INSTITUTE FOR FUTURE TECHNOLOGIES
DE VINCI HIGHER EDUCATION

Master thesis

How can sensing techniques redefine our interaction with plants ?

Matthieu SEGUI

October 12, 2024

Supervisor
Clément Duhart and Marc Teyssier

Contents

Abstract

This master's thesis explores the transformation of plants into bio-sensors, using their natural sensing capabilities to capture and interpret human-plant interactions. The research introduces a system where data from plants is used to generate real-time auditory feedback, creating a novel sound-based interaction mechanism. Additionally, the project extends this concept by developing a network of interconnected plant-based sensors, enabling distributed interaction and monitoring. The results demonstrate the feasibility of using plants as interactive bio-sensors, showing potential applications in human-computer interaction and bio-sensing. This work opens new possibilities for sustainable, plant-integrated technologies in both environmental and artistic contexts.

1 Introduction

1.1 Background motivation

I am a technology engineer specializing in embedded systems and their hardware/software architecture. At the request of my principal investigator, I developed a device that uses sensor data from plants to generate sound based on human-plant interactions. Despite having no prior experience with musical systems, this project allowed me to explore how data from plant sensors could be transformed into sound, providing a way to engage with plants through auditory feedback.

Building on this work, I am also focused on creating a network of plant-based sensors. By using the natural sensing capabilities of plants, I aim to extend the functionality of individual plants into a larger system that can capture and record interactions, forming a connected network of bio-sensors.

1.2 Context and overview

The development of new interaction methods and sensor technologies is rapidly evolving, driven by the increasing integration of advanced sensing devices into everyday life. This thesis explores an unconventional approach to sensor systems by exploiting the natural sensing capabilities of plants. Plants are naturally provided with thousands of sensors that respond to environmental stimuli, making them a resource for monitoring and interaction technologies.

The primary goal of this research is to understand and utilize the sensing abilities of plants, transforming them into bio-sensors. By doing so, the project aims to create networks of interconnected plants that can monitor their own environmental states and respond to human interaction. This approach could provide a sustainable alternative to traditional silicon-based sensor networks, potentially reducing the reliance on artificial materials and the environmental impact associated with them.

Additionally, this research has implications for the field of Human-Computer Interaction (HCI), offering new opportunities to explore plant-based interfaces. By incorporating plants into interaction systems, this thesis could introduce a natural interface that puts the spotlight on a more integrated relationship between humans and their surrounding environments. This exploration of bio-sensors as part of HCI opens up new paths for sustainable, living technologies that go beyond conventional electronic systems.

1.3 Problematic

The main problematic that this master thesis will focus on is :

How can sensing technologies redefine our interactions with plants ?

1.4 Research domain

The study of human-plant interaction covers a broad spectrum of research areas. This master's thesis focuses on the Human-Computer Interaction

(HCI) field, which examines the interfaces between people and computers. HCI sits "at the intersection between psychology and social sciences, on the one hand, and computer science and technology, on the other," [1] providing insights into how humans interact with technology.

In this thesis, the concept of interaction is extended to plants and nature, aiming to enhance plant capabilities and explore how plants can be used as interactive elements. By transforming plants into living sensors, the research also intersects with the fields of instrumentation engineering and electronics. These fields focus on developing new ways of capturing data to design functional sensors. Understanding plants as bio-living sensors requires the application of specific sensing techniques to access their natural abilities.

Additionally, this work involves the sensor development field, where plants are converted into functional sensors. This transformation represents an innovative approach to expanding the possibilities of plant-based sensing technologies.

1.5 Contributions

This thesis makes several contributions to the field of Human-Computer Interaction and sensor technologies:

1. **Development of Plant-based Bio-sensors and Sound Interaction System**: This thesis introduces a novel system that transforms plants into living bio-sensors, utilizing their natural sensing capabilities to capture human-plant interactions. The data collected from plant sensors is used to generate real-time auditory feedback, creating an innovative, sound-based interaction mechanism. This approach opens up new avenues for exploring plant interactions in both technological and artistic contexts.
2. **Creation of a Network of Plant Sensors for Distributed Interaction**: In addition to the single-plant sensor system, this research extends the concept by developing a network of interconnected plant-based sensors. This network allows for distributed sensing and interaction across multiple plants, enabling more complex data collection and processing. The system demonstrates potential applications in human-computer interaction, environmental monitoring, and artistic installations, highlighting its multidisciplinary impact.

2 State of the art

2.1 Plant as sensor

2.1.1 Human-Plant cohabitation

Plants have a lot of benefic effects on human. The study from Charles Hall and Melinda Knuth [2] explain all the benefits of plants on our human system. Watts and al. shows that urban "greening" (add green spaces in urban city) increase tranquility, relieve stress and anxiety [3]. An experience has been conducted in offices by Ikei and al [4] to expose roses to employees. The experience showed that the "parasympathetic nervous activity was significantly higher while viewing the rose". The subjects were more comfortable being expose to roses than people that were not.

On top of that, we, as human, are spending 85% of our live indoor [5]. We are subject to the *technostress*. *Technostress* is a term introduced by Brod and Craig [6] in 1984. Brod and Craig describe it as a modern disease caused by the inability of a person to interact and use information and communication technologies in a healthy way [7]. Lee and al [5] compared the stress caused by the doing a task on a computer to interacting with a plant (specifically transplanting it). The computer task was increasing the level of stress (increased diastolic blood pressure and sympathetic nervous system activity for example). The plant interaction on the other side was found to create positive feeling to the subjects. The plant is a good way of reducing the *technostress*.

Another study comes to the same conclusions. Hassan and al. [8] studied the effect of plant interaction on young adults subjects to stress induced by electronic devices (similar to *technostress*). They concluded that significant differences in blood pressure occurred between the subjects interacting with a plant (transplanting) and the one that add to do the task on the electronic device (phone).

It's understandable that people see plants as un-stressful and not harmful. This makes them a perfect human-machine interface.

On top of that, Kellert and al. studied the benefits of adding plants and vegetation into architecture. They defined the *Biophilic Design* [9]. *Biophilic Design* is an architectural and design approach that integrates natural elements into built environments, increasing human connection with nature [10]. The *Biophilic Design* is based on the *Biophilia Hypothesis* [11], which posits that humans have an inborn tendency to look for connections with nature and other forms of life. *Biophilic Design* ideas spread a mean to promote well-being, productivity, and sustainability in urban settings. This design philosophy incorporates natural light, ventilation, vegetation, water features, and organic materials to mimic natural environments, putting forward psychological and physiological benefits. The *Biophilic Design* has, of course, been specifically studied for offices architectures.

Research shows that *Biophilic Design* improves cognitive function, reduces stress, and enhances creativity and productivity in workplace and educational places [12]. Moreover, *Biophilic Design* extends beyond design it aims to increase environmental sustainability by using nature-inspired materials and technologies that reduce energy consumption and enhance

Figure 1: *Biophilia Hypothesis* architecture example. This building is the perfect example of Thermal and Airflow Variability from the *Biophilia Hypothesis* principles.

indoor air quality. The integration of living systems, such as vertical gardens and green roofs, not only improve biodiversity in urban areas but also helps regulate temperature and manage water coming from rain [13] (ref figure 1).

Recent innovations have expanded the application of *Biophilic Design* in smart buildings, where sensors and digital interfaces track environmental conditions, creating dynamic environments that respond to human and ecological needs [14]. This approach aligns with the growing trend of human-centered design in architecture, where spaces are designed to maintain physical and emotional well-being. *Biophilic Design* is a critical component of the future of sustainable and health-focused architecture.

This framework aligns well with the concepts of the Internet of Plants, as both reinforcing human interactions with nature through technology.

2.1.2 Biosensors

The term *Biosensor* refers to an analytical device (a sensor) that involves a biological sensing element [15]. *Biosensor* are mainly used in the healthcare, agriculture, food safety and to do environmental monitoring. Indeed, they are mainly made to measure chemical substances or biological reactions [16]. At their core, *Biosensors* typically consist of a bio-recognition element (enzymes, antibodies, nucleic acids, or even whole cells) and a transducer that converts biological responses into measurable signals, such as electrical, thermal, or optical signals.

Advances in bio-sensor technology have revolutionized personalized medicine and point-of-care diagnostics, with devices like glucose sensors [17], pregnancy tests, and wearable health monitors widely used today. These innovations offer real-time data and enable continuous monitoring of physiological conditions, improving disease management and patient outcomes. Beyond healthcare, *Biosensors* are integral to environmental monitoring, particularly in detecting pollutants and hazardous substances, using microorganisms or plant tissues to assess soil and water quality.

The research is now progressing towards plant-based *Biosensors*, which increase plants' natural capabilities to sense environmental changes such as detecting polluted soils [18]. These *Biosensors* represent a sustainable alternative to traditional silicon-based devices and align with the Internet of Plants.

2.1.3 Human-Plant interaction

The human plant interaction has been studied. Seow and al. [19] created a framework that is able to detect when something (and someone) interact with a plant. However, this is not any plant, the plant used is the *Mimosa Pudica*. This plant is special, when something touches its leaves, the plant closes its leaves to protect them from the danger [20]. An electrical impulse is released and is catch by the device Seow and al. developed. The electrical signal is easy to catch and thus can be used as actuator. However, the plant needs time and energy to re-open the leaves. This framework also can't be generalized to other species of plants.

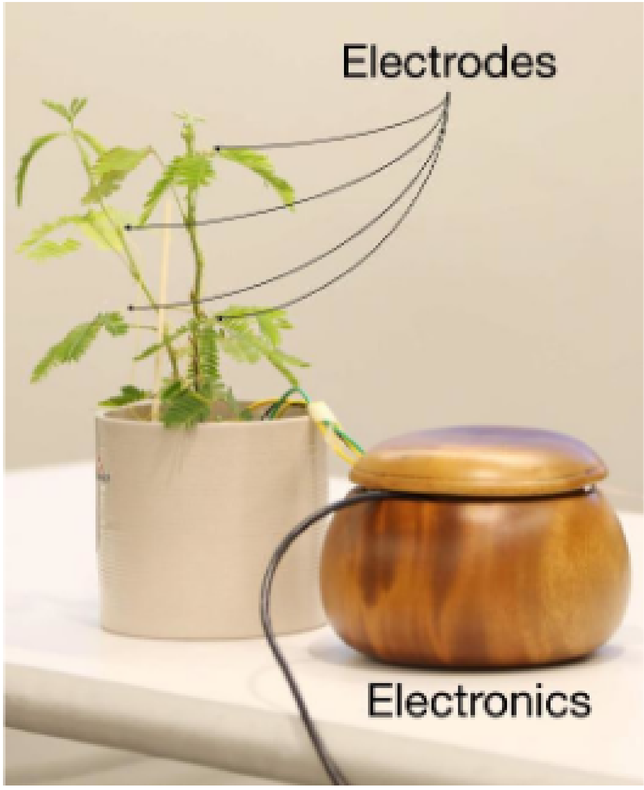

Figure 2: *Pudica framework* made by Seow and al. The *Mimosa Pudica* is a special plant that react to the interaction by closing its leaves. The framework captures the electrical impulse and then interprets that an interaction happened.

