Soil Moisture Indicator

using

Applied Technologies

Problem that needed to be solved:

House plants need frequent watering. With the busyness of daily life, it can be easy to forget to water your plants, and not notice them drying up! If the plant had a system that could indicate its needs, it would help us remember to water and keep plants healthy and alive.

Ideas for Solutions 1:

Idea 1- Gravity powered water device

We thought of using a water reservoir that is taller than the plant to feed water into the plants soil. The water flow would be controlled by a valve, operated by a person. We realized that this would still require as much attention and interaction as normal watering, and could still lead to forgetting. This idea didn't get past the drawing phase.

Ideas for Solutions 2a:

Idea 2- Air pressure powered drip water device

We thought of using a sealed jar to deliver water to the plants soil. The jar would be sealed with silicone and filled with water most of the way with a tube coming out of the lid. The jar would be turned upside down so the tube is at the bottom and the air is at the top. Heating the jar would increase the air pressure and push water out through the tube and to the plant.

We used a reptile heating lamp to heat the jar, and it appeared to work, but over multiple tests we found that the water would flow regardless of the heat and the device continued to deliver water freely. This design lacked a controlled method of delivering water and seemed to have too many variables for a realistic and reliable way to water plants. This idea had a prototype but we moved on from it.

Ideas for Solutions 2b:

Idea 2- Air pressure powered drip water device

Ideas for Solutions 3:

Idea 3- Microcontroller powered moisture sensor with leds

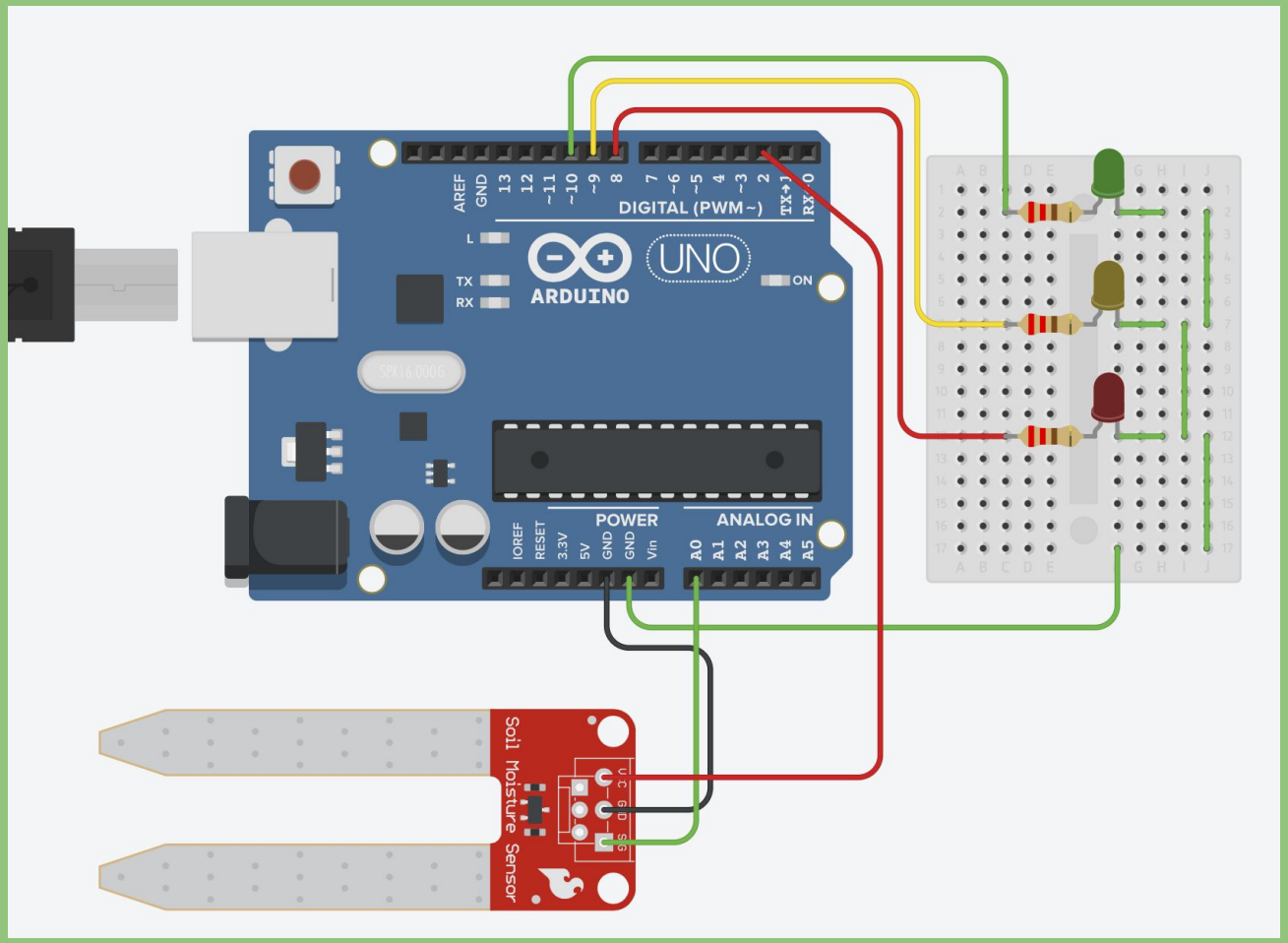
This idea came from wanting only a reminder to water, rather than more interaction and maintenance than normal watering. We realized that a soil moisture sensor paired with a microcontroller could monitor the soil moisture, and help remind us that the plant needs water with some indicator. With many options available for indicators, we chose LEDs as a visual aid to show moisture levels.

