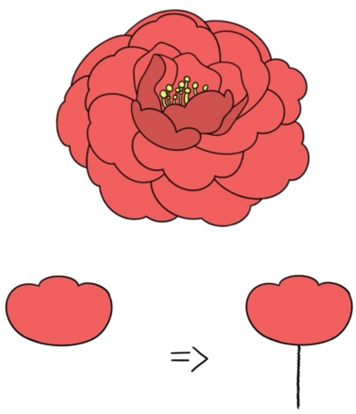
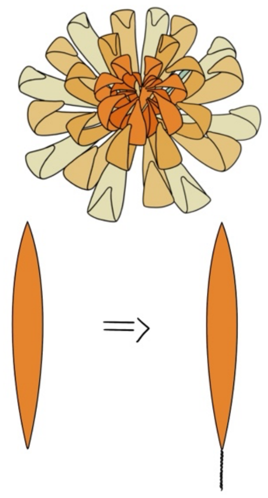
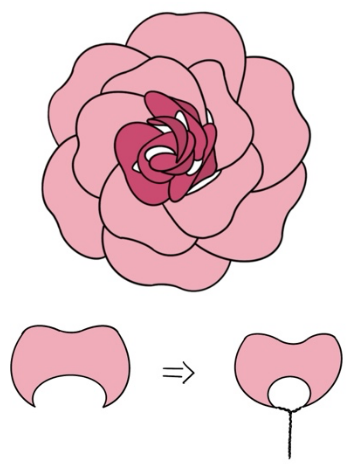
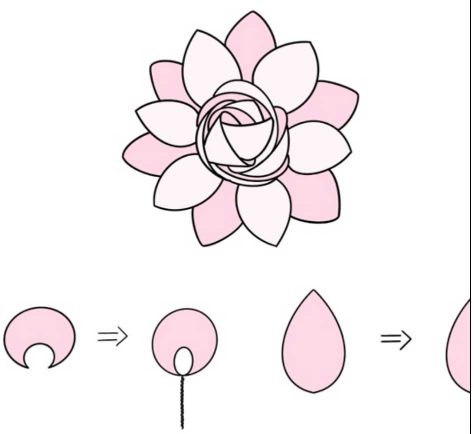
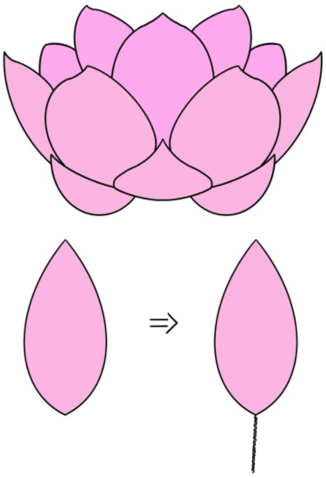
Design iteration and testing

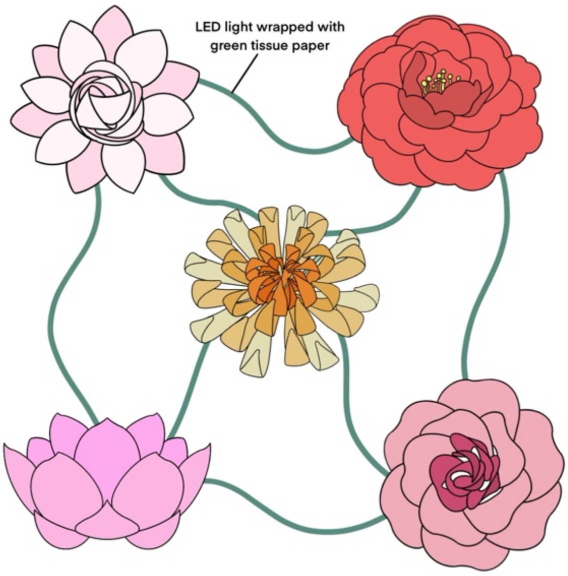
Structural Design for the Robotic Flowers:

Diagram	Flower Name	Characteristics	Reasons for Choice
	Peony	Layer 1 = 3 Layer 2 = 5 Layer 3 = 7	The stacked petals of the peony create a rich three-dimensional impression, making it an excellent choice for mechanical building design. Through a programmable mechanical system, it is possible to replicate a peony slowly flowering and responding dynamically to environmental changes, perfectly merging the beauty of nature with robotics.

	Chrysanthemum	Layer 1 = 5 Layer 2 = 6 Layer 3 = 7	Chrysanthemums feature a radial petal arrangement, which allows for a dynamic look with several layers. Mechanical devices can replicate the natural movements of petals unfolding layer by layer, slowly rotating, or gently swaying in the wind, increasing interactivity and viewing experience.
	Rose	Layer 1 = 3 Layer 2 = 5 Layer 3 = 7	The Chinese rose has the distinct feature of flowering once a month. It can endlessly bloom, wither, and bloom again. This process can be simulated using a mechanical device. For example, it can cause flowers to gradually close and then blossom again, showing the theme of vitality and cyclical

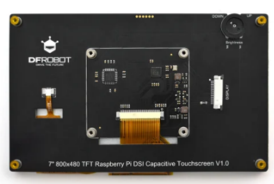
			rebirth. The mechanical structure enhances the characteristic of the Chinese rose's frequent blooming, resulting in an artistic piece with a stronger sensation of life.
	Gardenia (Cape Jasmine Flower)	Layer 1 = 3 Layer 2 = 3 Layer 3 = 6 Layer 4 = 6	Gardenia petals are organized uniformly and closely, with a structure that unfolds in a spiral pattern, making them ideal for opening and shutting mechanical flowers. Mechanical devices can be used to slowly rotate, unfold, or close the petals, replicating the process of gardenias from bud to full bloom, making it more vibrant and natural.

	Lotus	Layer 1 = 3 Layer 2 = 6 Layer 3 = 5	The petals of a lotus flower open sequentially from the outside in. A multi-layered mechanical construction can be utilized to manage the petals' slow blooming, replicating natural growth and giving it a more vibrant appearance. The lotus flower blooms gently in the early morning before going close up in the evening. This type of change can be performed with servo motors, giving the mechanical flower a bionic rhythm.

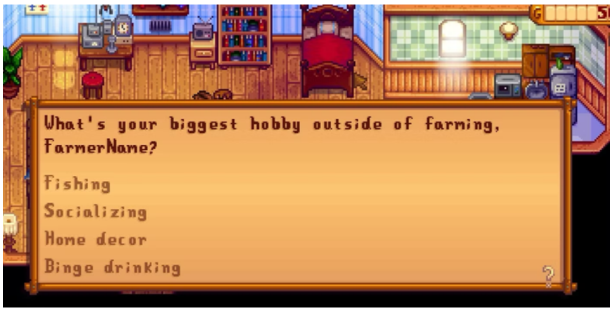


For our outcome we want to create a microcosm, simulating how the five flowers will react to different resource levels. We want to clearly show that the CMN regulates this through transporting signals and resources. Therefore, we want to add roots that go between the plants, and we will create these by embroidering them on water-soluble material. This will work effectively as we can create a wispy texture that is easy to manipulate. Additionally, this allows us to sew the LEDs into the roots rather than using full LED strips. To the side of the environment, there is a rectangle, this is representative of the sign. Through it, the user will be able to interact with the resources in the environment. Initially we wanted the users to directly interact with sensors so that the experience was pervasive, however this posed several health and safety concerns. Especially as water and electronics don't mix. Therefore, we will use a screen where the user can select which resource they want to manipulate. Then there will be physical objects, such as buttons or potentiometers, where the user can control the resource value.

