# **The Water Minder**

Text

Description automatically generated

By Arrnavv Chawla and Varsha Sundar

**1. Idea**

Plants are extremely essential to human life and without them the earth would be inhabitable. Our goal was to create a project that, on a small and achievable scale, helps increase or achieve sustained greenery. Water is the most crucial element a plant needs to survive, and so we decided that our project should revolve around plants and their need for water.

**2. Problem Statement**

Whether someone lives in a huge mansion or a business flat, potted plants are always customary. Both our mothers adore sheltering lots of potted plants in our homes. And everyone in the family forgets to water them on a quotidian basis. Thus, we decided to create a device which can remind us to water plants every day.

**3. SDG in play**



**4. Constraints**

* The device had to be waterproof. All circuitries should be enclosed.
* It had to be portable, so that it would be easy to stick into any potted plant.
* It had to be sturdy and durable, to withstand pressures from watering of the plant.
* It should be aesthetically appealing as an ugly circuit inside a plant is never appreciated.
* It must be able to stand in the soil without support.

**5. Solutions Brainstorming**

We produced many ideas such as using certain thresholds to determine the amount of water the plant would need, based on the surrounding’s temperature and humidity. Creating solar powered self-watering mechanisms and creating an app that will gamify the process of watering plants. But in the end, we decided on a device that judges whether a plant needs water or not from the basis of its soil moisture readings. Initially we had decided on a prototype with a buzzer generating different sounds to signal as to whether the plant is thirsty or not. But after talking to mentors like Naveen from Phy Farms and Anirudh Shukla we gained multifarious user insights and then finalized on a more complete method of communication between the user and the plant – Messaging and LEDs.

