

HYDROFLORA (A CONTEXT-AWARE WATERING CAN)

By

Delilah Dzulkaflī

Idris Ispandi

Design Document for ECE 445, Senior Design, Spring 2026

TA: Mingrui Liu

Professor: Viktor Gruev

Team 55

1. Introduction.....	3
1.1 Problem.....	3
1.2 Solution.....	4
1.3 Visual Aid.....	5
1.4 High-level Requirements.....	5
2. Design.....	6
2.1 Block Diagram.....	6
2.3 Subsystem Description.....	7
2.3.1 Sensing Subsystem.....	7
2.3.2 User Interface Subsystem.....	9
2.3.3 Control Subsystem.....	12
2.3.3.1 Watering Can Controller.....	12
2.3.3.2 Sensor Node Controller.....	14
2.3.4 Water Dispensing Subsystem.....	15
2.3.5 Power Subsystem.....	18
2.4 Tolerance Analysis.....	21
2.4.1 Pump Dispense Accuracy.....	21
3. Cost and Schedule.....	22
3.1 Cost Analysis.....	22
3.2 Schedule.....	25
4. Ethics, safety, and societal impact.....	28
5. Citations.....	29

1. Introduction

1.1 Problem

Many people care for multiple houseplants with different watering needs, but watering is typically done by intuition and inconsistent habits. Because plant type, pot size, soil type, and moisture all affect how much water a plant actually needs, manual watering often results in overwatering or underwatering. Overwatering can lead to root rot, fungus gnats, and wasted water, while underwatering causes plant stress, slowed growth, and wilting. However, existing reminders or generic schedules don't adapt to real-time soil conditions, and fully automated irrigation systems can be too expensive, complex, or impractical for small indoor plant collections. Thus, there is a need for a simple, low-effort tool that helps users deliver the correct amount of water per plant based on measured soil dryness and plant/pot-specific requirements, without requiring a permanent installed system.

From a broader perspective, our project addresses environmental sustainability by reducing water waste because only the exact optimal amount of water needed for the plants will be dispensed to avoid overwatering. This project also relates to economic efficiency by preventing the loss of expensive indoor greenery as problems relating to watering that might cause the death of plants can be avoided. Furthermore, studies [11] have shown that indoor gardening has become a vital component of psychological well-being. By providing a tool that ensures plant health, our project directly supports the mental wellness of the society.

1.2 Solution

Our solution is a Smart Watering Can System. We say that the watering can is context-aware because it does the watering job based on the feedback received from the components surrounding it. The project consists of modular, low-power sensor nodes and a centralized intelligent dispensing unit. The sensor nodes are placed permanently in plant pots to monitor soil moisture levels using capacitive sensing, which avoids the corrosion issues typical of resistive probes. These nodes broadcast real-time data to the watering can via a wireless protocol (Bluetooth/WiFi).

The user interacts with a microcontroller-based interface on the watering can. When the user selects a plant through the buttons/rotary knob, the system compares the current moisture reading against a pre-set threshold for that specific plant species. Upon activation, a peristaltic pump dispenses a precise, calculated volume of water. This implementation moves plant care from being "scheduled-based" to "data-driven", ensuring that the exact lack of moisture is replenished with accurate calculations instead of human estimation.

1.3 Visual Aid

Below is the diagram of how different components will be placed in our system, with the sensor nodes attached to the plants, and the control unit on the watering can.