Sign Design:

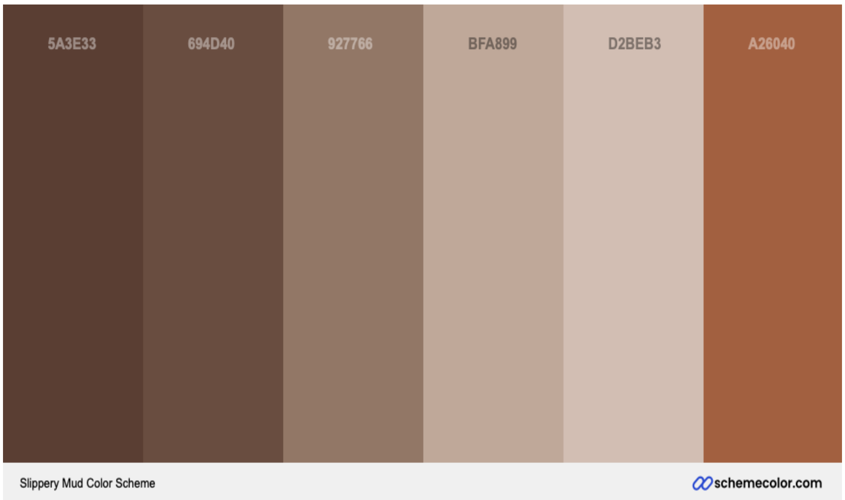


For the graphical user interface in the sign, we will be using a DFRobot 7" Raspberry Pi LCD touchscreen V1.0[1]. This means we will be coding the interface with Python, aside from the Scientific Computing with Python course we will be completing, we don’t have much prior experience using it to create interactive interfaces. Luckily, some libraries make creating the GUI easier to program. Such as Guizero[2] and Kyvi Python[3].



Example of a dialogue box in Stardew Valley[4] | The sign object in Minecraft[5]

Therefore, we will draw inspiration from the design styles of popular games such as Stardew Valley and Minecraft. As well as following a neutral colour palette, sticking with soft earth tones as seen in the image below.



[1] 7’’ TFT_Display_with_Touchscreen_V1.0_SKU_DFR0678-DFRobot (no date) Dfrobot.com. Available at: https://wiki.dfrobot.com/7%27%27%20TFT_Display_with_Touchscreen_V1.0_SKU_DFR0678 (Accessed: May 6, 2025).

[2] Shturma, S. (2020) 7 tools for GUI development on Raspberry Pi, Women Make. Available at: https://medium.com/women-make/7-tools-for-gui-development-on-raspberry-pi-67fd7cd9226e (Accessed: May 6, 2025).

[3] Kivy: Cross-platform Python framework for NUI (no date) Kivy.org. Available at: https://kivy.org (Accessed: May 12, 2025).

[4] Dillon, J.C. and Zalace, J. (2020) Stardew Valley: 15 story mods you should try, TheGamer. Available at: https://www.thegamer.com/stardew-valley-best-story-mods/ (Accessed: March 3, 2025).

[5] (No date j) Minecraft.wiki. Available at: https://minecraft.wiki/w/Sign (Accessed: March 3, 2025).

Slippery Mud Colour Scheme by schemecolour[6]

UI Theory with Gamification and Pixel Aesthetics in our Ecological Installation:

This interactive installation blends ecological theory, robotic technology, and UI principles to simulate plant communication networks inspired by the mycorrhizal system. Central to its design are UI theories that enhance engagement and deepen the user's emotional connection. Donald Norman's affordance theory[7] guides the creation of intuitive interaction points—such as sensors and tactile plant interfaces—while feedback and feedforward mechanisms ensure real-time, responsive changes through LEDs and movement. These interactions are visually expressed using a pixel aesthetic, reflecting both retro digital systems and the modular logic of biological networks.

The installation employs gamification to foster user engagement through cooperative challenges and resource-balancing tasks. For instance, participants must maintain plant health by manipulating environmental inputs like moisture or nutrients, which are visually tracked via animated LED pathways. As challenges escalate, flow theory is applied to keep users immersed through clear goals and adaptive difficulty.

Empathic design principles shape the emotional tone of the experience. If neglected, plants may dim or wilt, encouraging users to respond and care. This emotional feedback loop mirrors real ecological consequences and enhances user agency. Pixel-based visuals reinforce this dynamic, turning root systems into glowing data maps that reflect the health of the network.

Drawing from embodied and extended cognition theories, the installation allows users to externalise their influence on a living system, blurring the boundaries between human, machine, and nature. The tactile and visual interface design mimics tangible user interfaces (TUIs), emphasising physical interaction over screen-based input.

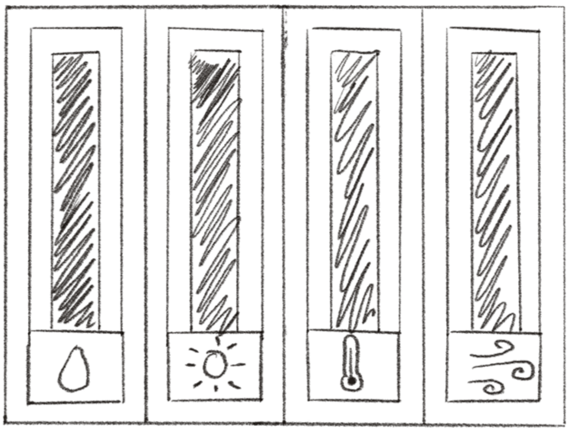
Wireframes:

https://www.figma.com/proto/b3TYG9PI3CrwJM41BB03CA/UI-for-Plant-Simulation?node-id=1-75&t=r86gIGLLbwRix7tp-1

Since making the first wireframe, we have had meeting No.8. We realised that the user won't be able to actively interact with the screen except by using potentiometers to interact with the elements. Therefore, instead of interacting with each element individually, they will all have to be on the same screen. Before, the design was too touch-oriented, forcing the user to interact more with the screen rather than the hardware. Additionally, as the user has to manually enter each resource screen, it could be confusing which potentiometer corresponds to which resource. If they are twisting the wrong potentiometer, then they won't be able to tell which resource is being manipulated, and they will have to enter each page individually.

