to generate and record melodies. If no fabric is detected, the system switches to Consult Mode, where users can browse and play stored recordings. When connected to a computer, it enters Transfer Mode, enabling file management via micro-USB.

2.3 Functional and nonfunctional requirements

MeloPad's functionality is defined by the following core requirements:
• R1: **Audio Output** – The system must generate and play sound through headphones.
• R2: **Sound Configuration** – Users should be able to adjust volume (R2.1), select virtual instrument sounds (R2.2), and define a frequency range (R2.3).
• R3: **Portability** – The system must be portable, including a rechargeable battery (R3.1), weighing no more than 1kg (R3.2), and having a maximum dimension of 30 cm (R3.3).
• R4: **Capacitive Sensing** – *MeloPad* must detect user interactions through capacitive sensors.
• R5: **Recording Functionality** – Users should be able to start (R5.1) and stop (R5.2) recordings via button presses.
• R6: **Playback and Management** – The system must support playback (R6.1), deletion of a recent recording (R6.2), browsing stored files (R6.3), and removal of saved recordings (R6.4).
• R7: **Display Interface** – A screen must provide visual feedback and system status information.
• R8: **File Transfer** – *MeloPad* should allow users to transfer recordings to a computer via an integrated file management application.

These requirements have guided the development and verification of the prototype to ensure it meets the needs of the end user.

3. TECHNOLOGY AND DESIGN CHOICES

The *MeloPad*'s functionality is achieved through a combination of hardware components and developed software. The core of the device is the Bela Mini platform, chosen for its low-latency audio processing and compact size. Bela Mini runs Linux and is well-suited for the creation of interactive musical applications. Its development is supported by a user-friendly Integrated Development Environment (IDE) [4] accessible via the web, offering tools for coding, file management, and real-time audio visualization through a virtual oscilloscope.

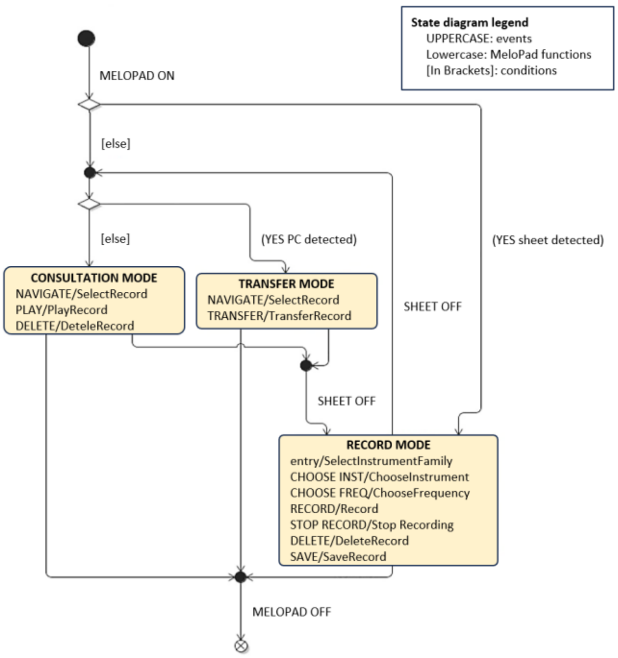

Figure 2. *Melopad*'s state diagram.

The whole hardware and software design was started after completing the first phase of this project, where the use cases and the system states were specified. Based on Figure 2 the next step was to start with the hardware and software design following the chosen requirements.

3.1 Hardware design

For user interaction and sound generation, *MeloPad* utilizes several key hardware elements:

• Trill Craft Sensor: The device employs a Trill Craft capacitive sensor [5]. This sensor, developed by Bela, features thirty inputs, allowing for the connection of conductive materials to detect variations in capacitance caused by touch. In *MeloPad*, this sensor is crucial for interpreting the user's input on the interchangeable "sábanas".

• Interchangeable "sábanas" (Sheets): A key innovative aspect of *MeloPad* is the use of physically interchangeable covers called "sábanas". These sheets are designed to simulate the interfaces of different musical instruments, such as a piano. *MeloPad* can detect the placed sheet through a detector circuit, and the software then offers corresponding instrument sounds in a menu. For example, a "sábanas" simulating an electronic piano was created to validate the concept. The connection of the "sábanas" is detected by the Bela, which then determines the appropriate instrument.

• OLED Screen and Buttons: A small OLED screen is integrated into the device to display basic options related to the placed "sábanas" and general information. Navigation through menus and selection of functions are facilitated by three buttons.

• Potentiometer: A potentiometer is included to allow the user to adjust the volume level of the audio output.

• Audio Output: Audio generated by *MeloPad* is outputted through a 3.5mm mini-jack stereo connector for headphones.