Sato and al. used the process of capacitive sensing to detect interaction with objects of our daily lives [21]. In this paper, they proposed a device called *Touché* that use swept frequency capacitive sensing to detect touch interaction (ref figure 3) but also more complex interactions (such as interacting with a finger, the whole hand...). More complex interactions are captured using machine learning algorithm.

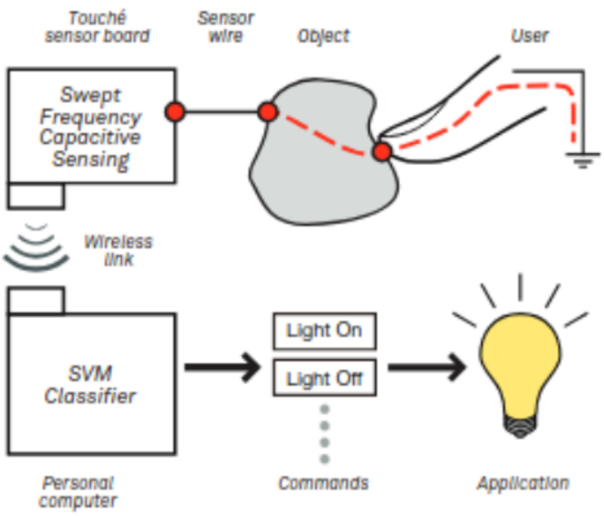

Figure 3: *Touché* made by Sato and al. The *Touché* architecture that is based swept frequency capacitive sensing. The user is closing the circuit by representing the ground level.

This paper doesn't apply the device to plants. Poupyrev and al. [22] used the device on plant to demonstrate the usage. This swept frequency technique is usable and better than the previous single frequency technique as it captures more data. In their article, Honigman and al. [23] adapted the *Touché* device to be use with an Arduino* microcontroller. This allowing people to reproduce the set-up easily.

2.1.4 Touch sensors

Usual sensors uses physical properties to capture the data. For instance, in 1999, Hinckley and al. [24] were building a touch sensor made of conductive paint. The conductive paint is used as an electrode. In the circuit, a component is generating a 30 Hertz square signal. When the user interact with the conductive surface, it shifts and induce delay in the square wave. The delay induce by the user hands is caused by its natural capacitance. This circuit, however, gives a boolean output based on a threshold. The answer is *touched* or *untouched*.

When thinking of touch sensors, we think about the trackpad/touchpad that we daily use in our personal computer. Those sensors use resistive or capacitive sensing. Both of these techniques are based on electrical properties. Resistive sensing is based on the perturbation of the resistance in a circuit. Whereas capacitive is based on the capacitance. Capturing those specific properties are usually the basic of touch sensors.

For instance, Olberding and al. created a cuttable multitouch sensor based on capacitive sensor [25]. This specific sensor allows to create something similar to a trackpad but with different shapes.

Reading the capacitance at a fixed frequency is working and is already used in our daily lives. However, to capture more complex interaction

* Open source compute unit

and to rely on the data, adding another dimension of data is useful. Swept Frequency Capacitive sensing consists on emitting a electrical signal and then read the capacitive values similar to what a classic sensor would do. However, the electrical signal is not generated at a fix frequency but is generated following a changing frequency. Sato and al. introduced first this technique in the touché device [21]. This allows to get richer information on the output of the device. *Touché* allows then to capture the information on a variety and a multitude of daily objects. Indeed, searching through many frequencies make it simpler to find small changes and though different kind and type of interactions.

2.1.5 Humidity sensors

In order to capture more data than only the touch interaction, we are interested into moisture sensors.

Lee and Lee [26] wrote a review on the different kind of humidity sensors. It exists many kind of humidity sensors. They expose many kinds of moisture sensors: resistive, capacitive, gravimetric, optical, piezoresistive and magnetoelastic sensors.

Resistive and capacitive moisture sensors differ mainly in their operating principles and durability. Resistive sensors measure soil moisture by detecting changes in electrical resistance between two probes inserted into the soil, with lower resistance indicating higher moisture levels. While these sensors are cost-effective and simple, they tend to degrade over time due to electrolysis, especially in saline conditions, reducing their reliability. Additionally, resistive sensors are sensitive to temperature changes, which can lead to signal drift, although they offer good sensitivity and typically exhibit a linear response to changes in moisture. In contrast, capacitive sensors work by measuring changes in the dielectric constant of the soil, without direct contact between the electrodes and the soil. This design makes capacitive sensors more durable and resistant to corrosion, offering greater long-term stability and accuracy. They are also less affected by temperature variations. However, capacitive sensors require careful design, with the ratio between the thickness of the sensitive material and the sensor geometry being crucial, as the electrodes' periodicity and the thickness of the sensitive coating layer significantly impact the sensor's response [27]. Despite these complexities, capacitive sensors tend to provide more reliable and stable readings over time, though at a higher cost.

2.1.6 Sonification

Sonification is "the use of non-speech audio to convey information or perceptualize data" [28].

The sonification has been used for a long time using the human ear sensor to interpret data. For instance, the Geiger counter is using the sonification to measure the ionizing radiation. This sensor is "beeping" and "crackling" depending of the level of radiation.

Herman and al [28] created two sonification models to represent particles moving. They are saying that the sonification is allowing a multidimensional representation of the data captured. Humans are sensible to sounds. The sound allows a "rapid screening" because it is easier to listen to a sound than read a graph or a text. The sound when combined with visualization is bringing more granularity and understanding. They are also adding that we are using sound to diagnosis issues on a daily basis ; giving the example of a car mechanic that is having a failure and that we can predict.

The sonification has been used in many projects. Ballora and al. [29] designed a project that mix the fluctuation of the stock market with the apparition of keyword on Twitter social network. They were mapping specific and defined keyword with specific sounds. The designed an app to be able to tweak and change the volume of the different sounds. The same procedure is applied to the variation of the stock price. They conducted this experiment in order to add another possible channel of information in the market place. The traders and people working there are already flooded with visual information. This new channel could allow to process more data. The result were encouraging. However, this experiment rose issues. People were annoyed and disturbed by the sound. They were lowering the volume and thus not hearing it. Some other people were just too much concentrated on their task that they did not hear the variations. More experiment should be conducted to refine the sound created and the level needed.

In the same direction, Ballora and al. [29] also developed a device to monitor the heart rate of people using sonification. It takes the output of an electrocardiogram and transform it into a sound. Each of the ECG's frequencies were pass into a filter and then given a specific sound. The output of the sound creation is then listened to and interpreted. This application with a trained ear can help diagnosis the sleep apnea through heart variability.

2.1.7 Sonification on microcontrollers

MCUs[†] [30] is a kind of small computer. Those devices can be used to generate sound. The most common way of doing electronical music is to use MIDI[‡] [31]. MIDI has been created in order to create music with digital computer. MIDI do not describe directly the audio signal but the human actions to create the signal (such as turn the knob left, push the slider...). MCU are able to produce those kind of directives [32][33]. However, the MCU can produce MIDI but MIDI does not directly generate sounds. A synthetizer is needed to create the sound described.

For our use case of embedding the device, we look at MCU that were able to directly generate the signal from a DAC[§]. Projects had been conducted with many microcontrollers such as a small 8 bits AVR microcontrollers (ATmega32) [34]. This paper does not include limitation of such a product but we can guess that the 8 bits microcontroller is limiting the sound quality. A larger project from Shaer and al. [35] is including an Arduino Mega controlling the visual effect of the project, but also the interaction sensors. The Arduino Mega is then sending MIDI information to Teensy 3.2. The Teensy is then generating the sound. The project is still too large to be fully embedded but the Teensy 3.2 is a promising compute unit.

[†] microcontrollers
[‡] Musical Instrument Digital Interface
[§] Digital to Analog Converter

The Teensy 3.2 is running at 72 MHz, way faster than the ATmega32 that is operating at 16MHz. The frequency is essential when trying the produce sound signals.

2.1.8 Commercial products

Generating music from plants is not a new concept. Several commercial products are available on the market. Looking at *PlantWave* device from the eponym company, the device is able to generate sound from the plant. The device is built using a small box with two electrodes. The electrodes are then connected to the plant. The device generates sound. The process of sonification is not described and thus is blurry as it is a patent technology. The device is also not open-source

Figure 4: The PlantWave product on a demonstration picture (source from their website)

Another commercial product is the *Music of the Plants* product. How the product works is even more opaque and the company building the product is sometimes associated with cult and esotericism. The product is also not open-source.

2.2 Internet of Plants

The Internet of Plants in not a new word. Nevertheless, the term is not that spread around. Aliev and al. [36] evoked the word. The paper explain that the word is based on the IoT[¶]. The paper is using this word to describe a system of sensors to monitor plants and crops. This is really close to the IoT as it is using silicon made sensor to retrieve the data from the plants.

Internet of Plants is especially used for agriculture. In the paper of Steeneken and al. [37], they are explaining that the use of specific sensors could and should be very efficient to boost the productivity of the crops. The paper is also using talking about classic connected sensors but applied to plants.

[¶] Internet of Things

Like the previous authors, Kitano and al. [38] also used this specific word for sensor connected agriculture. I think that the IoP of plants in that use is more a specific application of the IoT instead of a specific field of research.

2.2.1 Distributed system

The Internet of Plants relies on the paradigm of *distributed systems*. Van and al. define distributed systems as " collection of autonomous computing elements that appears to its users as a single coherent system." [39]. Distributed systems scaled up with the improvement of computation power and networking link speed. It is now possible using networking to spread calculation power around a room or even the world and to rely on this computation power as a part of a working system.

Even if distributed systems spread up thanks to the increasing networking speed, they are far from being new. In 1985 Kleinrock and Leonard [40] already explained the principle of distributed systems. They get inspired by nature looking at bees that are communicating to fetch food, ants are also communicating but adding also their strength to get back the food. At this time, adding computation power was a really interesting way of expanding the power on a local network.

Distributed system complicates the monitoring of the system and add complexities and faulty points in the system. There have been a lot of research about the best way of linking pieces of a distributed system. Distributed system are usually based on a communication protocol such as the internet protocol (IP), bluetooth mesh or even Lora. Relying on those already existing communication protocol ease the deployment of the application. However, congested communication protocol or distance can introduce latencies and thus data failure and inconsistency. It is important in a final product that the data is checked and verified. We are them introducing the CAP (consistency, availability and partition tolerance) theorem also called Brewer theorem. It has been introduced by Brewer in 2000 [41]. This theorem explains that it is impossible to have full data consistency, complete availability and partition tolerance in a distributed system. Only two of the three capabilities can be achieved. You always have to think about the balance and what is the most important in your system. It was mainly thought especially for the distributed databases but can also be used for all the distributed systems that are including a network communication.