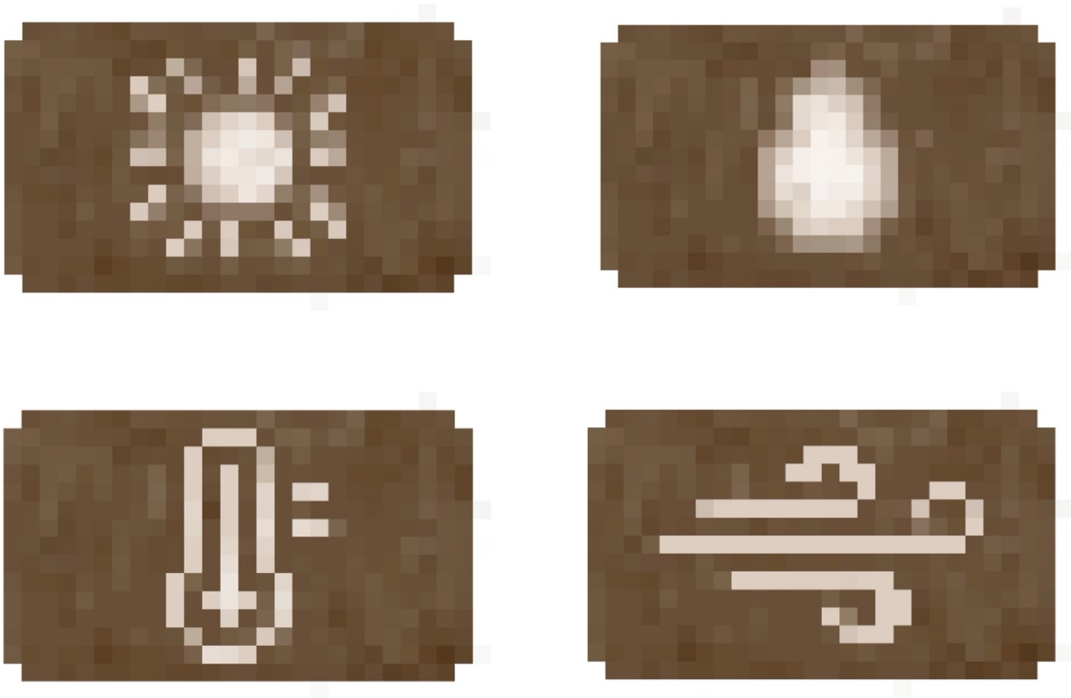
[6] Slippery Mud color scheme (no date) SchemeColor. Available at: https://www.schemecolor.com/slippery-mud.php (Accessed: April 28, 2025).

[7] Norman, D. A. (1988). The psychology of everyday things. Basic Books.



As seen in the new design above, the users will now be able to see all the resources simultaneously. It simplifies the interface and the user's interaction with it. It also creates a more scientific impression.

Button Designs:



Sun (Light) | Water Droplet (Moisture) | Thermometer (Temperature) | Wind (Air Flow)

Each button will open a new window where there will be information about the resource's impact on both the environment and how it relates to the common Mycorrhizal network. Initially, the buttons were just going to be labels indicating which progress bar relates to which resource. But I decided to add extra interactivity, enabling the user to learn about how everything interconnects and how they directly impact each plant.

Initial Prototypes:

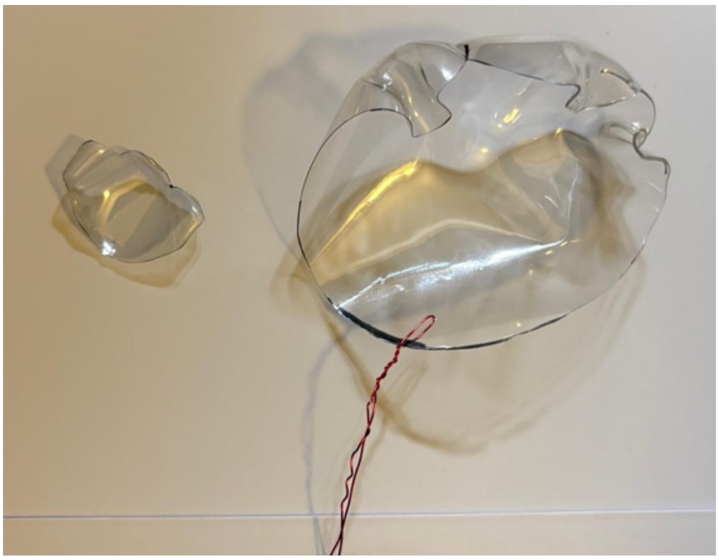
Kitchen Foil:

In the prototyping stage of our mechanical flower project, we experimented with various materials for crafting petals. One of the most promising outcomes came from using tin foil.

As shown in the image above, we cut out individual chrysanthemum-like petals and quickly assembled them into a flower formation. Tin foil proved to be a lightweight and malleable material, allowing us to easily shape and adjust the curvature of each petal. This flexibility is especially beneficial when integrating with servo motors, as it reduces the mechanical load and ensures smoother movement.

Compared to other materials we tested (like plastic bottles or origami paper), tin foil required less effort to manipulate and offered a balance between structural stability and sculptural expressiveness.

Plastic Bottles:

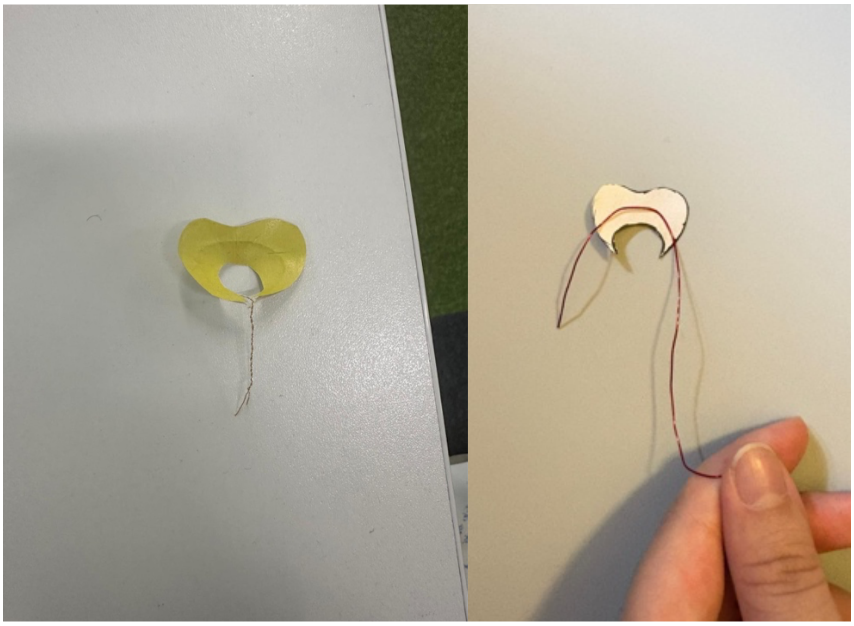


As part of our early prototyping process, we experimented with using cut plastic bottles as a petal material. The idea was to repurpose transparent PET plastic, cutting it into organic petal shapes and softening the edges with a lighter to achieve natural curves and textures that mimic real flowers.

In the image above, you can see one of our test pieces. While the plastic could indeed be reshaped when heated, the process was difficult to control, and the final form often did not match our intended aesthetic. The material tended to warp unpredictably and lacked the softness and expressiveness we were aiming for.

Additionally, the weight of the plastic became a serious issue. Even after attaching a supporting metal wire at the base, the connection was unstable, and we worried that the servo motors would not have enough torque to reliably move the petals. Given these limitations both in control and functionality, therefore we decided to abandon this material and continue exploring lighter, more flexible alternatives.

Paper:



Another material we explored during prototyping was origami paper. Its lightweight nature and wide availability in various colours and textures made it an attractive option for creating expressive and delicate petals. In the image above, we tested a basic shape using a single sheet of yellow origami paper with a support wire.

While origami paper offered great visual versatility and was easy for servos to move, the material posed some significant challenges. It lacked structure, making it hard to hold a desired shape unless we embedded metal wire along its edges or centre. Even then, shaping the petals required careful folding and reinforcement, which proved time-consuming and not easily repeatable for multiple flowers.

Additionally, origami paper is delicate and susceptible to tearing or collapsing over time, especially under repeated mechanical stress from servo movements.

For these reasons, despite its aesthetic potential and colour flexibility, we decided not to proceed with origami paper and instead focused on materials that provided better structural integrity and ease of production, such as reinforced tin foil.