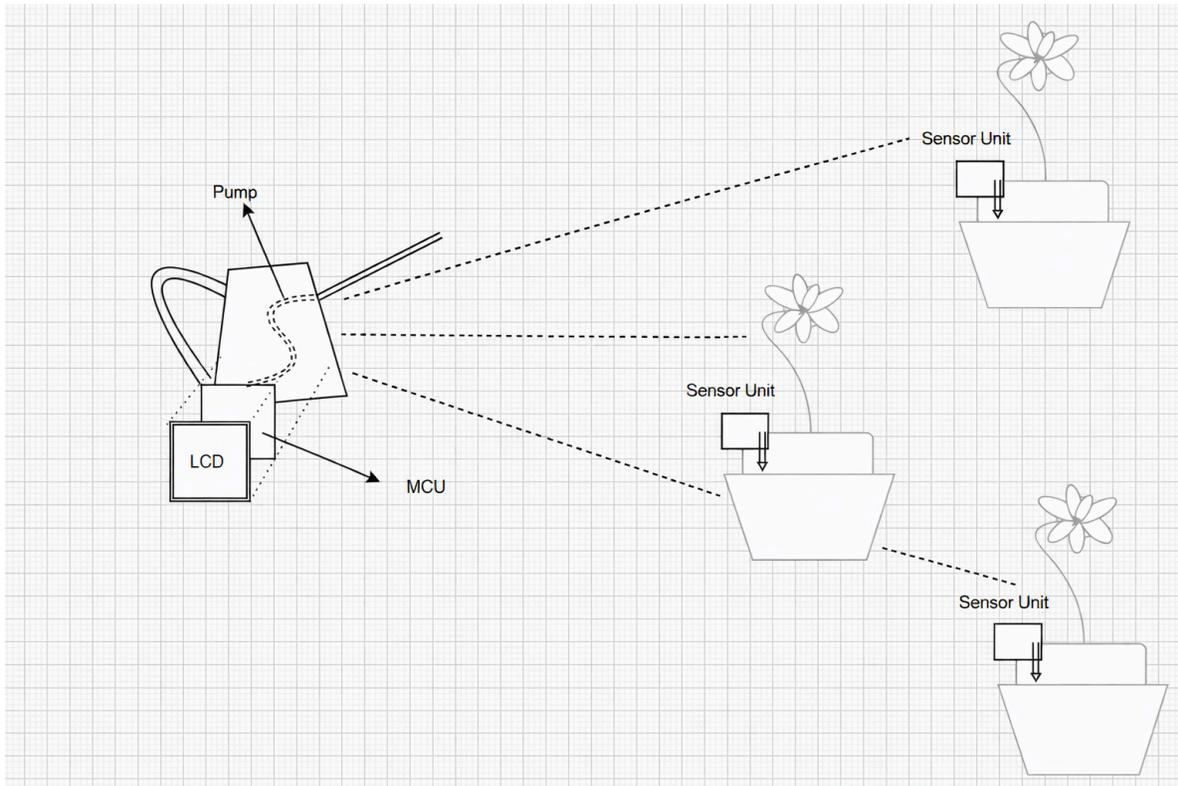


Figure 1: Visual aid of solution

1.4 High-level Requirements

- **Accuracy:** The system must dispense the calculated volume of water with a precision of $\pm 20\%$ of the target volume.
- **Connectivity:** The sensor node must maintain a reliable wireless connection and update soil moisture data to the Main Control Unit at a range of at least 3 meters in an indoor environment.
- **Longevity:** The sensor node must operate continuously for a minimum of 24 hours on a single battery charge while broadcasting data at 5-minute intervals.
- **Variation:** The system must be able to recognize different types of plants according to user input and dispense the correct amount of water based on the current plant's data.

2. Design

2.1 Block Diagram

Below is the block diagram for our system which is broken down into two main blocks serving different subsystems, namely the sensor node block and the main control unit at the watering can.