Product Design:

The design is based off of the use of a microcontroller and a soil moisture sensor. The sensor is wired to the microcontroller via three wires: one for power that will turn the sensor on, one for ground for the sensor and one for the reading the sensor takes of the moisture level. The microcontroller needs to be programed via coding so it has instructions on what to do with the sensors reading. We used TinkerCad code blocks for the instructions.

The microcontroller will use the reading, and then turn on and off LED lights depending on the range it falls into. A red LED for low moisture/dry, a yellow LED for medium moisture, and a green LED for moist/watered.

Product Design continued, TinkerCad prototype:



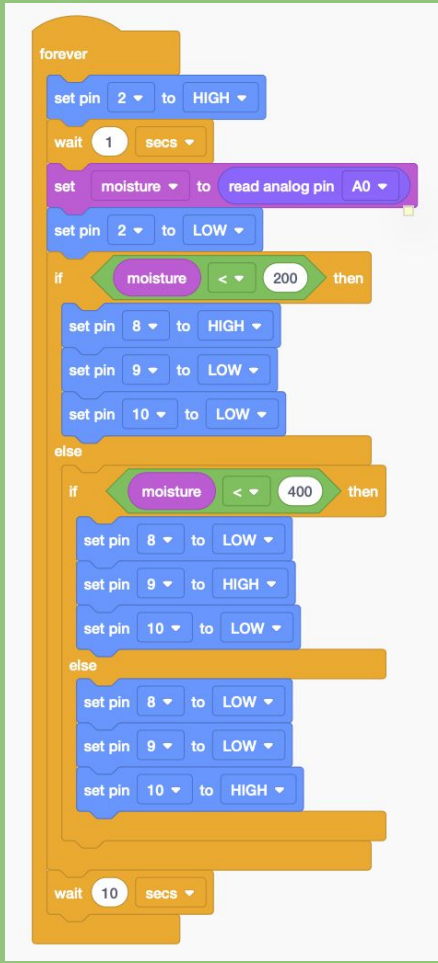
Product Design continued:

The code tells the microcontroller to turn the sensor on, take a reading and send it to the input pin on the microcontroller. It turns the sensor off and then tells the microcontroller to turn on and off the correct LED lights based on the reading and which range it's in. For demonstration purposes, the code repeats every 10 seconds, but can be changed to read and adjust the leds after any amount of time.