Crepe Paper:

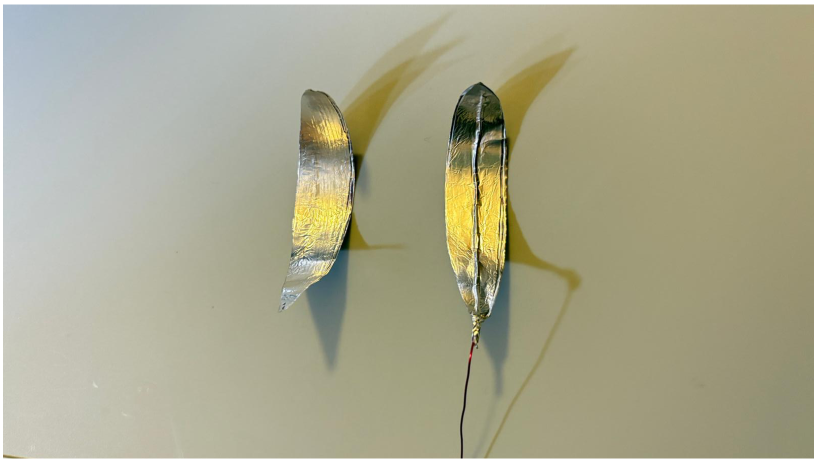
We also tested crepe paper as a potential material for our flower petals. Among all the materials we experimented with, crepe paper offered the most visually appealing texture and colour. Its delicate ridges and vibrant tones closely mimic the organic qualities of real flower petals, adding a natural softness to the form.

In the prototype shown below, we crafted a large petal shape and attached a support wire for shaping:

Crepe paper is relatively easy to sculpt and holds curved shapes well with minimal support, which made it quite enjoyable to work with. However, the major downside we encountered was its weight—especially when layering multiple sheets for structure or scale. This added bulk would strain the servo motors, potentially reducing the responsiveness and longevity of the mechanism.

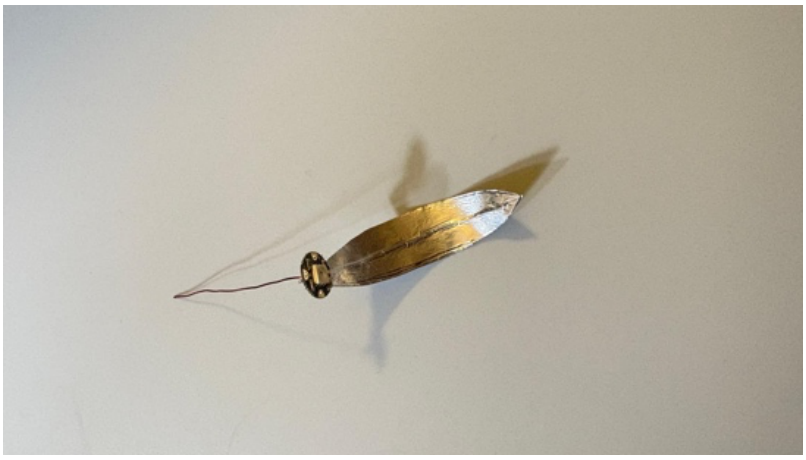
Although we appreciated its aesthetic potential, we ultimately chose not to use crepe paper in the final version due to functional concerns, prioritizing a balance between visual quality and mechanical feasibility.

Material Improvement: Reinforced Tin Foil Petals



After testing our initial prototype using single-layer tin foil petals, we noticed a critical issue: the petals were too fragile and easily deformed during handling or actuation by the servo motor. To solve this, we developed a reinforced version of the petal structure.

As shown in the image above, we folded the tin foil several times to increase its thickness and durability. Additionally, we embedded a thin metal wire through the centre of each petal. This wire acts as a skeleton, allowing us to shape and fix the petal into a desired form with much greater control.

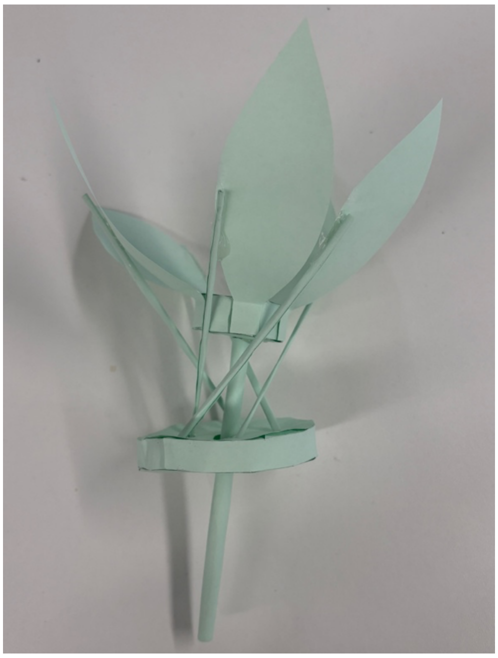


Following the structural reinforcement of our tin foil petals, we explored how to enhance the visual aesthetics and interactivity of the flower. One idea we developed was to integrate programmable LED lights into the centre of the flower.

As seen in the image above, the reflective surface of the silver foil offers a unique opportunity: it can amplify and scatter the coloured light emitted by LEDs. By programming the LEDs to display different colours in response to various inputs (e.g., water, wind, danger), the petals can reflect these hues across their curved surfaces, creating a glowing, immersive effect.

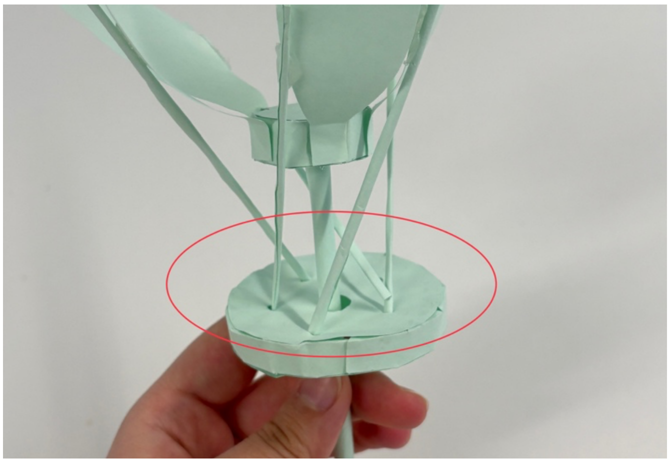
Mechanics:
Initial Paper prototype:

To explore the mechanical logic of flower opening and closing, I first created a simple paper model. This quick prototype helped me understand how rotational movement could translate into petal motion using simple materials.