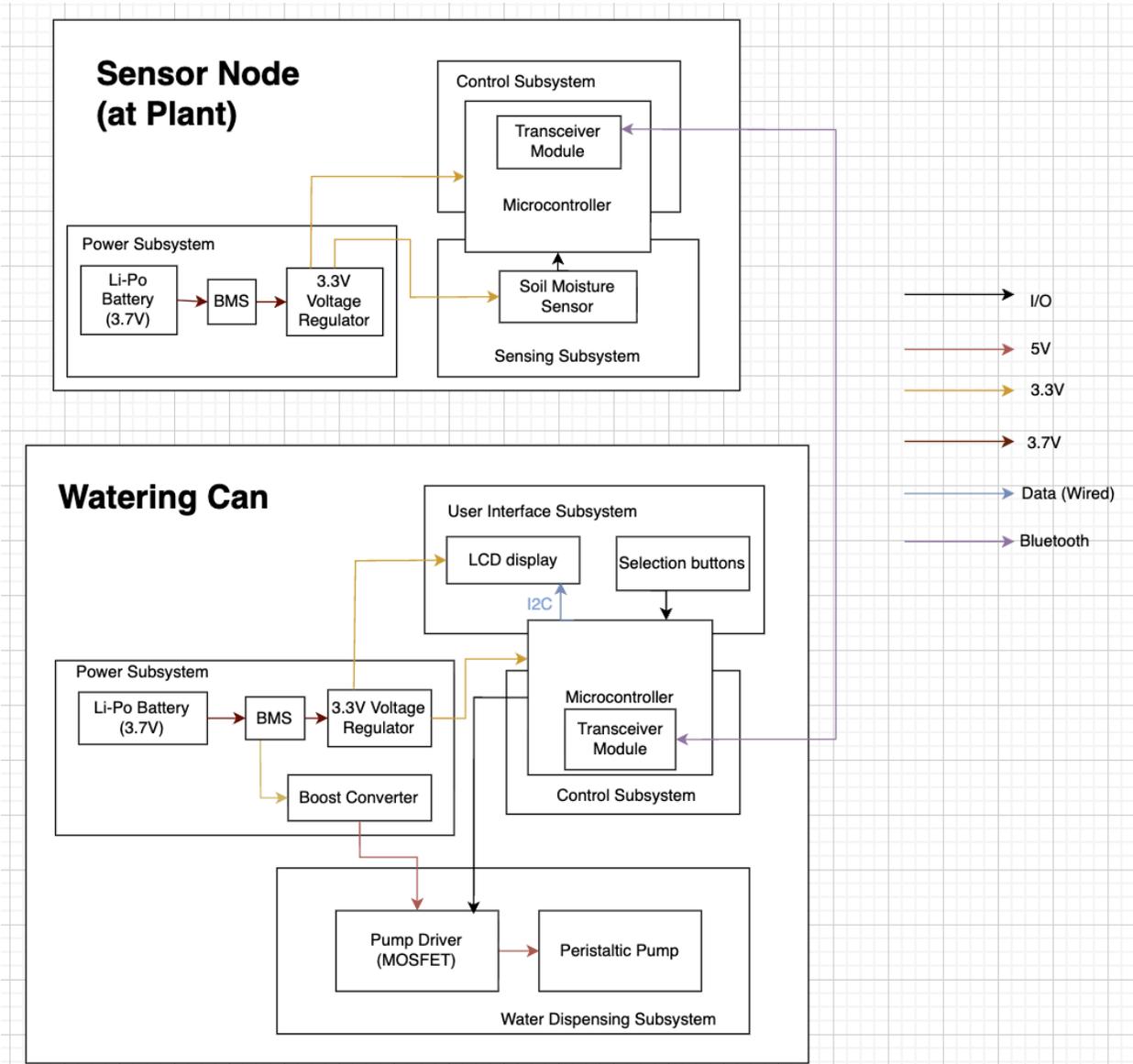


Figure 2: System block diagram

2.3 Subsystem Description

2.3.1 Sensing Subsystem

This block is located in the plant pot, and is the source of all system data. It measures the soil moisture using a capacitive probe and converts it into a digital value via the ESP32-S3's internal ADC. It then broadcasts this data wirelessly to the main control unit on the watering can. Without the sensing in this block, we cannot get the accurate moisture level, thus will fail to determine the amount of water a plant needs.

The sensor that we plan on using is a capacitive soil moisture sensor. Its advantage compared to a resistive sensor is that the probes are made of corrosion resistant material which allows them to have long service life. The figure below shows the recommended depth of sensor to be inserted in a plant pot based on the datasheet.

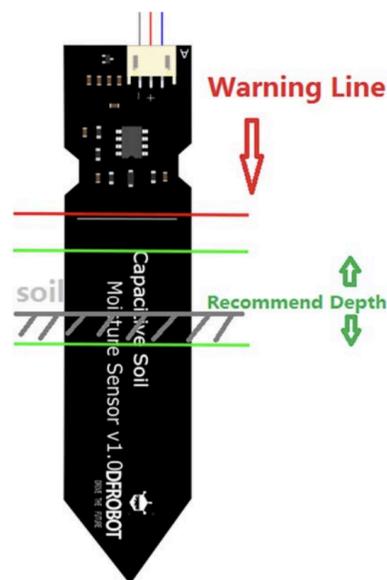


Figure 3: Diagram of capacitive moisture sensor

Before the sensor can produce useful readings, we have to calibrate it first. The following steps will be made for the calibration:

1. Record the sensor readings when exposing the probe to air. This will be the boundary value of “dry” soil.
2. Record the sensor readings when exposing the probe in a cup of water until the upper green line on the above figure. This will be the boundary value of “wet” soil.

- Since we want to get three sections of the range corresponding to “Dry, Moist, Wet”, we will take the difference of the two boundary values and divide it by three. Do note that most plants cannot stay in the “Wet” range for too long due to root rot, so one possible action for this is to disable the pump for plants in the “Wet” section.

Once we are done with calibrating the sensor, we can use the readings and convert them to a soil moisture percentage with the following formula:

$$\text{Moisture \%} = \frac{\text{Value}_{\text{Raw}} - \text{Value}_{\text{Dry}}}{\text{Value}_{\text{Wet}} - \text{Value}_{\text{Dry}}} \times 100$$

These moisture percentages will then be used by the main control subsystem to determine whether a plant needs watering.