The powering on and off of the sensor is a design choice because the sensor is a resistive type, and would corrode and stop working over time as the electricity and moisture would degrade the sensors probes.

The microcontroller is powered by four AA batteries.

Product Design continued, code and parts list:



Name	Quantity	Component
U1	1	Arduino Uno R3
D3	1	Red LED
D2	1	Yellow LED
D1	1	Green LED
SEN1	1	Soil Moisture Sensor
R1 R2 R3	3	220 Ω Resistor

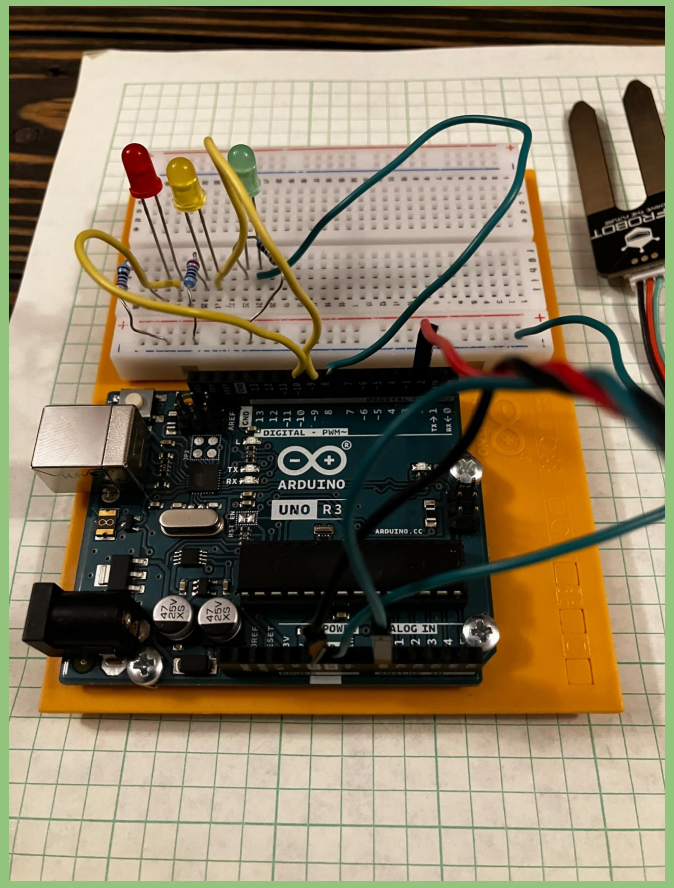
Product Test & Analysis:

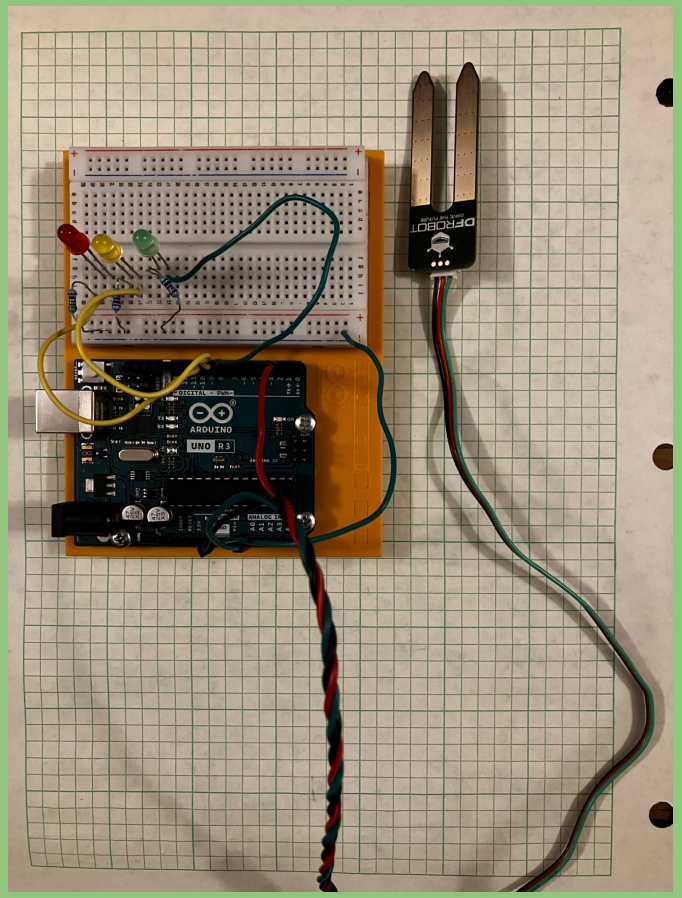
We ran multiple tests. First we tested the soil of a plant that had been recently watered, this test proved that the sensor and green LED were functional. Next we checked a plant with soil that was barely damp to the touch. The results were exactly as expected, the yellow LED turned on telling us that the plant was at medium moisture.

Finally, we tested a succulent that was dry which turned on the red LED and proved that all 3 settings were functioning as expected. We then spent time checking various plants throughout the house getting various results and adding water as needed.

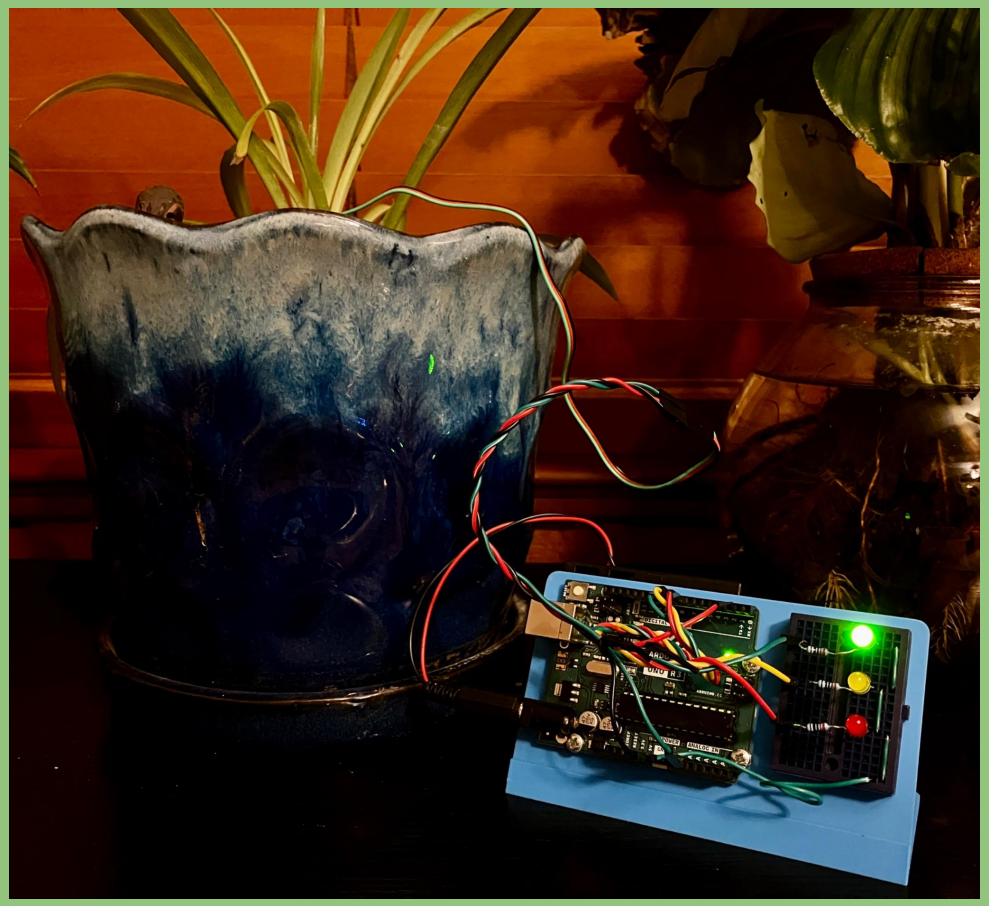
The coded ranges were adjusted so that the range of the sensor was divided into thirds, one for each LED, to provide general levels for simplicity and as a base to start from.