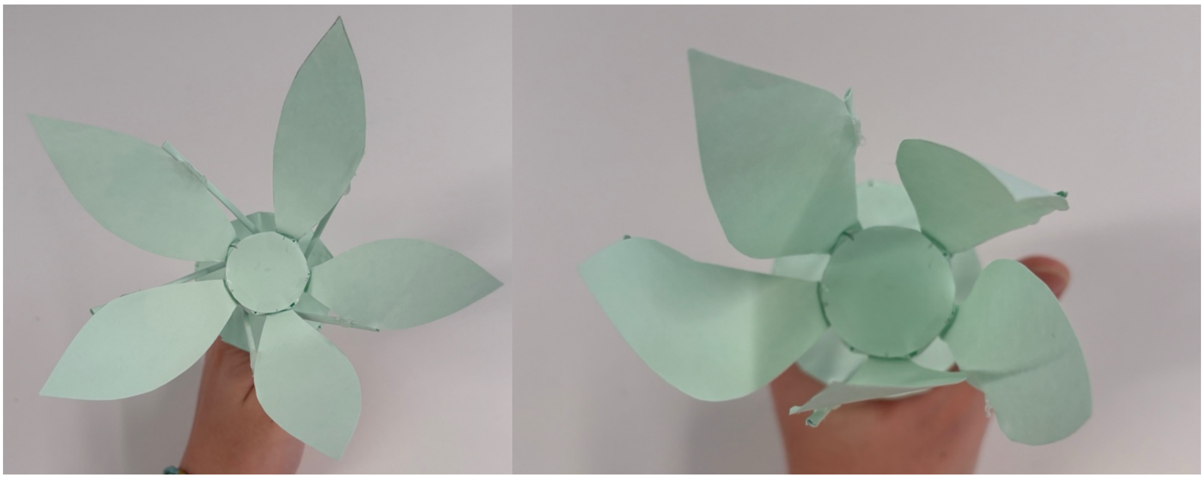
- I began by cutting out five paper petals, which were glued onto a flat paper cylinder that served as the flower's central hub.
- Separately, I made another, larger paper cylinder, and punched five holes evenly around its top edge.
- To simulate motion, I rolled thin paper strips into rods, creating makeshift paper linkages. Each rod was:
- Inserted into one of the holes on the cylinder — loose enough to pivot freely,
- Glued to the back of a petal at the other end.

This setup meant that when I rotated the outer cylinder (acting like a disk or cam), the rods would push or pull the petals, causing the flower to open or close.



The flexibility at the joint where the rods connect to the rotating disk was crucial. If that end was fixed too tightly, the rods would resist movement or bend. But by allowing the rods to pivot freely within the holes, the motion transferred smoothly and evenly, producing a clear and effective blooming motion.

The following images show the functional result of the initial paper-based petal mechanism in two distinct states:

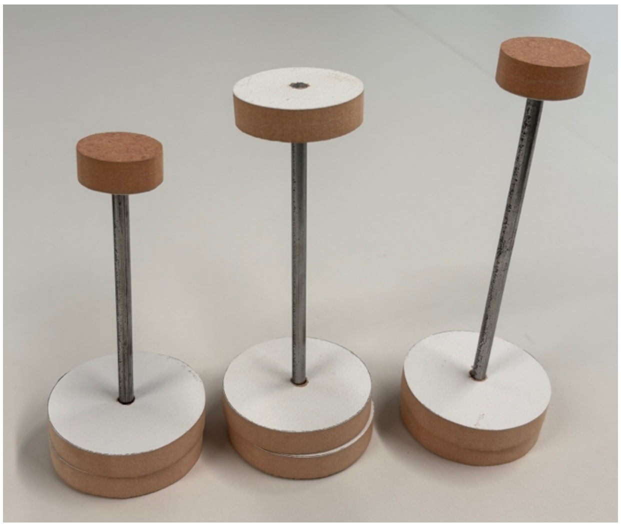


Left: Fully Open Position

The petals are extended outward, demonstrating the maximum opening angle achievable by rotating the disk. The paper rods, connected loosely through holes in the disk base, push the petals outward as the disk turns. This mimics the natural blooming action of a flower.

Right: Closed Position

In this position, the disk has rotated back, pulling the rods inward and allowing the petals to fold toward the centre. Because the rods are not rigidly fixed into the holes, they pivot naturally during the transition, enabling a smooth and symmetric closing motion.



As we moved toward final construction, we decided to replace the original paper disks with more durable materials for both the flower centre (top disk) and the disk used to control the petal rods (middle disk). To achieve this, we consulted the sculpture technician, who assisted us in precision-cutting several disks using templates and mounting them onto a central metal rod.

- **Top Disk (Flower Core):**

  A smaller disk mounted at the top of the rod, representing the central hub of the flower. This disk is **fixed** in place.

- **Middle Disk (Rod Controller):**

  This is the most crucial part,this is designed to **insert and hold the petal-controlling rods**. In the current build, this disk is **not yet fixed** and can **slide freely** along the metal rod.

- **Bottom Disk (Base Support):**

  The largest disk is mounted near the base to **anchor and stabilize** the entire structure.

Next Step: Material & Mounting Solution

We now need to explore ways to:

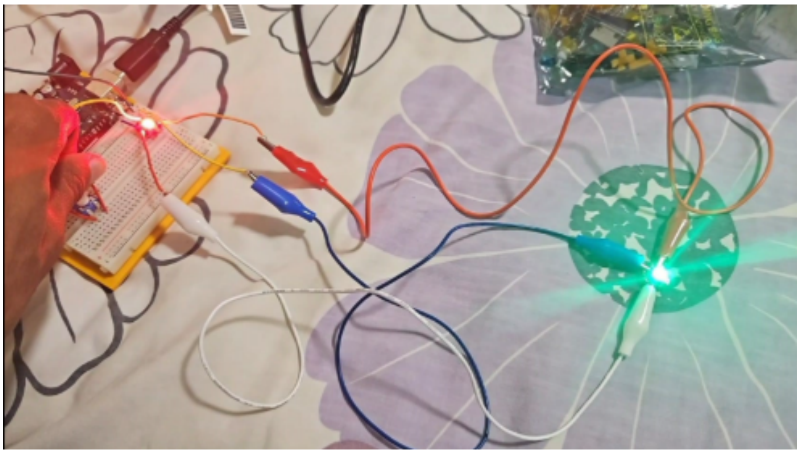
- Fix the middle disk securely to the rod, allowing it to rotate with the shaft when driven by a motor.
- Or, alternatively, replace the middle disk with a material that is easier to drill, mount, and lock onto the rod, such as:

  - Acrylic or laser-cut plywood with a central locking mechanism,
  - 3D-printed PLA with a press-fit or set-screw hole,
  - Foam board + insert nuts, for temporary testability.

This step is essential for achieving a stable and motor-compatible transmission system, ensuring that the flower's opening movement can be mechanically repeated and scaled.

In addition to rethinking the middle disk, we're also considering replacing the central metal rod. While it provides good structural support, it is relatively heavy, which could become an issue once the system is motor-driven. A lighter material, such as a wooden dowel or plastic rod, would reduce the load on the motor and make the entire mechanism more efficient and easier to work with during assembly.

Flora LED Prototyping



Link to access video:
https://drive.google.com/drive/folders/14wHf8tIFDz5ofBWvF16JLvhIsOVXaJpZ?usp=drive_link

Components Included:

- 1 potentiometer (for analog input)
- 2 Adafruit NeoPixel LEDs
- Arduino Uno microcontroller
- Shared breadboard and jumper wires

Prototyping with Flora NeoPixels during the Easter break was an exciting and occasionally frustrating dive into physical computing. My objective was simple: to control a pair of colorful LEDs using a single potentiometer such that they're sewed together to create the 'trail -like' effect when the resources arer being transferred to the flowers by the user. However, as with most hardware projects, the wiring turned out to be slightly more intricate than expected.
One of the first challenges I encountered was related to the NeoPixels' power and signal stability. Initially, I connected everything directly through the breadboard, assuming a basic VCC-GND-Signal setup would be sufficient. However, the LEDs either didn't light up at all or flickered erratically. It took some debugging and rechecking the Adafruit documentation only to realize that I hadn't grounded the circuit properly. I had forgotten that NeoPixels are sensitive to common grounding; the Arduino and the LEDs needed to share the same GND line or the signal wouldn't register correctly. Once I fixed that, the flickering stopped.
Another hurdle was working with the data input line for the LEDs. NeoPixels require a precise digital signal, and even slightly loose jumper wires caused unpredictable behavior. After a few rounds of swapping wires and tightening connections, I proved to come with a simple wiring that worked and set up the project to be successful in wiring more LEDs. This gave me more consistent performance, and for the first time, both LEDs lit up as expected in response to the potentiometer. Additionally, my wiring and conduction issues were trivial as they're sewable

and are intended to a conductive thread that will assure that the voltages reach each NeoPixel.