On the monitoring side, Joyce and al. [42] explained how to build a monitoring on a distributed system. Monitoring in the case of a full solution is mandatory to be able to capture and react to eventual failure on the system.

2.2.2 Distributed instruments

Now that we looked at what is a distributed system, we can go further in what is interesting to us and talk about distributed instrument. Tanaka wrote a chapter in the Oxford Hanbook of computer Music [43] describing the arrival of new instruments based on sensors. The author, in the chapter, first describe the evolution of the definition of an instrument [44]. Then the author is reviewing the new possibilities on the new instruments going through bio-signal instruments, tabletop and surface instruments. The chapter is ending by talking about ensembles in music and then to network music. Network music is exposed as a new generation of ensembles where you can spread the ensemble in space, playing instruments in remote. Pushing even further, this chapter describes network instruments to not only being a replacement of on-site music ensembles but also a new distinct mode of play. The chapter is citing the *Public Sound Objects* project from Barbosa and al. [45]. Barbosa and al. created a space that allows people to create music collaboratively on a wider area: The Internet.

Distributed instruments are then a new way of thinking musical exhibitions and experiments. It brings a new level of possibilities.

2.2.3 Sonification using software

Sonification, the process of converting data into sound for interpretation, has seen significant advancements through the use of software frameworks. On this specific part of the state of the art we will focus only on softwares and applications used for sonification. Modern tools enable real-time, interactive sound generation, which is particularly useful in analyzing complex datasets across disciplines such as neuroscience, environmental monitoring, and human-computer interaction. Kramer and al. [46] explored and wrote a study on the status of the sonification field.

It exists three main sonification platforms : Max/MSP, Pure Data, and SuperCollider. They all offer flexible environments for designing custom sound mappings, supporting a wide range of applications [47][48].

These tools often integrate algorithmic processes, sensor data, and even machine learning, producing sophisticated auditory representations that are increasingly immersive. Both SuperCollider and Pure Data are open source projects. Also Max/MSP is only available on Windows and MacOS based machines which can be a limitation when working with sonification Linux server.

Max/MSP and Pure Data work similarly (ref figure 5 and figure 6)

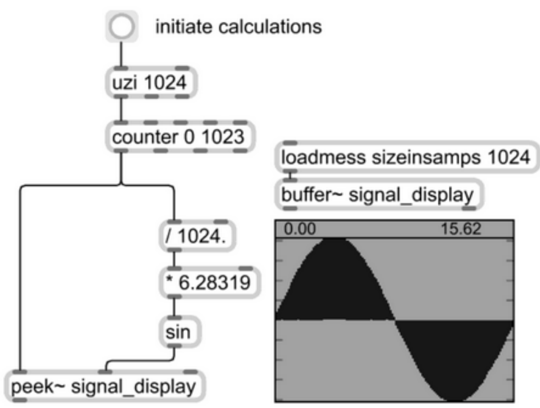

Figure 5: An example of a patch generating a sine wave on Max/MSP

SuperCollider is algorithmic based and less graphical (ref figure 7).

Web-based technologies, such as the Web Audio API [49], have further expanded accessibility, making interactive sonification more widespread.

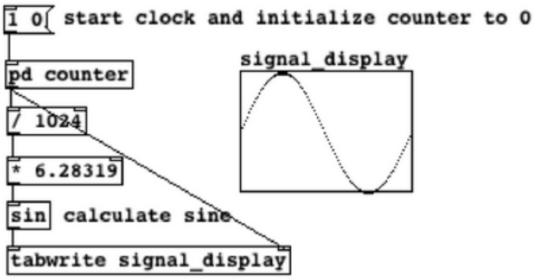

Figure 6: An example of a patch generating a sine wave on Pure Data

```
(
{               // Open the Function
    SinOsc.ar(    // Make an audio rate SinOsc
        440,     // frequency of 440 Hz, or the tuning A
        0,     // initial phase of 0, or the beginning of the cycle
        0.2)     // mul of 0.2
}.play;            // close the Function and call 'play' on it
)
```

Figure 7: An example of a patch generating a sine wave on SuperCollider

Peng and al. created a workstation called *SIREN* based on this Web Audio API [50]. The development of the Web Audio API really eases the development of new sonification frameworks but also the collaboration between artists. It reduces the investment necessary to use sonification software as all the people already has a web browser.

Even if new tools are emerging or old tool getting upgraded, sonification struggle to be totally widespread [51]. Artistic design continues to push the boundaries of how we interpret data through sound but it stills confined in research or artistic fields.

3 Plant as sensor

3.1 Technical choices

Before diving into the implementation, several technical choices were made to shape the overall design of the system. The first major decision was selecting the ESP32 microcontroller as the core processing unit due to its versatility, offering built-in WiFi, multiple GPIO pins, and analog-to-digital converters, which allow for handling sensor data and wireless communication efficiently.

Additionally, the system's audio capabilities were a critical consideration, leading to the inclusion of a basic amplification circuit to boost sound output, ensuring compatibility with external speakers. The design also included a micro-SD slot to allow for expandable storage, enabling future enhancements like storing pre-recorded sounds or more detailed data logs. These foundational decisions laid the groundwork for a scalable and reliable system, ensuring smooth integration of the components that follow in the implementation.

3.2 The electronic interface

The electronic interface is the interface that allows a compute unit to capture and interpret the plant signal and communication. The interface is a device made by us for this use case. The printed circuit board (PCB) device is composed of 3 main parts:

- The core of the circuit, the microcontroller, an ESP32 Wroom 32
- An electronic filter connected using an electrode to the plant
- A sound part of the PCB that is including an audio amplifier, a volume knob and a terminal block to connect a speaker

The design of the PCB has been done using the open source software Kicad. As said previously, the circuit contains 3 parts.

The core of the circuit is the computation part, including the microcontroller, an ESP32. All the other devices of the circuit are connected to the ESP32. The choice to use a devkit has been done to ease the electronic conception and to avoid any communication and soldering issue with the MCU*.

The PCB also includes a dedicated slot for a capacitive humidity sensor (ref figure 9). This humidity sensor is added to reinforced the data captured by the main sensor/filter. This sensor captures the moisture level in the plant's soil, providing valuable insight into the plant's overall health and hydration status.

By integrating this additional sensor, the system can monitor environmental factors alongside the main sensor's data, such as the plant's response to touch or other interactions. The humidity data complements the primary sensor readings which can be used to refine the interaction model or influence the sonification process. This added layer of data could permits more precision and complexity when doing the sonification process.

* Microcontroller Unit

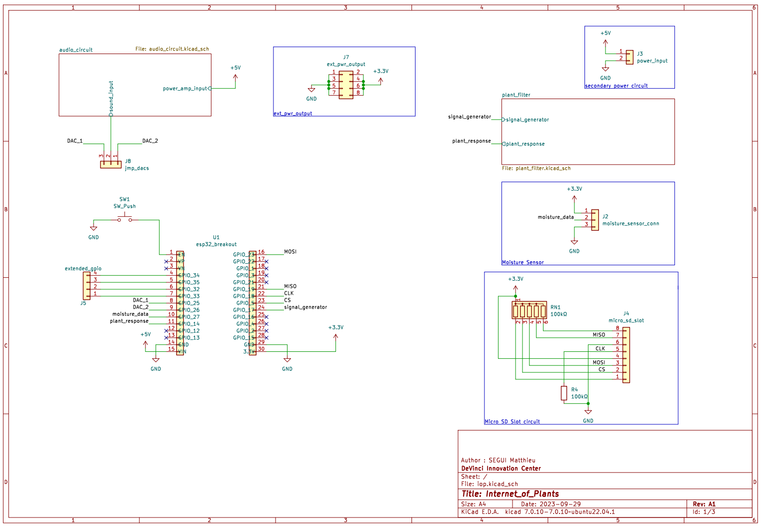

Figure 8: The main schematic of the device. This main schematic regroups components and sub-sheets (including the audio circuit and the filter circuit). The main component is an ESP32 Wroom DevKit. All other components are linked to this microcontroller

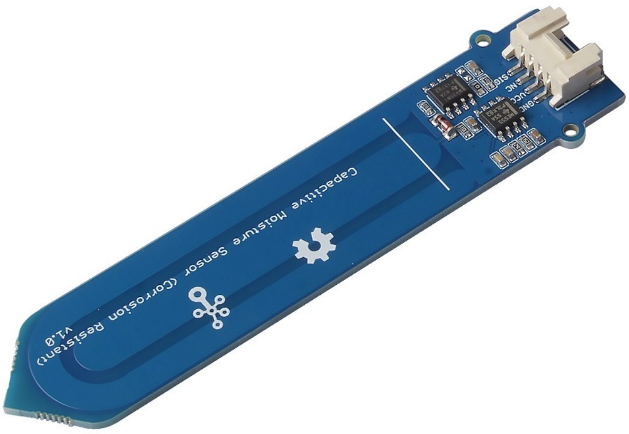

Figure 9: The sensor chosen is the capacitive moisture sensor from Seeed Studio: The Grove. This sensor is able to capture the moisture level in the plant's soil, providing valuable insight into the plant's overall health and hydration status. However, the connector on the PCB allows to connect any other sensor that has 3 pins (VCC, GND, DATA) connection.

The circuit component that allows us to read data from the plant is the electronic filter. This filter has been designed by *Jakub Nikonowicz* and *Łukasz Matuszewski* from *Politechnika Poznańska*. Thanks to them, I adapted it for my application on my embedded device.

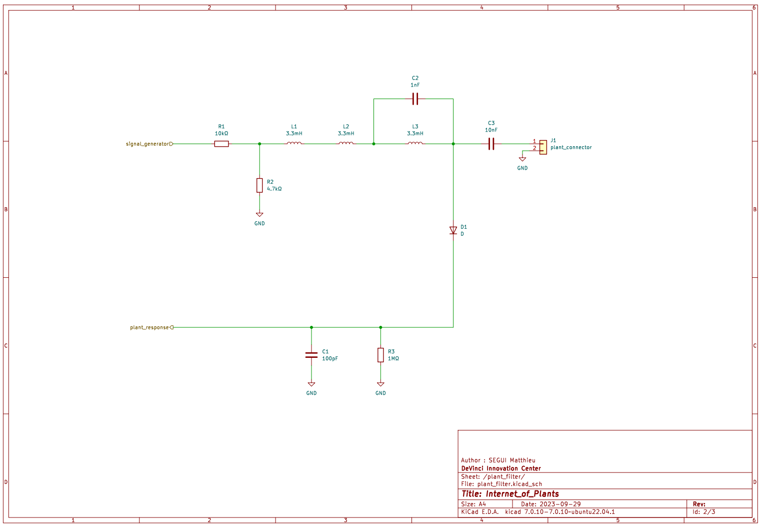

Figure 10: The electronic circuit designed to capture the interaction by analyzing the electronic frequency response. The circuit includes 3 resistors, 3 inductors and 3 capacitors as main components

This filter is ending by a crocodile clamp that is directly connected to the plant.