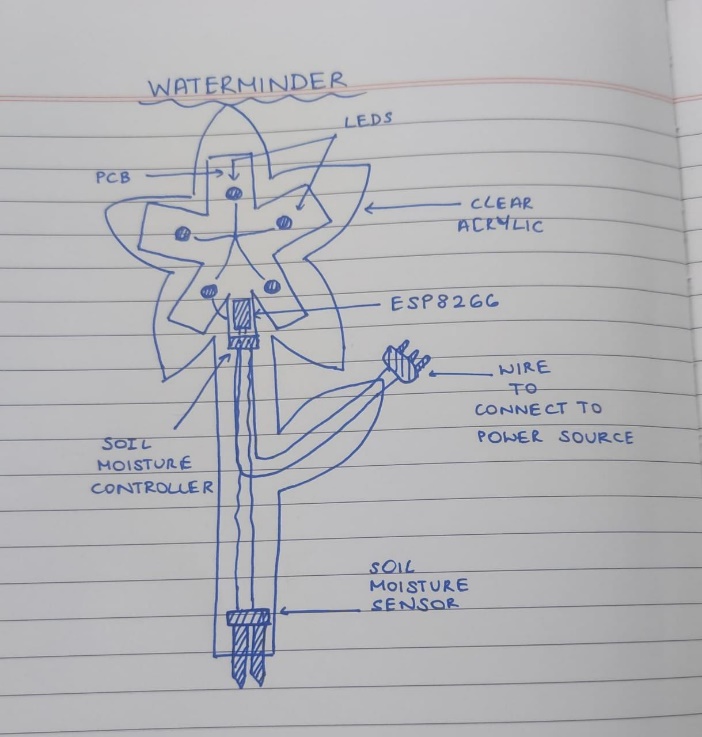
**6. Final Solution**

* **Name of the project**

**The WaterMinder**

**As a compound word from Reminder and Water, as it reminds you to water your plants.**

* **Final Detailed sketches**

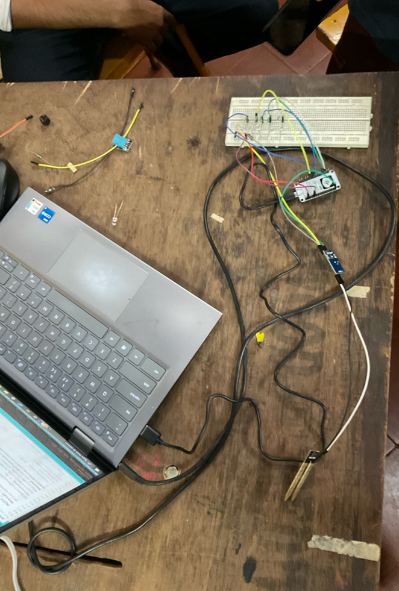


* **Paper Prototyping images and videos**









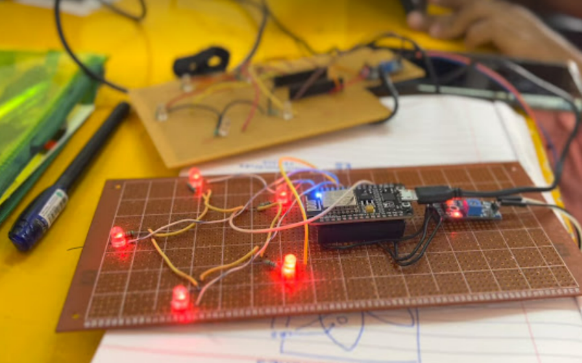
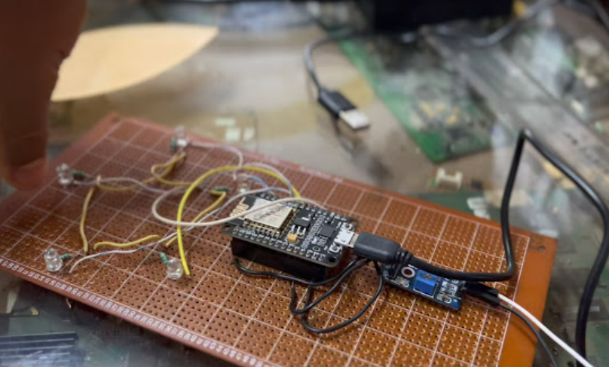


* **Final Prototyping images and videos**

****







* **Design process**

The device would have the shape of a flower. Which looks attractive as well as shelters controllers and circuitries inside the petal structures. The soil moisture sensor would be placed at the bottom tip of the stem which would then go into the soil. A leaf attached to the stem would act as a cover and outlet for the power cable. This flower casing was made from 5 mm transparent acrylic cut in the laser cutter. Two pieces were then screwed together to form a 3-D case.

* **Programming**

**ESP8266 CODE**

All versions of our codes through the process of prototyping have been attached in the Source Files section.

**ADAFRUIT FEED**

****

**IFTTT APPLET**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

* **Materials and machines used**

Materials

* Bread board
* Buzzer
* Arduino uno
* PCB Zero Board
* ESP8266
* Red Blinking LEDs
* Clear acrylic
* Soil moisture controller
* Soil moisture sensor
* Female to female jumper wires
* Potted plant
* Solder

Tools

* Wire clippers
* Scissors

Machines

* Solder iron
* Laser cutter
* Drill Machines
* **Skills used**

Throughout the project we were able utilise various skills we learnt over the course of the past year in our online as well offline skill building sessions. When we were creating the first prototype of the project, we used the Arduino UNO to connect various devices and we used the circuit making skills we had learnt in the Electronics and Robotics class. Along with that, we also had to code and make modifications to various libraries used in the program to code the sensor to give us the required output. We used our IoT skills to connect using Adafruit.IO and IFTTT to allow ESP8266 to broadcast through telegram. We also got a head start on our PCB course as we learnt how to use PCB Zero and soldering to make a compact, breadboard-less, version of our device. While designing the flower-based outer casing for our device, all the CAD lessons we had taken came handy. We made the 3-D outcome of our casing before laser cutting it to avoid any mistakes. While putting together the final project we also got to use power tools such as the driller to hold it in place. To sum up, all the skill building courses helped us create this final prototype in a way that would have been impossible to even think of before. The mentors guided us throughout the way and made sure that we had their support throughout this fun making process.

* **Challenges faced and solutions**

Facing challenges had become part of our daily routine during the making of this project. Since a huge part of our project depended on the soil moisture sensor, we first started working on coding and programming for it, and already we had met our first hurdle. We had failed to understand the soil moisture sensor. It worked on the principle that for zero moisture it gives a reading of maximum value of 10-bit ADC, that is 1023. Hence, we were very confused when our dry soil gave reading of 1000+. After doing research we realised the blunder that we made.

The next issue we faced was deciding what way we wanted to communicate with the user that their soil moisture was low. We started with using a buzzer that produced a loud annoying repetitive noise to draw their attention. We soon realised this would be regressive as it would annoy the user. And then, we decided to use a technological solution – sending messages via an ESP8266.

Once we jumped to the idea, we first decided to send messaged through WhatsApp only to discover that WhatsApp API was difficult to find. We even tried sending the alerts to the user through notifications using Blynk IoT. But after talking to the mentors, we realised we could stick to our first idea of sending messages by simply changing the platform to telegram as its API was easily accessible through IFTTT. And thus, we came to our final project.

* **Testing images**



A picture containing text

Description automatically generated

A person and person holding flowers

Description automatically generated with low confidence

The Makers: Varsha Sundar and Arrnavv Chawla

* **Reference links used through prototyping**

<https://www.programmingelectronics.com/an-easy-way-to-make-noise-with-arduino-using-tone/>

<https://www.programmingelectronics.com/an-easy-way-to-make-noise-with-arduino-using-tone/>

<https://tutorial.cytron.io/2020/01/15/send-sensors-data-to-adafruit-io-using-esp32/>

<https://io.adafruit.com/api/docs/?cpp#including-an-adafruit-io-key>

<https://t.me/ifttt?start=o_PEfLvWOQqmO3dz8QxBv0tQ>

<https://randomnerdtutorials.com/how-to-install-esp8266-board-arduino-ide/>

<https://www.instructables.com/How-to-Connect-Soil-Moisture-Sensor-and-ESP8266-to/>