The code uses the Adafruit_NeoPixel library to control the LEDs. It reads the analog value from the potentiometer, maps it to the number of LEDs to illuminate, and updates the LED colors accordingly. A short delay is included to reduce flicker.

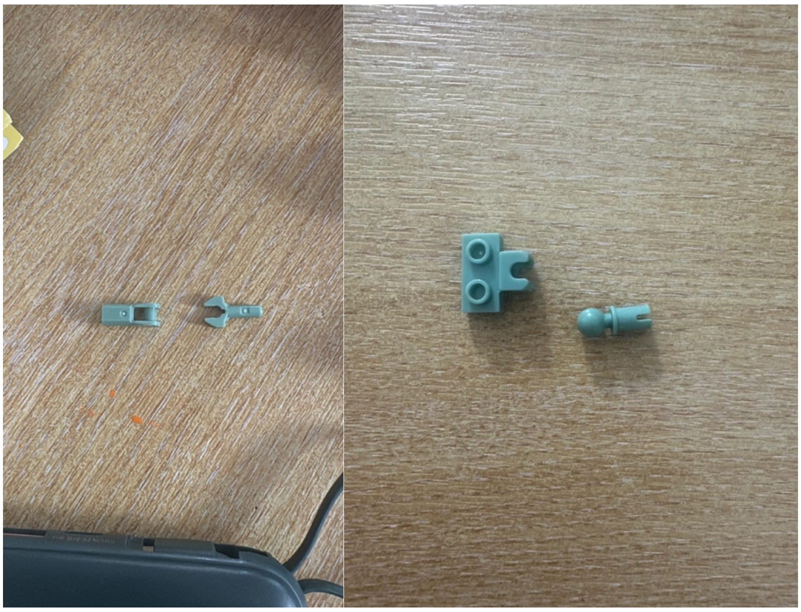
The potentiometer itself introduced a few subtleties. While the wiring was simple — three pins connected to GND, 5V, and analog input A0 and interpreting the signal in the Arduino sketch required some tuning. The raw analogRead values ranged from 0 to 1023, and mapping them directly to control two LEDs meant that I had to carefully calibrate the thresholds. Too narrow, and the LEDs would flicker on and off with the slightest touch. Too wide, and the response felt delayed or inaccurate. I eventually found a balance using map() to convert the analog range into discrete control for 0, 1, or 2 LEDs. The final result was a simple yet satisfying control scheme: turning the knob gradually brought one LED to life in red, then added a second one glowing green, with interchangeable lights per our creative direction with resource colors.

Current Issue: Middle Disk Movement

While this layered assembly provides a solid physical foundation, we identified a key issue:

If the system is to be motor-driven, the middle disk must rotate — but since it currently slides freely, it cannot transmit rotational motion effectively. This undermines the core mechanic that controls the petal movement.

Lego Samples



In the original paper prototype, the middle disk was drilled with holes to allow manual insertion of control rods, which pushed and pulled the petals during rotation. While this worked well for initial testing, it lacked mechanical precision and long-term durability.

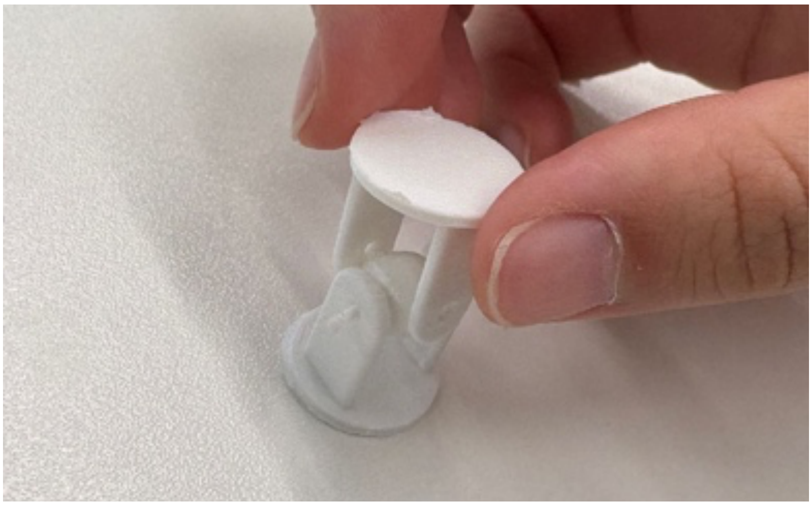
To improve this, I began exploring more flexible and modular connection methods. I discovered that certain LEGO Technic joints, specifically ball-and-socket or clip-based connectors, could serve as ideal replacements.

As shown in the images, these connectors allow:

- Smooth rotational freedom,
- Sturdy yet detachable connections, and
- Multi-angle movement, which could absorb slight misalignments and reduce tension on the petals.

Integrating these components into the middle control disk would eliminate the need for tightly aligned holes and allow the flower structure to respond more organically to motion. This also opens up future possibilities for more modular flower assemblies, where rods or petals can be easily removed, replaced, or reconfigured.

3d printing:



To further improve the flexibility and modularity of our flower control system, we explored using a universal joint mechanism as the linkage point between the central disk and the flower's control rods. After searching for suitable models online, we selected a design and 3D-printed a prototype.

Structure

The printed joint consists of four components:

- Two outer "socket-like" parts, each with two small holes for fixing;
- A central ball connector with four receiving holes on its surface;
- Small locking pins, meant to be inserted through the aligned holes to hold the joint together.

This structure, in theory, would allow rotational freedom on two axes, providing the flexibility needed to adapt to flower petal movement.

Challenges Encountered

- Due to the small size of the holes and pins, the parts did not fit together properly after 3D printing, the tolerances were too tight.
- The locking pins couldn't be inserted without force, which risked damaging the fragile parts.
- Even if functional, the manual assembly process for each joint proved too time-consuming and impractical for scaling to multiple flowers.

Conclusion

While the concept was promising, this particular model wasn't suitable for our application in its current form. Moving forward, we may:

- Redesign the joint to have larger tolerances or snap-fit features,
- Use Lego Technic joints or off-the-shelf ball-and-socket connectors,
- Or explore simpler pivot joints that offer enough flexibility without complex assembly.

After the challenges faced with the small pin-based joint, we explored a second 3D-printed universal joint model, this time using a screw-locking mechanism for greater stability and ease of assembly.
The design features:

- A hemispherical base that serves as a socket, into which a ball-shaped joint is inserted;
- A threaded top cap, functioning like a screw, that tightens over the ball to hold it securely in place;
- The ball remains fully rotatable within the socket, offering flexible motion while still being locked into the structure.

This setup allowed us to simulate a pivoting point that is both secure and adjustable, making it ideal for applications like controlling flower petal rods, where multidirectional freedom is needed without sacrificing structural integrity.