The communication between the Bela Mini and peripherals like the OLED screen and the Trill Craft sensor relies on the I2C (Inter-Integrated Circuit) serial communication protocol. I2C uses only two signal lines (SDA and SCL) for data transmission and synchronization. Each I2C device on the bus has a unique address, allowing the Bela (as the master device) to communicate with each peripheral individually. In *MeloPad*, the OLED screen has an address of 0x3c, and the Trill Craft sensor's address is 0x30.

For the physical design of *MeloPad*, the 3D modelling programme Blender was used to find the appropriate design and create the 3D printable model.

3.2 Software design

On the software side, the project is primarily developed using C++ within the Bela IDE. The software structure follows the standard Bela program flow with setup(), render(), and cleanup() functions. Several Bela-specific libraries are utilized, including:

• Bela.h: Provides access to core audio settings and input/output buffers.

• Trill.h: Enables communication with and control of the Trill Craft sensor.
• AudioFile.h: Facilitates the reading and writing of audio files to the device's memory, using the AudioFileUtilities namespace with functions like load() and write(). The setSamples() function is used for saving the generated melody, and loadMono() is used for reading from memory.

The project incorporates the Synthesis ToolKit (STK) [6], an external C++ library for real-time audio synthesis and processing. STK provides a wide range of virtual instruments, which are utilized in *MeloPad* to generate the different instrument sounds selectable by the user. Integrating STK required installing the library via the Bela's Ethernet connection (on a regular Bela board, as Bela Mini lacks Ethernet) and learning its API. The general structure for using STK involves creating instrument objects, setting their sample rate (Stk::setSampleRate(BelaSampleRate)) and frequency (instrument->setFrequency(frequency)), and using the sound generating functions: noteOn(frequency, amplitude) which generates a note but does no send it to the output; tick() to enable real-time audio production by advancing a sample on each call; and noteOff(amplitude) to turn off the generated note adjusting the fade out.

The software architecture also includes custom-developed classes like Display and Teclado (Keyboard) to organize the code for screen updates and sound generation, respectively. A Debouncer class, adapted from an open-source Bela resource, is used to handle button presses and prevent misreading due to bouncing. The U8g2 library is employed for controlling the OLED display via I2C. The entire software is configured to run automatically upon Bela Mini startup. It is possible to find the whole software on a Github repository[1].

This combination of Bela Mini's real-time audio capabilities, the Trill Craft sensor, interchangeable "sábanas" for varied user interfaces, and the STK library for sound synthesis forms the technological foundation of the *MeloPad* prototype.

4. DEVELOPMENT CHOICES AND PROTOTYPE

This project has involved a major development effort as it is a complete prototype of a software-hardware product. It was inspired by the Joué device [1], the first of the class of synthesisers whose keyboard is not fixed but interchangeable.

[1]Github Repository of *MeloPad*'s Software: https://github.com/JasminThieme/MeloPad

4.1 Mechanical design decisions

To minimize the size of *MeloPad*, the Bela Mini was chosen, which is more compact compared to the standard Bela version (56 x 35 x 21 mm vs. 87 x 54 x 27 mm). The Bela Mini is suitable for this project as no analogue outputs (such as those for speakers) are needed, which are exclusive to the standard Bela.

Initially, a touchscreen was considered for menu navigation, but it was quickly replaced with buttons due to the complexity and cost involved. Nevertheless, a screen was kept to display information and show the options activated by the buttons. After researching various screen sizes, the first version of *MeloPad* was created, which initially featured a small screen around one inch in size. This option would have required many push buttons, which would have made the design bulky. As a result, a design without a screen was proposed, using an LED for each button. However, this idea was soon discarded as one of the goals of the project was to implement a visual interface.
After extensive searching, a suitable 2.42-inch screen was found, leading to the second design version: five buttons with individual functions. Finally, the number of buttons was reduced to three and their functionality would change depending on the options displayed on the screen. This resulted in a complete implementation of both the screen and buttons.

Figure 3. *MeloPad* final 3D-design.

The choice of buttons and the potentiometer was based on a compromise between three factors: usability, space, and design, as the goal was to minimize the size while achieving an elegant, simple and intuitive prototype. After acquiring the chosen screen and buttons, the final design was adjusted as shown in Figure 3. The 3D model created in Blender has been shared on a Github repository[2] as well to complete the design and for future consultation.

[2]Github Repository of *MeloPad*'s 3D Model: https://github.com/JasminThieme/MeloPad-3D

4.2 Tests and implementation of "sábanas" (sheets)

The Trill Craft capacitive sensor from Bela was chosen due to its ability to create and customize sensors. With 30 capacitive inputs, it allows for the development of a wide range of projects in various sizes and applications. To ensure the proper operation of *MeloPad*, tests were carried out with different conductive materials, and the best option was to use a wire without the insulating plastic, as it has a bigger conductive surface and would be more sensitive to touch.