The last part of the circuit is the sound output/rendering. This circuit includes a small amplifier, the LM386 from Texas Instruments. The rest of the circuit are components needed in order to induce amplification on the signal without creating to many noise and saturation.

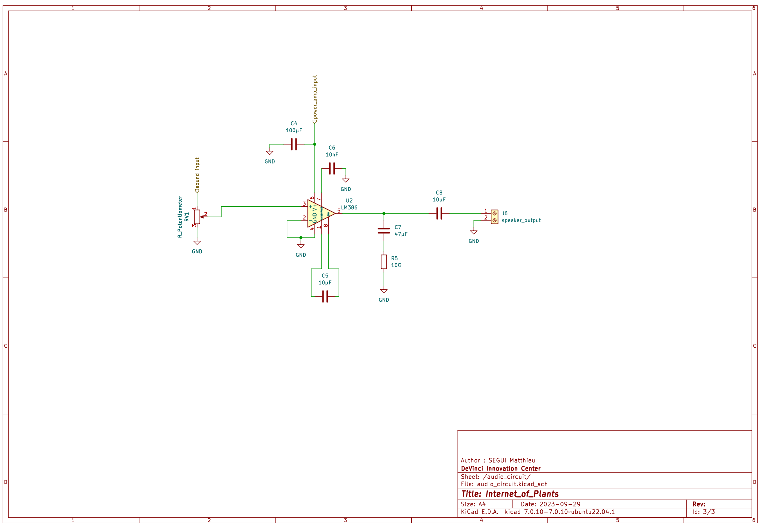

Figure 11: The sound output part of the circuit that is used to render the sound. This part includes a small amplifier, the LM386. The circuit also includes the components necessary to control and handle the amplification (reduce noise and saturation)

The PCB is built to be able to add a micro-SD slot (ref to 8). This micro-SD slot allows to increase the storage of the embedded ROM. This can be used to store pre-built sounds and music to avoid generating a sound on-board.

Once the schematic was completed, the next step was routing the tracks. There are several methods for PCB routing, including single-sided, double-sided, and multiple-layer designs. We opted for a double-sided PCB on each side because this configuration simplifies component placement, making the layout more efficient and organized. Additionally, it is significantly more cost-effective compared to multi-layer PCBs, striking a good balance between performance and budget. Indeed, in terms of routing, PCBs can be single-sided, double-sided, or multi-layered. Single-sided boards are simple and low-cost but limited in complexity. Double-sided PCBs, with copper on both sides, allow for more efficient routing and are commonly used for moderate complexity. Multi-layered PCBs offer even greater routing density, suitable for advanced electronics such as computer motherboards or phone components.

In terms of track width, we used 0.2mm wide tracks for data signals to maintain signal integrity, while the power tracks were designed with a width of 0.8mm. This wider width ensures that the power tracks can safely handle up to 800 mA, providing sufficient current capacity without overheating or voltage drops.

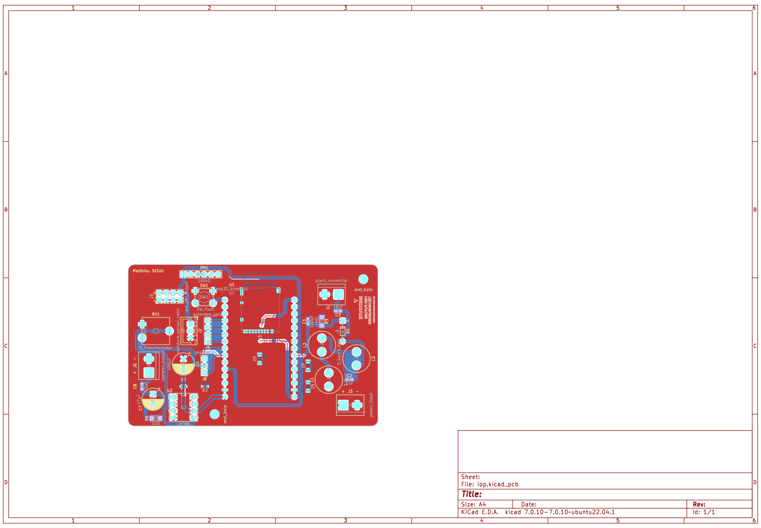

Figure 12: Double sided PCB routed. This view allows to see the track used for components inter-connection. The tracks width is 0.2mm for data tracks and 0.8mm for power tracks.

Kicad also allows us to generated a 3D view of the future PCB. This allows us to imagine what the PCB will look like when it will be manufactured.

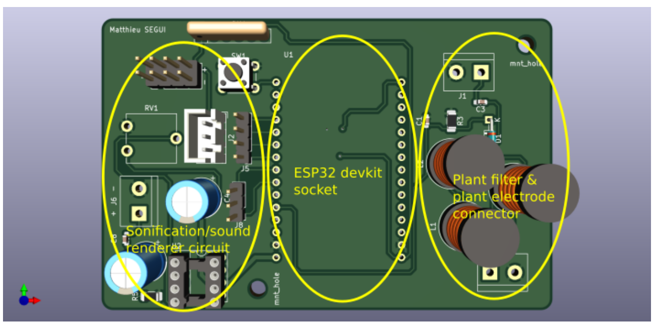

Figure 13: Front 3D rendering of the built PCB. The rendering is done using open source software: Kicad

3.3 Human interaction

3.3.1 Use of the sensor/filter

The device is able to capture the human interaction with the plant. The touch interaction is inducing changes in the impedance, capacitance and inductance of the plant. Values are captured using the GPIO (General Port Input/Output) 14 of the ESP32 DevKit. This GPIO is able to read analog data and convert them to digital values using analog to digital converter (ADC). The values are a floating point number between Values are fluctuating depending on the interaction.

The possibilities of

3.3.2 Sonification on the device

The human interaction goes through the sonification on the device. ESP32 embeds a 8-bit digital to analog converter (DAC). The embed DAC is enough to play low quality sounds. Combined with the micro-SD card including in it, it is a media player. Using the *Arduino Audio Tools* library it is possible to read MP3, Wav and other audio format. Taking the sensor value as an input, you can then apply a pitch shift on the data that will be rendered on the DAC. The pitch shift is a value included between 0 and 100. We are mapping the max value of the sensor and the min value to this range using the *map()* Arduino function.

The data is sent to the DAC and rendered. *Arduino Audio Tools* also includes a way to communicate data to the DAC.

There are two DAC available on the ESP32. DAC 1 and DAC 2 respectively on GPIO25 and GPIO26. The IoP device allows to choose between the one you want by using a removable jumper.

The output of the DAC is too low to be able to render it on a speaker. The amplification circuit allows a small amplification of the output. The amplifier is a basic and simple one. I added a 10k Ohms potentiometer to be able to tweak the volume. The sound quickly reach amplifier limitations and is becoming saturated.

The amplified output is sent to a terminal block that acts as the connection interface to the audio speaker. The audio speaker chosen in our specific example is a 8 Ohms 3 Watts speaker.

Figure 14: 8 ohms 3 Watts basic speaker used in our example.

3.3.3 User study

Abstract of user study This study explores human-plant interaction. This study has been conducted in order to understand what kind of interaction we have to detect in order to have the best and more natural kind of interaction with plants. The results will be applied in the Internet of Plants (IoP) project which intends to create a fully connected bio-organ system. The IoP is looking to reduce the gap between humans and plants by creating a symbiotic relationship between nature and technology. We envision a world where our daily objects are responsive.

Introduction Plants represent a full ecosystem of evolution, adaptation and communication.

In the context of the Internet of Plant (IoP) project, this study aims to extract the natural interaction between people and plants. This experiment explores the interactions the IoP device will have to detect to create a symbiotic relation between human and plants. The physical touch is the starting point of a sonification process. Sonification is "the use of non-speech audio to convey information or perceptualize data" [**kramerSonificationReportStatusa**]. Three distinct plant species—*Dypsis lutescens, Pachira glabra, and Dracaena*—are employed as subjects to extract user perceptions and interactions within this framework.

The methodology engages students from the engineering school and two researchers. The participants are asked to interact with the plants and imagine the sounds that could be generated by the plants.

The correlation between plant height and trunk interactions reveals environmental factors impacting human-plant dynamics. Additionally, interactions are categorized based on intensity, spatial displacement, and duration.

Methodology

Participants The study is conducted on 22 participants. Participants are mainly composed of engineering students. The participant set includes 15 males and 7 females. The age of participants is between 19 and 22 years old. Exception for three participants that are older than 22 years old.

The Procedure We introduced the subject telling participants : "We're in the very near future. You are looking at plants that make music when you physically interact with them (it is not actually the case, but imagine it). Explore their capabilities." Using this prompt, we tried not to bridle them to much but approach them to the physical interaction component. Subsequently, participants were given time to explore the potential musical capacities of the plants at their own pace. We conducted the study without providing any guidance during the exploration phase. In instances where participants encountered difficulty initiating exploration, the prompt was reiterated to encourage the participants to explore. This methodological approach was designed to capture the intuitive and natural human-plant interaction. Also, we avoided any kind of communication or talking between 2 participants to reduce the potential bias.

Materials/Tools To proceed and conduct this user study, we chose 3 different plants from 3 different species.

Dracaena: It has long leaves and fragile perceived trunk but also flexible. The plant is 95 cm tall.

Figure 15: The N°1 plant is a *Dracaena*.

Pachira glabra:We chose to use this plant for its large leaves and its wide trunk. The plant is 110 cm tall.

Figure 16: The N°2 plant is a *Pachira glabra*.

Dypsis lutescens: The *Dypsis lutescens* is composed of many trunks and stems. On top of that, the leaves are numerous and tight. The plant is ... tall.

Figure 17: The N°3 plant is a *Dypsis lutescens*.

The experimental space The experimental space featured three distinct levels of height, each corresponding to one of the three plants introduced to participants.

Figure 18: User study space setup. The setup is built from our lab space.

Data collection To capture the participant's interactions with the plants, a collaborative approach was adopted, involving two researchers to provide dual perspectives. Throughout the exploration phase, both researchers took notes, documenting the diverse ways in which participants engaged with the three distinct plants. The researchers explicitly specified the plant involved in the interaction in order to extract special features related to a specific plant.

The written notes retrieved descriptions of participants' actions, movements and interactions. The dual-observer strategy tends to reduce the potential biased.