Why This Worked Better

- No small fragile parts to insert manually;
- Easy to assemble and disassemble;
- Provides controlled rotational freedom, similar to a real ball-and-socket joint.

After testing both designs, we ultimately decided to use the second universal joint model with the screw-locking mechanism. Compared to the first version, this design offered significantly better ease of assembly, structural stability, and rotational flexibility. All of which are crucial for supporting the movement of the flower petals in our system.

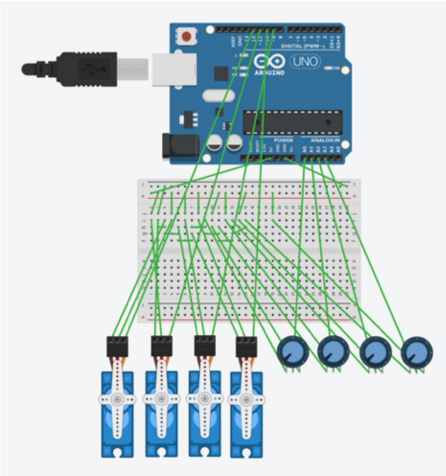
Circuit Designs:

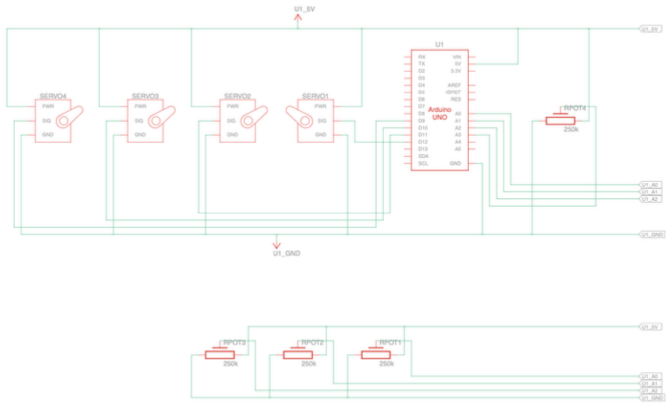
Circuit Design: Servo Motor and Potentiometer Control

This diagram illustrates the core wiring setup for our mechanical flower system, which includes:

- 4 potentiometers (used for manual input)

- 4 servo motors (each driving the opening/closing of one flower)
- All components connected to an Arduino Uno via a shared breadboard





Wiring Breakdown:

Potentiometers (x4):

- Each potentiometer has three pins:
  - Left pin (GND) → connected to Arduino GND via breadboard
  - Right pin (5V) → connected to Arduino 5V via breadboard
  - Middle pin (Signal) → connected to Arduino analog input pins A0 to A3
- Function: Each potentiometer outputs an analog signal (0–1023), which is mapped in the code to a servo angle (0–180°)

Servo Motors (x4):

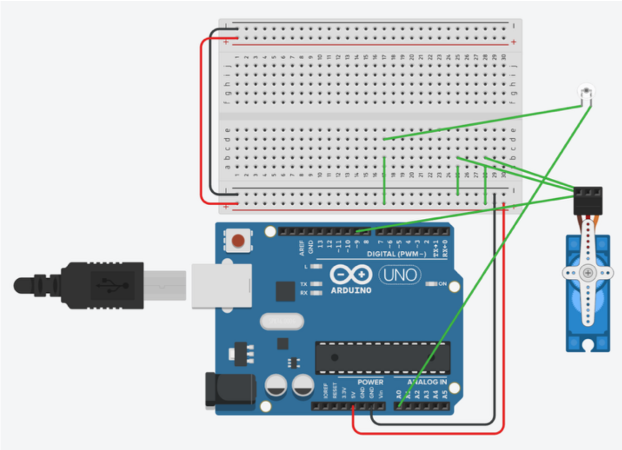
- Each servo has three wires:
  - Brown or Black (GND) → connected to GND rail on the breadboard
  - Red (Power) → connected to the 5V rail (Arduino or external supply if needed)
  - Yellow/White (Signal) → connected to digital PWM pins D9 to D12
- Function: Each motor receives a PWM signal from the Arduino and rotates accordingly to move the flower petals

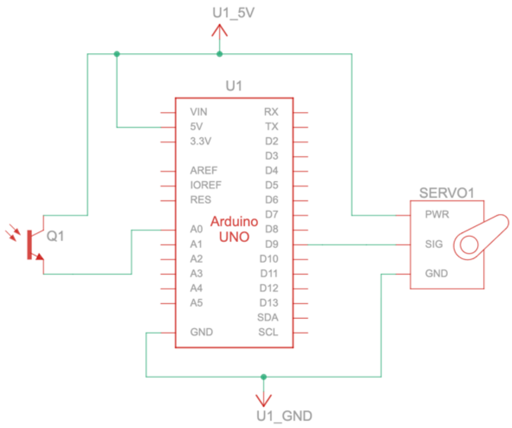
Power and Breadboard Layout:

- A common 5V and GND rail distributes power to all potentiometers and servos.
- Wires are carefully organized to minimize interference and ensure accurate signal transmission.

Circuit Design: Light Sensor and Servo Motor Control

This circuit diagram shows how a Grove Light Sensor is used to control a servo motor via an Arduino Uno. The goal is to make the motor (e.g. controlling a flower) respond to ambient light, opening or closing based on light intensity.





Wiring Breakdown:

- Grove Light Sensor:
  - Connected to analog input pin A0 on the Arduino.
  - The sensor outputs an analog voltage (0–1023) depending on light level.
  - VCC and GND of the sensor are connected to the 5V and GND rails on the breadboard.
- Servo Motor:
  - Signal wire connected to digital PWM pin D9.
  - Power (red) and ground (brown/black) wires are connected to the 5V and GND rails via the breadboard.
  - This allows the motor to rotate in response to the light level input, based on mapped servo angles in code.

Power & Breadboard Setup

- The Arduino's 5V and GND pins are extended to the breadboard rails to power both the light sensor and the servo.
- All component connections are made via the breadboard, making it easy to test and prototype.

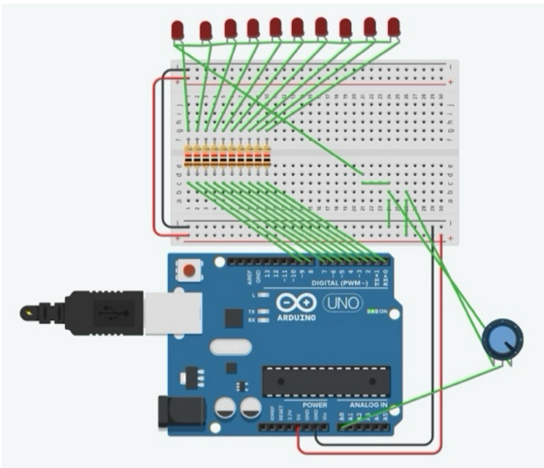
How It Works

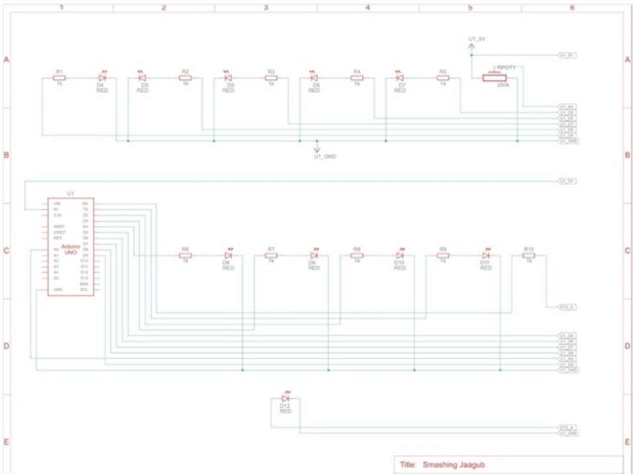
- When the light intensity increases, the sensor reading goes up.
- In code, this reading is mapped to a servo angle (e.g., 0–180°).