Two types of sheets were planned to be created: one to mimic a string instrument and another for a keyboard. The keyboard design was the simpler of the two, requiring one position per note. However, the string instrument design was more complicated due to the number of notes involved. Initially, multiplexing the strings to generate tones based on the combination of cables touched was considered, but it was discarded due to the limited position detection capability of the Trill Craft sensor. Eventually, one idea was to assign each guitar tone to an input on the sensor, this allowed for five frets to be covered with a single sensor, and by combining two sensors, a complete guitar neck could be built. The final test for the guitar sheet was based on three frets and three strings, with the sensitivity adjusted to ensure proper response. Due to time constraints, the guitar sheet implementation was discarded, and only the piano sheet was completed.

For the piano sheet, a simple design was used, with one cable per key to detect the capacitance variations. The software implementation utilized the equal temperament chromatic scale, where each semitone is calculated based on the twelfth root of two to calculate the frequencies of the two octaves of notes as in Eqn. (1):

$$f_n = f_0 * (\sqrt[12]{2})^n \quad (1)$$

where n is the note number relative to the fundamental, f_0.

4.3 Power supply for *MeloPad*

One challenge in designing *MeloPad* was that the same USB-micro connector of Bela Mini would be used for both power and connection to external devices. This means that the rechargeable battery and the USB-micro connector for connecting to a computer needed to share the same USB-micro port on the Bela.

The solution was to implement a sliding switch. In one position, the switch connects the Vcc input of the Bela to the Vcc of the rechargeable battery. In the second position, it connects the Vcc input from the external USB-micro connector, to connect an external device. For data transmission, the cable connected to the Bela needs to have four wires: GND, Vcc, Data+, and Data-, therefore the cable connected to Bela is a 4-wired one. For powering the device, only the first two wires are used as shown in Figure 4.

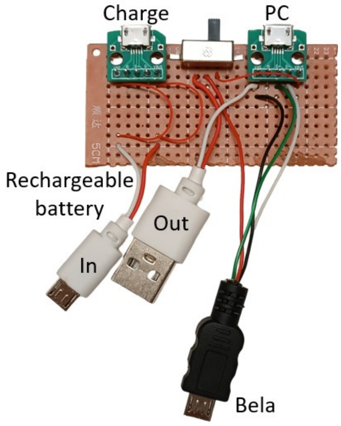

Figure 4. Power and connections piece.

The solution was to implement a power and connection piece which includes two USB-micro female connectors exposed to the outside, one micro-USB male connector with 2 wires for powering the external battery, one with 4 wires to connect the Bela, and a USB-A connector for charging the external battery. This set of components is housed inside the case.

4.4 Implementation of the display

The *MeloPad* features a 2.42-inch OLED display with a resolution of 128x64 pixels, offering a high contrast with white text on a black background. OLED technology provides individual pixel illumination, resulting in clearer visuals compared to LCD screens. The display serves as the main interface for user interaction, where information is displayed, and options are selected.

The screen shows various states depending on the action being performed. For the use case Record these software states include:

- **INITIAL**: Displayed when the device is powered on.

- **CHOOSE_INST**: The user selects an instrument (e.g., Rhodes, Hammond, Wurley) for the piano sheet.
- **CHOOSE_OCT**: The user chooses the octave for the selected instrument, adjusting the frequency range.
- **INFO**: Displays the current settings. Audio output is generated.
- **RECORD**: Activated when the user starts recording their melody.
- **STOP_REC**: Shown after recording ends, with options to save or retry.
- **SAVE_MEL**: Displays a confirmation message when the melody is saved.

In the Consultation use case (when no sheet is detected), the display shows a file list where users can play, navigate, or delete recordings. Screens for this process include:

- **LISTA**: The file list. One file per screen. Options to choose *next*, *play*, *previous*.
- **PLAY**: Displays the playback state. Option to delete the record.
- **ELIMINADO**: Confirms deletion of a recording.

This OLED display is crucial for guiding the user through different interaction stages, ensuring a seamless user experience while maximizing visual clarity.

4.5 Final prototype design

The distribution of the components inside the *MeloPad* case has been challenging as the components almost entirely occupy the available volume of the case, resulting in a very compact space, see **Figure 5**.