At the beginning of the experiment, the *Dypsis lutescens* was on the floor, the *Dracaena* was on a chair and the *Pachira Glabra* was on a table. At the middle of the experiment, we switched the *Dypsis lutescens* and the *Pachira Glabra* to see if the participants would interact differently with the plants. The set-up of the experiment is shown in Figure 18.

Results The data given by the user study allowed us to define 5 main types of interaction. Those interactions are defined by the way the user interacts with the plant. The 5 main types of interaction are :

- Grasp : user uses the whole hand to grab trunk or leaves.
- Pinch : user uses 2 to 3 digits to grab trunk or leaves.
- Slide : user uses his/her hand or finger to slide on the plant whether is on a leave or on the trunk. The action is continuous.
- Pet : user uses his/her hand to cuddle the plant or to pass through the leaves. The user is moving his/her hand in space. She/he is not staying still or staying on a particular object.
- Tam Tam : user taps on the plant mainly using the whole hand.

Looking at the results, we extracted the table 1.

Table 1: Raw results extracted from the user study

Plant/Interaction	Group 1		Group 2	Group 3	
	Grasp	Pinch	Slide	Pet	Tam Tam
Plant N°1	4	8	4	4	2
Plant N°2	9	3	3	3	10
Plant N°3	10	1	5	7	3
Total	23	12	12	14	15

With the extraction of the result we were able to design a bar chart. The graph is grouping the interactions by plant. The height of the bar is the number of participants that performed the interaction. The graph is shown in figure 18.

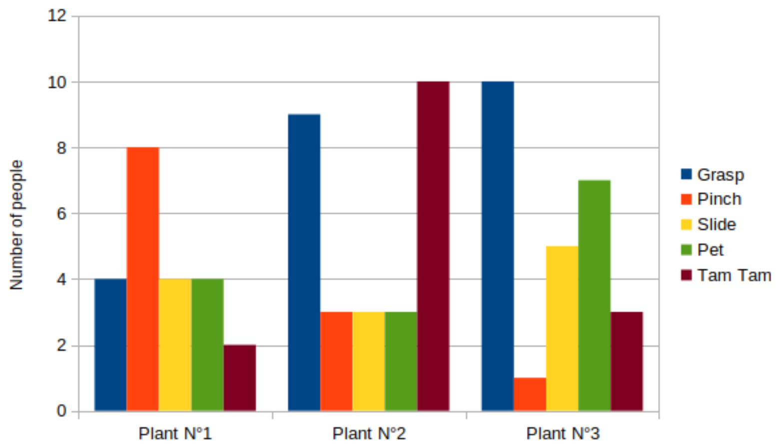

Figure 19: Bar chart that is extracting the main types of interaction regarding each plants.

In the end, of the 22 participants, 15 were already familiar with the project and 7 were not.

Discussion Looking at the results, the interaction were various depending on the plant. Thus, we can extract main interactions that are linked to the plant type. Looking at table 1, people are more inclined to use their hands as tam tam or grasp the *Pachira glabra*. However, for the *Dracaena* users prefer to pinch the trunk or leave. Participants decided to grasp whether a pack of trunk or leaves when it came to *Dypsis lutescens*. This is induced by many factors including the leaves shape, the width of the trunk.

It was observed that when the plants were positioned at higher elevations on the table, individuals tended to engage more with the trunk of the plants.

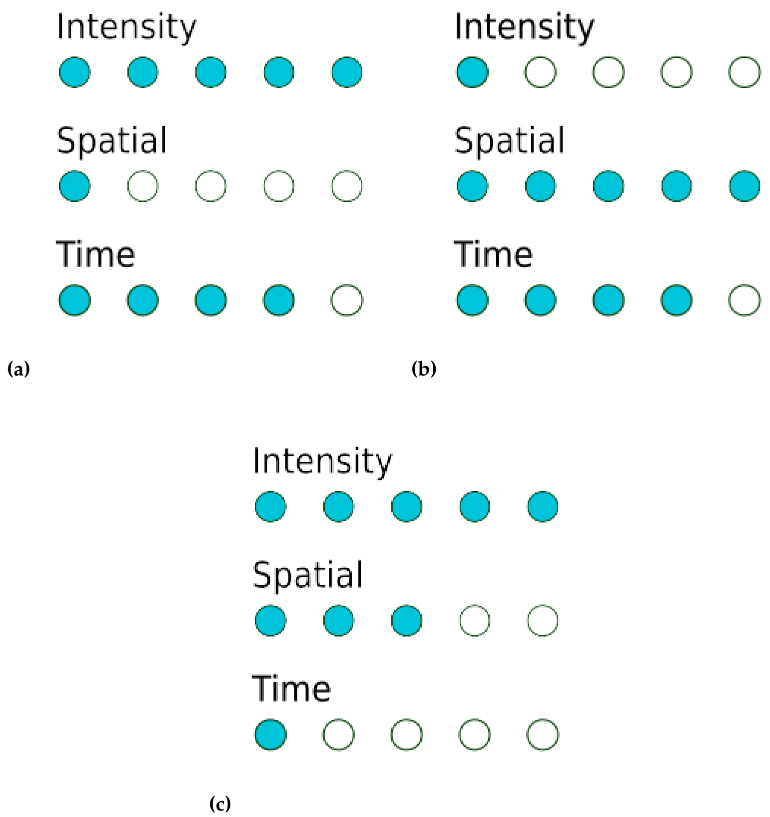

Figure 20: Figure showing graphically the intensity of the 3 types of factors we defined. (a) Group 1 : pinch and grasp. (b) Group 2 : slide. (c) Group 3 : pet and tam tam.

Looking at table 1, we decided to group interaction. This was done by grouping type of interaction depending on 3 main factors :

- The intensity factor : what is the intensity of the interaction (ex : pinch is lighter than grasp)
- The spatial factor : what is the interaction displacement.
- The duration factor : what is the interaction duration (ex : tam tam is instantaneous).

The "Group 1" includes the pinch and grasp interaction. Indeed, looking at the 3 factors we defined, the pinch and grasp are high in intensity and long in duration but people stay still in space. This group of interaction can be defined as **binary interaction**. The user is either grasping or not.

The "Group 2" includes the slide. The slide interaction is long in time, it moves in space but low in intensity. This group of interaction can be defined as **continuous interaction**.

Whereas, the "Group 3" includes the pet and Tam Tam. These 2 interactions are really high in intensity, people usually tam tam and pet in different places but those interactions are short in time. This group is defined as **repetitive interaction**. The user is repeating the same action over and over again.

The participants we interviewed introduced a bias in the results. They were all students from the engineering school and thus, they all had a similar background. Some of them were already familiar with the project.

Conclusion During our study on the Internet of Plant project, we've captured insights into how people might interact with plants in a future where they make music through touch.

Our three chosen plants influenced how participants engaged with them. We observed everything from gentle petting to energetic drumming on the plants. Interestingly, we found that when the plants were higher up, participants tended to focus more on the trunk.

By grouping interactions based on factors like intensity and duration, we gained a clearer picture of how people approached these musical plants. It turns out that certain interactions, like grasping and pinching, were more common, while others, like sliding, had their own distinct appeal.

Regarding to the results we thought about what could be done with the defined interactions. For instance, the sound generated from the interaction could be linked to the kind of interaction. People doing Tam Tam on the plant will expect a drum sound. Whereas, people performing a slide will expect a sound closer to a continuous organ sound. The possibilities are endless and the only restrictions are the capabilities of the device capturing the interaction.

3.4 Evaluation ?

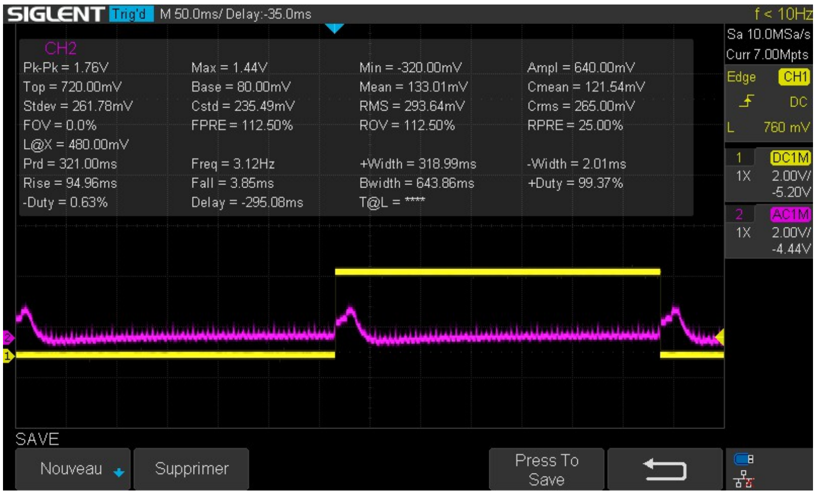

Figure 21: Signals coming from the system. The yellow signal is the signal generated by the ESP32 (PWM). The purple one is the signal that we capture at the output of the filter. This signal is the one that goes in the Plant.

3.5 Discussion

The final product is an embedded device that include signal filtering, wireless communication and embedded sonification.

A better audio amplifier could be explored in order to reduce the distortion of the sound. Adding an external digital to analog converter is also a possibility in order to upgrade the output. However, this possibility adds new components that will increase the size. Exploring the I2S protocol opens better output. The I2S (Inter-IC Sound) protocol is used to transmit digital audio between devices like microcontrollers and DACs. It enables high-quality stereo audio transfer, using a master-slave setup to synchronize data with clock signals. Ideal for applications needing accurate sound transmission.

3.6 Conclusion

In conclusion, the standalone electronic system demonstrates significant potential in transforming natural plants into interactive bio-sensors, using the capabilities of a powerful microcontroller like the ESP32. The system's ability to capture plant responses to human touch and translate them into digital data is made possible by its efficient electronic interface. This architecture enables real-time interaction and offers new approach to human-plant interaction. This is also driving research into the use of plants' natural capacities as sensors. However, despite the microcontroller's strengths, the system still has notable limitations.

One major challenge lies in the sonification process, where the system struggles with producing high-quality, nuanced sound outputs due to the basic 8-bit DAC and limited audio amplification. Additionally, the data processing capabilities of the standalone device are limited, restricting the complexity of interactions it can detect and interpret. The sensor accuracy also has room for improvement, particularly in capturing fine-grained variations in plant interaction, which could taint the overall user experience and reduce the immersion.

To overcome these standalone limitations, a more sophisticated architecture is required. By connecting multiple devices in a distributed system, the Internet of Plants approach can enhance the processing power, improve sonification through external software, and allow for more complex data analysis. It could unlock the full potential of plant-based sensors. The next section will explore how this expanded network can significantly enhance both the system's capabilities and user experience.