- This allows the flower to open or close depending on brightness — for example, opening in the dark and closing in bright light, depending on how the logic is defined.

Circuit Design: LED Visualization with Potentiometer Control

This circuit demonstrates how sewable LEDs (represented here by standard LEDs) can be used to simulate root signal transmission through progressive lighting, controlled by a single potentiometer.





Wiring Breakdown:

- Potentiometer:
    - Connected to analog input A0.
    - Acts as a manual control to simulate changing “signal intensity” or environmental input.
    - Center pin connects to A0, side pins to 5V and GND.

- LEDs (x10):
    - Connected to digital pins D2–D11 on the Arduino.
    - Each LED is connected in series with a current-limiting resistor to prevent burnout.
    - Anode (long leg) connects to digital pin, cathode (short leg) connects through a resistor to GND rail.

- Resistors:
    - Each LED has its own resistor (typically 220Ω–330Ω), placed on the same row to form parallel paths.

How It Works:

- As the potentiometer is rotated, the Arduino reads the analog value from A0 (0–1023).
- This value is then mapped to a range from 0 to 10, determining how many LEDs should light up.
- As the input increases, more LEDs light up sequentially, creating a “growing” light pattern.
- When turned in the opposite direction, LEDs turn off in reverse order, simulating the signal retreating.

This visual progression mimics mycelium-like root signal transmission, helping to reinforce the concept of information traveling through a connected system of flowers or plants.

Power Notes:

- All components share common 5V and GND lines from the Arduino via the breadboard rails.

User Testing:

After our final Presentation on the 12th of May 2025, we presented each student group and our Lecturer with a series of questions to help us gauge the reaction of the users to our outcome. Gaining an understanding of how well they could interact with it and what they thought we could improve on. In Hindsight, it would have been beneficial if we had done preliminary testing. Then we could have got feedback earlier on and had enough time to implement these changes, whereas doing the testing at this time has left us only 10 days before the deadline.

Questions	User Responses
How intuitive was the interaction—did you understand how to send a signal and interpret the system's response?	‘It was easy to see how our interaction would trigger something in the network.’
	‘I found it very intuitive as the interactivity aspect of this group’s project was the trait that I admired so much. The arrangement, coordination and functionality voiced about the system in the group presentation was extremely clear and it made it easy to understand.’
	‘The interactions were extremely interactive as moving the motors using the potentiometers. Additionally, the interface design of the screen was easy to navigate. I understood how to use each of the input devices used to evoke a reaction with the output devices. The systems responses seemed to be the flower moving in response and the LED lights between the flowers.’
	‘The screen-based interaction was incredibly intuitive and aesthetically

	pleasing. I was particularly fond of the Minecraft adjacent UX inspiration’
What did the visual and sensory feedback (LEDs, movement, etc.) make you feel or think about?	‘The LEDs were quite calm and made me think about water flowing, because of the colour and the sliding effect.’
	‘The visual aspect of the project (the beautiful flower garden) emulated multifaceted emotions of nostalgia of parks and the dystopian undertone of the extinction of flowers and the importance of their preservation and archiving’
	‘The height of the lotus felt appropriate to me within the concept of where lotuses are kept in comparison to the land-grown flowers. Additionally, the use of LEDs between the two flowers were integral to my understanding of the connectivity between flowers as roots. The movement of the flowers made me think of the movement flowers had in the wind.’
	‘The visual & sensory feedback was certainly evaluative of plants and flowers opening up- albeit at a faster and, perhaps, hyperbolic rate’
Did the simulation help you understand or imagine how mycorrhizal communication works in nature?	‘Definitely. Although I have a background in this already and I just loved seeing your work speak to these themes. Your simulation touched on so many interrelationships that made one of the most comprehensive ones I’ve seen!’
	‘Yes, it helped me understand the system and the nature of communication between plants.’
	‘The simulation allowed me to understand how plants (mainly flowers) are interconnected.’
	‘The simulation certainly helped to understand how mycelium communication works by linking it to the technological concepts that contemporary audiences maybe more familiar with.’
In what ways did this experience change or influence your perception of the connection between nature and technology?	‘It made a stronger connection in terms of how technologies like sensors, even though so tiny, can have a great impact and output in terms of scale.’
	The synergy of nature and technology amazed me: the paradoxical instance of man-made vs natural resources and how they can be beneficial to each other

	‘This experience allowed me to view the integration of robotics and nature as something that doesn’t require for the exact mimicry of flowers in nature for in robotics. It made me think that for nature and technology to be connected they could be combined without giving the impression that it is more from one field than the other’
	‘The experience prompted me to consider the similarity between technology and nature. Showing how nature and technology can be used not only in conjunction with one another but to serve and facilitate each other.’
What features would you like to see added or improved to make the system more engaging or meaningful?	‘I’d love to see the screen interaction system with the water, temp changing as we modify in the screen.’
	‘Perhaps in the future when the project becomes wide scale and they include different species of plants, they could add name tags for those, so it is easier to identify them.’
	‘I would have liked to see more LED connections between the flowers, but I understand this even due to time constraints.’
	‘Perhaps in the future, the flowers could be made to more closely resemble their real-life counter parts. Aside from that, the project was considerably exhaustive, and I look forward to seeing it on display in the Science Musuem!’

Feedback Session:

Group project formative feedback

Yadira Sanchez Benitez <Y.Sanchez-Benitez@soton.ac.uk> Friday 28 March 2025 at 16:48

To: Omar Yussuf (ay2u24); Jocelyn Eddy (jje1g24); Qingzhou Cai (qc3g24)

Hello everyone,

Your team project is on the right track. It is very exciting to see how you are bringing your inspirations into the final output. I specially encourage that you keep track of the artistic and scientific references as well. You already mentioned some, well done. There was a plenty of great information and context you provided in the presentation which showcases your team efforts. Keep this up! Your prototypes and designs are both creative and technically sound.

Please make sure that you upload all your different changes, team updates, obviously your code to the GitHub repository we created on Wednesday. Try and push code regularly, rather than all in one go. The brilliantness of GitHub and having projects there is that you can showcase you worked on the project often and consistently.

Let me know if you have any further questions at this point and feel free to contact me with any questions during the next few weeks, I will be around campus often from the 9th of April onwards.

Take care,
Yadira

Our first official feedback session occurred after our formative assessment in week 7, where we presented our current progress. At this stage, we have completed our designs and started prototyping. Whilst we have an ample number of inspirations, we could benefit from incorporating more artistic ones. Such as the work of Tommy Mitchell; a metal sculpture artist. It was beneficial to know that we were on the right track, it enabled us to move forward and spend more time on development rather than having to go back and improve our previous work. One of our main focuses now should be on learning how to use GitHub and integrating our work into it, gaining an understanding of how to clone repositories, make push and pull requests. It is also essential that we organise it into sections