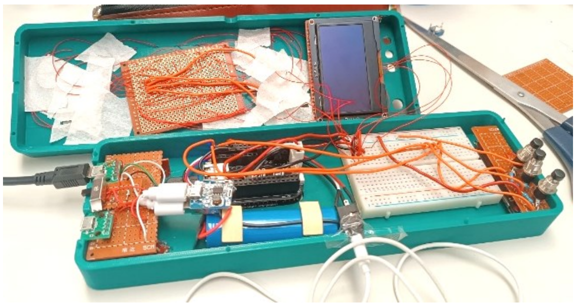

Figure 5. Components distribution in *MeloPad*'s case.

In the upper half of the case, the screen, capacitive sensor, and the cables connected to the sensor are placed. These wires are exposed to the outside through small holes visible in the 3D model, Figure 3. The cables for the sensor needed to be organized in such a way to avoid contact between them and minimize potential interference. The use of rigid cables made the ideal arrangement challenging. However, by maintaining the insulating plastic, interference was kept to a minimum, even when the wires were in contact.

Figure 6. Final look of the *MeloPad* prototype.

The final prototype looked compact and clean as shown in Figure 6. The piano sheet fit on the sensors area and the buttons would let one interact with the menu shown on the screen. Through the headphones the notes would sound when a position is being touched on the piano.

5. RESULTS AND DISCUSSION

5.1 Functionality and performance of the prototype

The *MeloPad* prototype demonstrated satisfactory functionality in its Record and Consultation modes, allowing the generation, recording and playback of sounds through an intuitive screen-based interface and buttons. The portability of the device, thanks to its compact design and rechargeable battery, meets the stated requirements. However, limitations were identified in the file deletion function and data transfer, which led to omitting the implementation of the third use case, Transfer.

5.1.1 Validation of implemented use cases

Verification tests showed that the *MeloPad* prototype largely meets the functional requirements set. In Record mode, the user interface based on the OLED display and buttons allowed an intuitive selection of instruments and octave ranges. Interaction with the capacitive sensors via the piano "sheet" generated accurate sounds and the recording function worked properly, allowing melodies of up to 20 seconds to be captured, played back and stored in Bela's memory.

In Consultation mode, navigation through the list of recordings was seamless, with options for playing and deleting files. However, a limitation was identified in the file deletion function, as files persisted in memory despite confirmation of deletion. This aspect requires improvement of the software.

5.1.2 Performance and Portability

The prototype demonstrated excellent transportability. Its compact (24.5 cm x 8 cm x 3.5 cm) and lightweight (350 grams) design makes it easy to carry and use in different environments. The internal rechargeable battery provides autonomy, eliminating the need for an external power supply.

5.2 Areas of improvement

The *MeloPad* project succeeded in developing a portable, self-standing device with an innovative design, fulfilling the main idea of creating a modular synthesiser with interchangeable "sheets" to simulate different instrument families. The intuitive interface and the ability to record and playback melodies enhance the user experience.

However, areas for improvement and future work were identified:

- **String Sheet**: Study of the layout of the sensors and optimisation of the design to accommodate as many inputs as necessary.
- **Sheet Material**: Explore more durable and tactile materials for the sheets.
- **Hardware Size**: Explore smaller microcontrollers to reduce the volume of the prototype.
- **Internal Cables**: Replace the internal rigid cables with flexible cables to improve durability and avoid disconnections.
- **Safe Shutdown**: Implement a safe shutdown button to protect Bela's operating system, avoiding data corruption.
- **Record Deletion**: Implement a record deletion function that allows users to delete files.
- **Transfer**: Develop a desktop application that allows users to transfer, consult and delete files from *MeloPad*.

5.3 Conclusion

The *MeloPad* project represents a progress in the creation of portable and modular digital musical instruments. Proposed improvements in future work could make *MeloPad* an even more versatile and accessible instrument for musicians and sound experimentation enthusiasts.

6. REFERENCES

[1] A. Douglas, "Joué Play MIDI controller review: Free your fingers and format", Popular Science, Sep. 30, 2022. [Online]. Available: https://www.popsci.com/gear/joue-play-midi-controller-review/. Accessed: Mar. 2025.

[2] Bela.io, "Craft beautiful interaction". Available: https://bela.io/. Accessed: Mar. 2025.

[3] Bela.io. "Bela Mini". Available: https://learn.bela.io/products/bela-boards/bela-mini/. Accessed: Mar. 2025.

[4] Bela.io. "Get to know the IDE". Available: https://learn.bela.io/get-started-guide/get-to-know-the-ide/. Accessed: Mar. 2025.

[5] Bela.io. "Working with Trill Craft". Available: https://learn.bela.io/tutorials/trill-sensors/working-with-trill-craft/#connecting-to-trill-craft. Accessed: Mar. 2025.

[6] Perry R. Cook and Gary P. Scavone, "The Synthesis ToolKit in C++ (STK)", 1995-2023. Available: https://ccrma.stanford.edu/software/stk/. Accessed: Apr. 2025.