4 Internet of Plants

4.1 Overview

The Internet of Plants, also called IoP, is a concept that aims to interconnect the plant device previously built. This in order to empower the device capabilities and to provide a better user experience. This project includes:

- A better sound quality by using professional sonification software
- The ability to create a full artistic experience by creating a distributed instrument
- Refining the interaction with the plant by using more complex data analyses

4.2 Communication

On the communication side, we benchmarked several communication protocol including internet protocol, Bluetooth, Bluetooth Low Energy and Zigbee (ref to table 2).

The IP protocol (whether it is on WiFi or Ethernet) has been chosen for this application for several reasons:

- **High bandwidth**: It supports the transmission of large amounts of data quickly and efficiently, which is essential for real-time interactions in the system.
- **Widespread availability**: WiFi and Ethernet are commonly available in most exhibition spaces, ensuring easy deployment in a variety of environments.
- **Multi-device connectivity**: The protocol allows the connection of multiple devices simultaneously, which is crucial for building a distributed system where many plants can interact with the server.
- **Compatibility**: Ethernet is already available on the server, and WiFi is built into the ESP32, eliminating the need for additional hardware.

The communication system is based on WiFi technology, leveraging the ESP32's built-in wireless capabilities. Both the server and the ESP32 devices are connected to the same local network. Data is transmitted from the ESP32 to the server using the IP protocol via a TCP (Transmission Control Protocol) socket. Each ESP32 opens a dedicated TCP socket to the server, and the server is capable of handling multiple sockets simultaneously—one for each connected device. The server is connected to the local network via an Ethernet cable, providing a stable, high-speed connection for data processing and sonification (ref figure 22).

However, a potential drawback of using IP networks is that they can become overcrowded, especially in environments with high network traffic, which may lead to packet loss or truncated data frames. To mitigate

Protocol	IP	Bluetooth	BLE	Zigbee
Handle multiple connections	Yes	No	No	Yes
Requires additional hardware	No	No	No	Yes
Subject to interference	Yes	Few	Few	Yes
Energy efficiency (using a battery)	Days	Months	Years	Years

Table 2: Comparison between different communication protocol to find the one that will suits our needs.

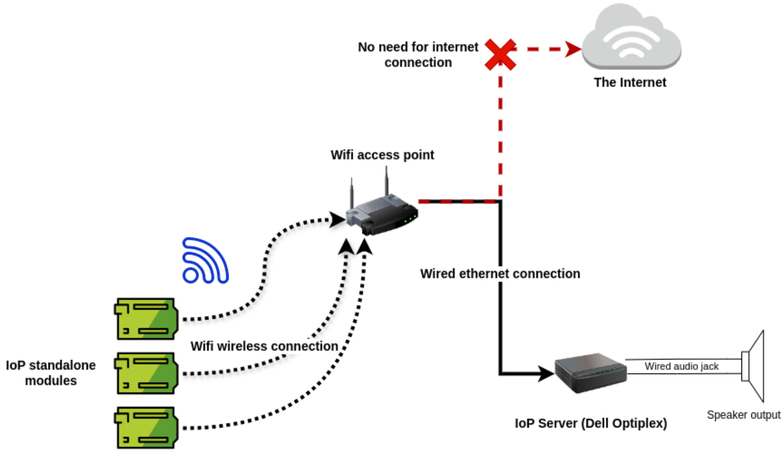

Figure 22: Schematic of the network architecture of the Internet of Plants project. The standalone modules are connected to the server using WiFi. The server is connected via Ethernet cable to the local access point. The server is also connected to a jack speaker working as output.

these issues, a start ("") and stop (";
n") character is embedded in the data transmission protocol. This structure helps the server recognize the beginning and end of each message, allowing it to discard incomplete or corrupted frames and process only complete, intact data. This approach minimizes the impact of network congestion by ensuring that even if a frame is truncated or lost, the server can still recover and process the next full message without error.

Additionally, this strategy helps maintain a balance within the CAP theorem (Consistency, Availability, and Partition tolerance). While it is impossible to achieve all three properties perfectly in a distributed system, the chosen protocol structure ensures data consistency by filtering out incomplete messages, availability by maintaining active connections across multiple devices, and some degree of partition tolerance in the event of temporary network failures. This balance is critical for ensuring reliable data transmission and processing in the Internet of Plants system.

4.3 Server

The server is a small fanless computer (Dell Optiplex) running Lubuntu. Lubuntu is a lighter version of Ubuntu that includes LXQt as desktop environnement. The choice of a distribution with graphical interface is induced by the use of *Pure Data* as sonification software. However, the server can be run without graphical interface when on production mode. The choice of the light desktop environment is induced by the fact that the server is a low resources computer. The server only include 1 GB of RAM which is a limitation. The server is connected to the local network using an Ethernet cable. The server is also connected to a jack speaker (ref figure 22).

Pure Data (PD) is an open-source visual programming language designed primarily for creating interactive multimedia applications, particularly in the fields of audio, video, and graphical processing. *Pure Data* is part of a family of patcher programming languages, which also includes Max/MSP. Unlike traditional text-based programming, PD uses a graphical interface where users connect "objects" with virtual patch cables to create complex data flows and signal processing chains. Its modular design allows for real-time manipulation of sound and graphics, making it a powerful tool for artists, musicians, and researchers interested in exploring experimental media.

Pure Data requires, in a development environnement, a graphical interface to test and debug. The *Pure Data* patch receive the data through a TCP socket. The data is processed through several operations. Then, if a threshold is passed, an interaction happened and the music is triggered. The music is outputted through the item *DAC* which means Digital to Analog Converter. The digital input from *Pure Data* is converted to a sound that speakers can output.

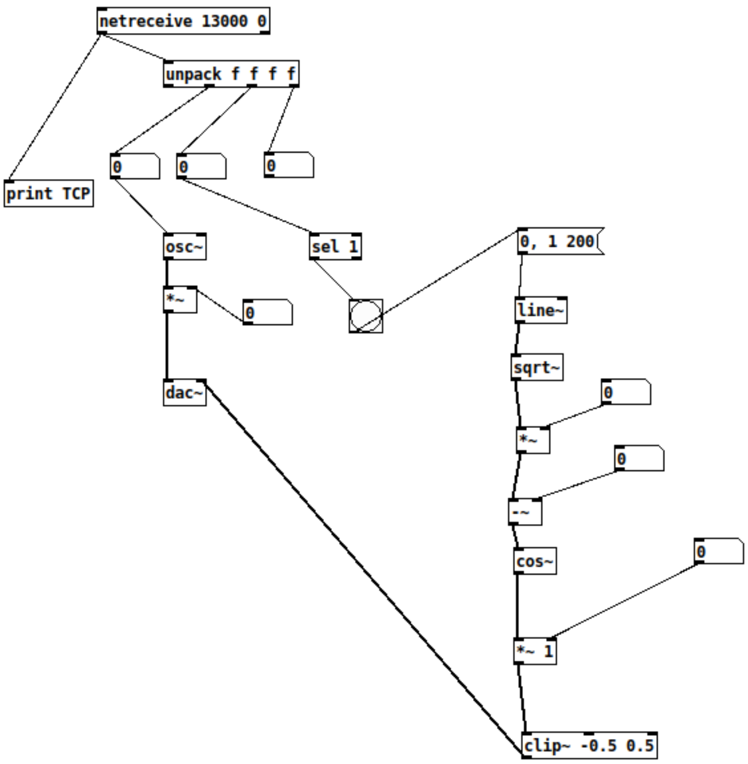

Figure 23: Basic *Pure Data* patch that is used for sonification of the data of the plant. In case of an art exhibition, the *Pure Data* patch can be upgraded to meet the artist needs

Pure Data is a sonification software that require data as input. In order to get and process the data, we designed a Python based software. The software is object oriented. The standalone IoP modules connect using WiFi to the receiver module of the software. The connection is made using TCP socket from the ESP32 to the server. The main module of the software then creates software abstraction of the standalone module. The abstraction module is processing and storing all the processed data. The main module then send the processed data to *Pure Data* patch using also local TCP socket.

4.4 Deployment

The server software is easily deployable. The software includes a shell script to deploy the server service (ref figure **??**). Indeed, the server relies on a *systemd* linux service. *Systemd* [52] is a Linux software that manage application that runs *daemons* or services. *Daemons* are pieces of software that run in background of the operating system. They are mainly started

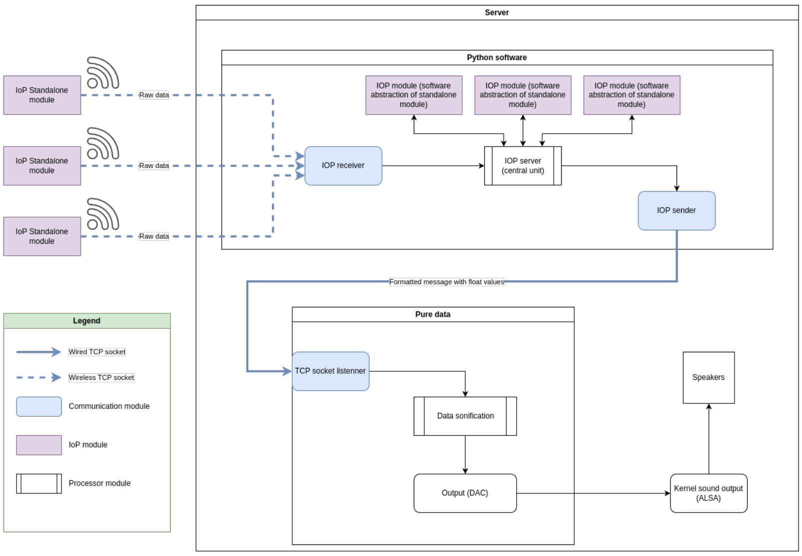

Figure 24: Architecture diagram of Internet of Plants project centered on server-side.

at the during the boot of the operating system.

The server is deployed using systemd service to allow the software to start with the operating system. It waits for the network interfaces to be up and running and opens the socket. The installation tool install Python dependencies, fill in the templated service file, install the service and enable it. The server is able to receive data from IoP standalone modules and use *Pure Data* software for sonification. The last step is to connect a jack speaker to the jack builtin output.

On the IoP standalone module side, it requires a software that is able to upload firmware to an ESP32 MCU. I recommend using PlatformIO which is an open source embedded software development platform. The firmware is developed using this platform. The source code is written in C++ using the Arduino framework. You flash the firmware to the chip after setting up the WiFi credentials. The module sends all the retrieved data to the server.

4.4.1 Distributed instruments

The Internet of Plants (IoP) architecture enables the creation of a distributed instrument, where multiple IoP standalone modules can be deployed across different plants to form an interconnected musical system. Each module, linked to the server via WiFi, sends data collected from plant interactions, which the server then processes. By assigning unique IDs to each module, the server can differentiate between the various plants, allowing for distinct musical outputs based on the origin of the data.