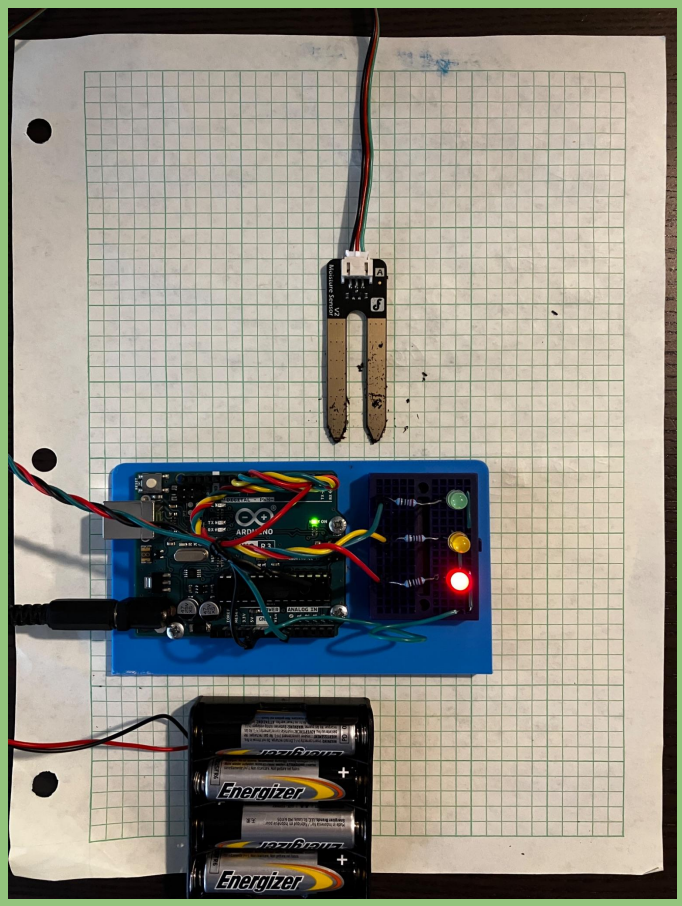
Product Test & Analysis, Prototype:





Product Test & Analysis, Final:





Improvements:

Improvements can be made by tuning LED ranges for different plants and soils. This would only require testing and observations, and minimal code modification.

Tuning how often the code activates the sensor and stores a reading would be a major improvement. Reading once each day is more realistic for real world use.

Another improvement could be to add small water pump, controlled by the microcontroller and operated only during certain levels. Adding a pump would require more electrical parts and coding, making it more complicated. However, this would allow for less interaction and even watering during vacations!

If the device were going to be used in one plant and not be moved, it could also be powered by an outlet with a power brick and cord that would plug into the microcontroller usb port. This would be one step closer to less interaction and maintenance.

Resources:

https://soiltesting.cahnr.uconn.edu/watering-houseplants/

https://docs.arduino.cc/learn/starting-guide/getting-started-arduino/

https://youtu.be/yi29dbPnu28

https://docs.arduino.cc/built-in-examples/basics/Blink/

https://wiki.dfrobot.com/Moisture_Sensor__SKU_SEN0114_