The system sends this data to *Pure Data* (PD), an open-source visual programming language for multimedia applications, where sound parameters such as pitch, tone, and rhythm can be finely controlled. This setup allows for highly customizable soundscapes, where each plant generates unique sounds based on user interaction, creating a rich, immersive musical experience. The distributed instrument thus transforms the interaction with plants into a collaborative musical performance, opening up new possibilities for art installations and interactive environments.

4.5 Results

4.5.1 Evauation

4.5.2 Art exhibition

To push further the distributed instrument, it is possible to build an entire musical experience for an art exhibition. The immersive experience can take place as a fully connected forest. The music would vary depending on the touch interaction that people have with the plants.

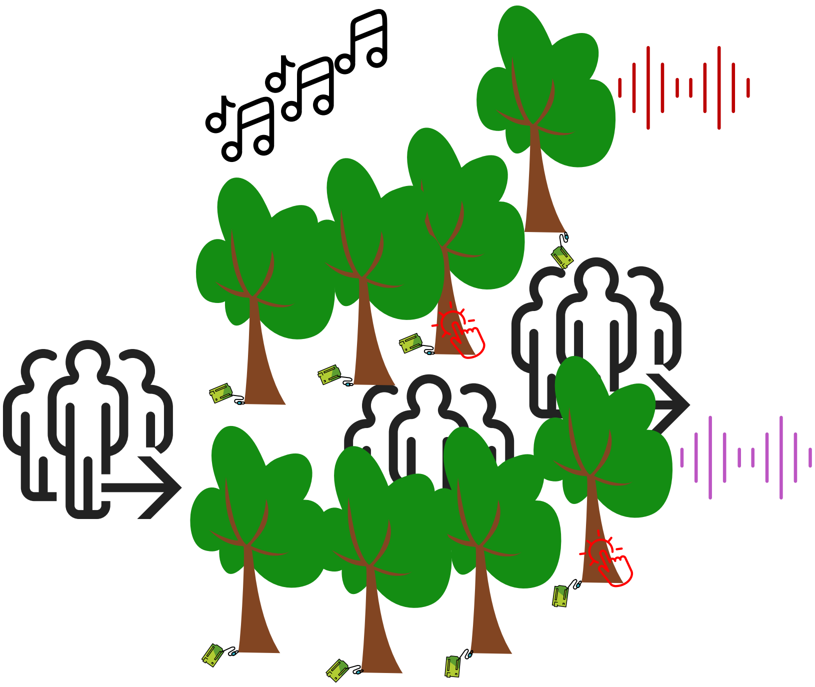

Figure 25: Schematic of the IoP art exhibition. The music varies depending on the interaction that people have with the installation. People are immersed into the full musical and sonification experience.

This exhibition could allow people to rethink the way the see and interact with plants. Plants are not seen anymore as decoration object but as a living being. The living being is here highlighted by the fact that plants can now express their state. The humidity level, the light intensity and other state signs are modifying the plant reaction.

Musicians can also work in pair with plants to create songs using music from the plants. The music coming from the plant is fully configurable with the sonification software.

4.5.3 Final product, limitations and future work

Final product is a software that allows the connection between multiple standalone IoP modules and a sonification software, *Pure Data*. The software handle multiple connection and apply a filtering and cleaning on the data retrieved. The data is then sent to *Pure Data* for sonification.

As a demo and in order to do a photo shooting, we built up a system using the standalone module, connected through wire (ref figure 26). We used the serial communication in this demo because we did not have wireless communication usable at the shooting site. The computer is showing a visualization from the incoming data. This is a waterwall like visualization (ref figure 27). The data is processed and displayed on the computer screen. The data is normalized on the y-axis. The x-axis is the sweep frequency. The graph signal at the bottom is the real-time signal. The computer is also producing the sound using *Pure Data* software.

Figure 26: Server-module demo. In this specific use case, the module is connected by wire to a computer. Otherwise, the module is usable wirelessly.The computer is acting as a server. On this version of the demo, the screen is displaying a figure that is evolving depending on the touch interaction. The computer, acting as the server, is producing the sound using *Pure Data* software.

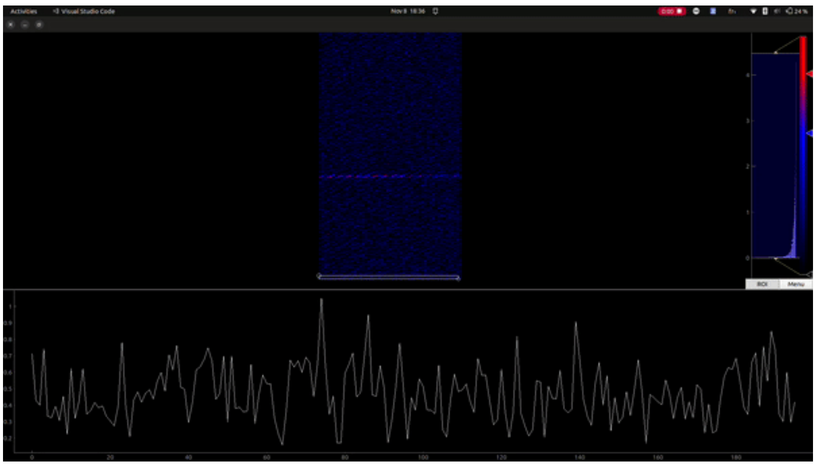

Figure 27: Waterfall like visualization of the data extracted from the standalone module. The data is processed and displayed on the computer screen. The data is normalized on the y-axis. The x-axis is the sweep frequency. The graph signal at the bottom is the real-time signal.

The Internet of Plants system faces several limitations in its current implementation, particularly in the areas of communication and data reliability. One major challenge is the use of WiFi as the primary communication protocol. In environments with high network traffic, WiFi can become overcrowded, leading to dropped frames and inconsistent data transmission. This congestion affects the real-time interaction between plants and the sonification process.

Another limitation is the potential for frame loss during data transmission. Although the system incorporates start and stop tags to verify the completeness of data frames, this method is not sufficient to prevent the loss of frames in overloaded networks. The absence of a more advanced error-handling mechanism results in incomplete or corrupted data, which reduces the accuracy of the system.

Furthermore, there is limited fault tolerance in the current communication setup. The system lacks mechanisms to ensure consistent data flow or recovery from network failures, which can disrupt the continuous interaction between the plants and the server.

Lastly, there is inadequate monitoring of the distributed network. The system does not provide real-time feedback on the health of the network or the individual plant sensors, making it difficult to detect and address failures quickly.

4.6 Conclusion

In conclusion, the Internet of Plants (IoP) is a revolutionary application of IoT principles to plant systems, enabling more sophisticated interaction between humans and plants. By leveraging distributed systems, sensors, and sonification software, the IoP creates immersive environments where plants can become interactive instruments, generating real-time musical responses based on human touch. This innovation extends beyond artistic installations, providing new opportunities in agriculture, environmental monitoring, and human-computer interaction. However, limitations such as network congestion, sensor accuracy, and audio quality need further work to fully unlock the potential of plant-based sensor networks.

5 Conclusion

In conclusion, this thesis explored the innovative potential of using plants as bio-sensors within a system that integrates hardware and software components to capture human-plant interaction and transform it into auditory experiences. Through the development of a standalone electronic system, including sensors, data processing capabilities, and sonification features, this research presents a brand new intersection of human-computer interaction, biology, and sound design.

The Internet of Plants (IoP) framework proposed here introduces a new way to think about plant-environment interaction, increasing human awareness of plants' responses to touch. The real-time sonification enabled by the IoP allows for immersive, interactive art installations and potential applications in fields such as agriculture and environmental monitoring. Despite the promising results, several limitations were identified, particularly regarding sensor accuracy, sound quality, and system scalability, which will require further investigation and improvement.

Future work may focus on overcoming these limitations by integrating more sophisticated sensors and audio technologies, while exploring the potential for broader IoT applications. Additionally, expanding the distributed system concept could enable more complex networks of interactive plants, offering new possibilities for environmental data collection, art, and human-plant symbiosis.

Acknowledgements

I would like to start by thanking my principal investigator, Dr. Duhart Clément for his guidance throughout this research project.

I would also like to thank my colleague and friend, Nicolas Stas for his support and encouragement during the development of this work. His ideas and feedbacks have been invaluable in shaping the direction of this research.

I would like to express my gratitude to Audrey Béarel for her unconditional support through all the success and failures of this project.

I would like to thank Jérôme Béarel for his help in times of emergency.

I also say thank you to my friends and colleagues who have supported me during this journey including, but not limited to, Claudio, Thomas, Mathis, Augustin.

Finally, I would like to thank my family for their unwavering support and encouragement throughout this journey.

References

Here are the references in citation order.

[1] John M Carroll. 'HUMAN-COMPUTER INTERACTION: Psychology as a Science of Design'. In: () (cited on page 3).

[2] Charles Hall and Melinda Knuth. 'An Update of the Literature Supporting the Well-Being BeneÂ¢ts of Plants: A Review of the Emotional and Mental Health BeneÂ¢ts of Plants'. In: (2019) (cited on page 4).

[3] Greg Watts. 'The Effects of "Greening" Urban Areas on the Perceptions of Tranquillity'. In: *Urban Forestry & Urban Greening* 26 (Aug. 2017), pp. 11–17. DOI: 10.1016/j.ufug.2017.05.010. (Visited on 06/26/2024) (cited on page 4).

[4] Harumi Ikei et al. 'The Physiological and Psychological Relaxing Effects of Viewing Rose Flowers in Office Workers'. In: *Journal of Physiological Anthropology* 33.1 (Dec. 2014), p. 6. DOI: 10.1186/1880-6805-33-6. (Visited on 06/26/2024) (cited on page 4).

[5] Min-sun Lee et al. 'Interaction with Indoor Plants May Reduce Psychological and Physiological Stress by Suppressing Autonomic Nervous System Activity in Young Adults: A Randomized Crossover Study'. In: *Journal of Physiological Anthropology* 34.1 (Dec. 2015), p. 21. DOI: 10.1186/s40101-015-0060-8. (Visited on 10/02/2024) (cited on page 4).

[6] Craig Brod. 'Technostress: The human cost of the computer revolution'. In: *(No Title)* (1984) (cited on page 4).

[7] Ayyagari, Grover, and Purvis. 'Technostress: Technological Antecedents and Implications'. In: *MIS Quarterly* 35.4 (2011), p. 831. DOI: 10.2307/41409963. (Visited on 10/02/2024) (cited on page 4).

[8] Ahmad Hassan et al. 'Effects of Plant Activity on Mental Stress in Young Adults'. In: *HortScience* 53.1 (Jan. 2018), pp. 104–109. DOI: 10.21273/HORTSCI12447-17. (Visited on 10/02/2024) (cited on page 4).

[9] Stephen R. Kellert; Judith Heerwagen; Martin Mador. *Biophilic Design: The Theory, Science and Practice of Bringing Buildings to Life*. Wiley, 2008 (cited on page 4).

[10] Paul Downton et al. 'Biophilic Design Applications: Putting Theory and Patterns into Built Environment Practice'. In: *KnE Engineering* 2.2 (Feb. 2017), p. 59. DOI: 10.18502/keg.v2i2.596. (Visited on 10/07/2024) (cited on page 4).

[11] Stephen R Kellert. 'The Biophilia Hypothesis'. In: () (cited on page 4).

[12] Stephen Kellert and Elizabeth Calabrese. 'The practice of biophilic design'. In: *London: Terrapin Bright LLC* 3.21 (2015) (cited on page 4).

[13] Catherine O. Ryan and William D. Browning. 'Biophilic Design'. In: *Encyclopedia of Sustainability Science and Technology*. Ed. by Robert A. Meyers. New York, NY: Springer New York, 2018, pp. 1–44. DOI: 10.1007/978-1-4939-2493-6_1034-1. (Visited on 10/07/2024) (cited on page 5).

[14] Wa Gao et al. 'Integrating User-Centered Design and Biophilic Design to Improve Biophilia and Intelligentization in Office Environments'. In: *Buildings* 13.7 (June 2023), p. 1687. DOI: 10.3390/buildings13071687. (Visited on 10/07/2024) (cited on page 5).

[15] S. Vigneshvar et al. 'Recent Advances in Biosensor Technology for Potential Applications – An Overview'. In: *Frontiers in Bioengineering and Biotechnology* 4 (Feb. 2016). DOI: 10.3389/fbioe.2016.00011. (Visited on 10/07/2024) (cited on page 5).

[16] Christiane Ziegler and Wolfgang Göpel. 'Biosensor Development'. In: *Current Opinion in Chemical Biology* 2.5 (Jan. 1998), pp. 585–591. DOI: 10.1016/S1367-5931(98)80087-2. (Visited on 10/07/2024) (cited on page 5).

[17] Hui-Chen Wang and An-Rong Lee. 'Recent Developments in Blood Glucose Sensors'. In: *Journal of Food and Drug Analysis* 23.2 (June 2015), pp. 191–200. DOI: 10.1016/j.jfda.2014.12.001. (Visited on 10/07/2024) (cited on page 6).

[18] Izabela Anna Chincinska. 'Leaf Infiltration in Plant Science: Old Method, New Possibilities'. In: *Plant Methods* 17.1 (July 2021), p. 83. DOI: 10.1186/s13007-021-00782-x. (Visited on 10/07/2024) (cited on page 6).

[19] Olivia Seow et al. ‘Pudica: A Framework For Designing Augmented Human-Flora Interaction’. In: *Augmented Humans 2022*. Kashiwa, Chiba Japan: ACM, Mar. 2022, pp. 40–45. DOI: 10.1145/3519391.3519394. (Visited on 10/24/2022) (cited on page 6).

[20] Alexander G. Volkov et al. ‘*Mimosa Pudica* : Electrical and Mechanical Stimulation of Plant Movements’. In: *Plant, Cell & Environment* 33.2 (Feb. 2010), pp. 163–173. DOI: 10.1111/j.1365-3040.2009.02066.x. (Visited on 05/26/2024) (cited on page 6).

[21] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. ‘Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects’. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Austin Texas USA: ACM, May 2012, pp. 483–492. DOI: 10.1145/2207676.2207743. (Visited on 05/26/2024) (cited on pages 7, 8).

[22] Ivan Poupyrev et al. ‘Botanicus Interacticus: Interactive Plants Technology’. In: *ACM SIGGRAPH 2012 Emerging Technologies*. Los Angeles California: ACM, Aug. 2012, pp. 1–1. DOI: 10.1145/2343456.2343460. (Visited on 05/26/2024) (cited on page 7).

[23] Colin Honigman, Jordan Hochenbaum, and Ajay Kapur. ‘Techniques in Swept Frequency Capacitive Sensing: An Open Source Approach’. In: () (cited on page 7).

[24] Ken Hinckley and Mike Sinclair. ‘Touch-Sensing Input Devices’. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems the CHI Is the Limit - CHI ’99*. Pittsburgh, Pennsylvania, United States: ACM Press, 1999, pp. 223–230. DOI: 10.1145/302979.303045. (Visited on 08/18/2024) (cited on page 7).

[25] Simon Olberding et al. ‘A Cuttable Multi-Touch Sensor’. In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. St. Andrews Scotland, United Kingdom: ACM, Oct. 2013, pp. 245–254. DOI: 10.1145/2501988.2502048. (Visited on 08/18/2024) (cited on page 7).

[26] Chia-Yen Lee and Gwo-Bin Lee. ‘Humidity Sensors: A Review’. In: *Sensor Letters* 3.1 (Jan. 2005), pp. 1–15. DOI: 10.1166/sl.2005.001. (Visited on 10/05/2024) (cited on page 8).

[27] Burcu Arman Kuzubasoglu. ‘Recent Studies on the Humidity Sensor: A Mini Review’. In: *ACS Applied Electronic Materials* 4.10 (Oct. 2022), pp. 4797–4807. DOI: 10.1021/acsaelm.2c00721. (Visited on 10/05/2024) (cited on page 8).

[28] T Hermann and H Ritter. ‘Listen to Your Data: Model-Based Sonification for Data Analysis’. In: () (cited on page 8).

[29] Mark Ballora et al. ‘Use of sonification in the detection of anomalous events’. In: *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2012*. Vol. 8407. SPIE. 2012, pp. 213–219 (cited on page 9).

[30] Adian F Rochim, Mukhlish A Aziz, and Adnan Fauzi. ‘Design Log Management System of Computer Network Devices Infrastructures Based on ELK Stack’. In: *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*. IEEE. 2019, pp. 338–342 (cited on page 9).

[31] Gareth Loy. ‘Musicians Make a Standard: The MIDI Phenomenon’. In: *Computer Music Journal* 9.4 (1985), p. 8. DOI: 10.2307/3679619 (cited on page 9).

[32] I Fazenda, Sao Paulo, and I S Junior. ‘Proceedings of the International Conference on New Interfaces for Musical Expression’. In: () (cited on page 9).

[33] I Fazenda, Sao Paulo, and I S Junior. ‘Proceedings of the International Conference on New Interfaces for Musical Expression’. In: () (cited on page 9).

[34] Tanvir Hussain and Md Mejbaul Haque. ‘AVR Microcontroller Implementation for Customized Sound Generation’. In: *International Journal of Electrical and Computer Engineering (IJECE)* 2.1 (Nov. 2011), pp. 112–119. DOI: 10.11591/ijece.v2i1.139 (cited on page 9).

[35] Bassam Shaer et al. ‘Interactive Capacitive Touch Music Table with Embedded Microcontrollers’. In: *The Journal of Supercomputing* 76.11 (Nov. 2020), pp. 8845–8865. DOI: 10.1007/s11227-020-03167-4 (cited on page 9).

[36] Khurshid Aliev et al. ‘Internet of Plants Application for Smart Agriculture’. In: *International Journal of Advanced Computer Science and Applications* 9.4 (2018). DOI: 10.14569/IJACSA.2018.090458. (Visited on 08/24/2024) (cited on page 10).

[37] Peter G. Steeneken et al. ‘Sensors in Agriculture: Towards an Internet of Plants’. In: *Nature Reviews Methods Primers* 3.1 (Aug. 2023), p. 60. DOI: 10.1038/s43586-023-00250-x. (Visited on 08/24/2024) (cited on page 10).

[38] Masaharu Kitano et al. 'Internet of Plants (IoP) Empowers Bottom-up Innovations in Greenhouse Horticulture'. In: *Environment Control in Biology* 60.1 (Jan. 2022), pp. 3–12. DOI: 10.2525/ecb.60.3. (Visited on 08/24/2024) (cited on page 11).

[39] Maarten van Steen and Andrew S. Tanenbaum. *Distributed Systems*. Third edition, Version 3.01. Erscheinungsort nicht ermittelbar: Maarten van Steen, 2017 (cited on page 11).

[40] Leonard Kleinrock. 'Distributed Systems'. In: () (cited on page 11).

[41] Dr Eric A Brewer. 'Towards Robust Distributed Systems'. In: (2000) (cited on page 11).

[42] Jeffrey Joyce, Greg Lomow, and Brian Unger. 'Monitoring Distributed Systems'. In: *ACM Transactions on Computer Systems* 5.2 () (cited on page 11).

[43] Roger Thornton Dean. *The Oxford Handbook of Computer Music*. Oxford Handbooks. New York (N.Y.): Oxford University Press, 2009 (cited on page 11).

[44] Atau Tanaka. *Sensor-based musical instruments and interactive music*. na, 2009 (cited on page 11).

[45] A. Barbosa and M. Kaltenbrunner. 'Public Sound Objects: A Shared Musical Space on the Web'. In: *Second International Conference on Web Delivering of Music, 2002. WEDELMUSIC 2002. Proceedings.* Darmstadt, Germany: IEEE Comput. Soc, 2002, pp. 9–16. DOI: 10.1109/WDM.2002.1176188. (Visited on 10/03/2024) (cited on page 12).

[46] Gregory Kramer et al. 'Sonification Report: Status of the Field and Research Agenda'. In: () (cited on page 12).

[47] Miller S Puckette. 'Pure Data: Another Integrated Computer Music Environment'. In: () (cited on page 12).

[48] James McCartney. 'Rethinking the Computer Music Language: Super Collider'. In: *Computer Music Journal* 26.4, (2002), pp. 61–68 (cited on page 12).

[49] Boris Smus. *Web audio API: advanced sound for games and interactive apps*. " O'Reilly Media, Inc.", 2013 (cited on page 12).

[50] Tristan Peng and Hongchan Choi. 'SIREN: A Case Study in Web Audio Based Sonification'. In: *Proceedings of the 26th International Conference on Auditory Display (ICAD 2021)*. Virtual Conference: International Community for Auditory Display, June 2021, pp. 126–130. DOI: 10.21785/icad2021.042. (Visited on 10/04/2024) (cited on page 13).

[51] John G. Neuhoff. 'Is Sonification Doomed to Fail?' In: *Proceedings of the 25th International Conference on Auditory Display (ICAD 2019)*. Newcastle upon Tyne: Department of Computer and Information Sciences, Northumbria University, June 2019, pp. 327–330. DOI: 10.21785/icad2019.069. (Visited on 10/04/2024) (cited on page 13).

[52] David Both. 'systemd'. In: *Using and Administering Linux: Volume 2: Zero to SysAdmin: Advanced Topics*. Berkeley, CA: Apress, 2020, pp. 379–410. DOI: 10.1007/978-1-4842-5455-4_13 (cited on page 28).