Requirements	Verification
<ul style="list-style-type: none"> The sensing node shall remain operational for at least 24 hours without recharge under normal sampling conditions (5-minutes interval). 	<p><u>Equipment:</u> Digital Multimeter (DMM)</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> Charge the sensor node’s battery to full capacity of 4.1 V±2.4%. Set the DMM to current mode (mA) and put it in series with the battery. Measure the current during sensor node’s transmission (I_{transmit}) and during sleep (I_{sleep}). Calculate the average current draw $I_{\text{average}} = \frac{(I_{\text{transmit}} * 5s) + (I_{\text{sleep}} * 295s)}{300s}$ Confirm that I_{average} is less than (Battery Capacity/24h). <p><u>Presentation of Results:</u> Calculated battery life based on measured current draw</p>
<ul style="list-style-type: none"> The sensing probe shall resist water-induced degradation for at least 30 days of continuous exposure. 	<p><u>Equipment:</u> DMM</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> Record the physical condition of the new sensor and measure its output voltage in both dry and wet soil condition.

	<ol style="list-style-type: none"> 2) Submerge the probe in soil for 30 days with periodical watering. 3) Perform weekly inspections and check for any oxidation or corrosion of the sensor probe. 4) On day 30, record the physical condition of the sensor and measure its output voltage in dry and wet soil and compare it to the Day 1 readings. <p><u>Presentation of Results:</u> Pictures of sensor's physical condition, Table of voltage readings on Day 1 and Day 30.</p>
<ul style="list-style-type: none"> • The sensing node shall transmit a valid moisture reading within an indoor range of at least 3 meters. 	<p><u>Equipment:</u> Measuring tape, Serial monitor</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Place the sensor node and the watering can's controller at a measured distance of 3 meters apart. 2) Monitor the serial output of the watering can's controller for incoming packets. 3) Confirm that 10 consecutive readings are received successfully without corruption. <p><u>Presentation of Results:</u> Table of success rate of readings at 3 meters</p>

2.3.2 User Interface Subsystem

This block is located on the watering can, and handles all the interactions with users. It consists of an OLED to display plant types and soil moisture levels corresponding to the plant, and a rotary knob for plant selection. It ensures the system is applicable for multiple plants in order to determine how much water should be dispensed.

We prefer an OLED over LCD to achieve a better readability under any lighting condition. Since the OLED's individual pixel is its own light source, when a pixel is black, it will be completely off. This then results in a high-contrast text that is easier to read under typical indoor lighting conditions, as compared to a LCD where it has a backlight that constantly stays on. The diagram below shows the pins of the OLED display that we will be using.

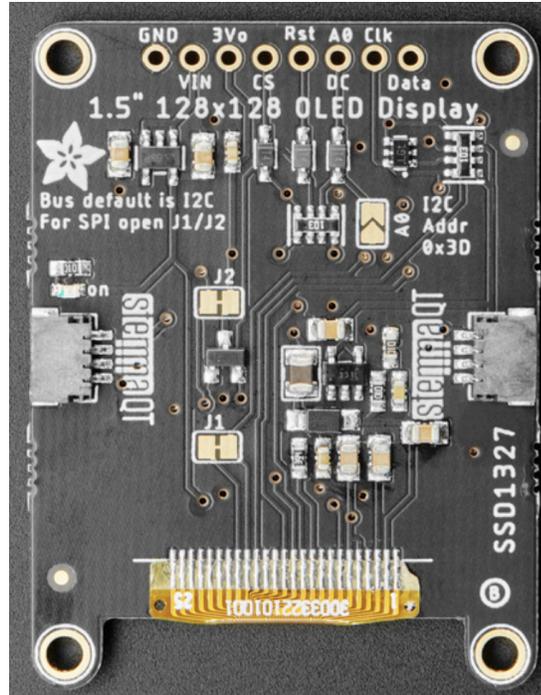


Figure 4: Diagram of pins of OLED display

Since we are planning on implementing the I2C protocol, the pins that will be used are Rst (for reset), Clk (for clock), Data (for sending the data). Below are the possible states in the FSM that we want to implement for the OLED:

- SLEEP: The screen is turned off to save power. Once the rotary knob is turned or clicked, it transitions to ACTIVE.
- ACTIVE: The screen will show the plant name with their moisture percentage level. If there is no input for 2 minutes, it transitions back to SLEEP.
- SCROLL: Cycle through three plant profiles.
- ALERT: The screen will have a small icon indicating extremely low moisture level to warn the users about which plant needs to be watered.

Requirements	Verification
--------------	--------------

<ul style="list-style-type: none"> The display shall show the current moisture percentage and selected plant name under typical indoor lighting conditions. 	<p><u>Equipment:</u> Visual inspection</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Place the display in two different room settings. 2) Trigger the sensor's moisture update by adding water to the soil. 3) Verify that the current moisture percentage and plant name are clearly visible from a 0.5m distance. <p><u>Presentation of Results:</u> Confirmation of visual legibility</p>
<ul style="list-style-type: none"> The subsystem shall not cause unintended pump activation during input interaction. 	<p><u>Equipment:</u> DMM</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Connect DMM to the Pump MOSFET Gate pin. 2) Using the rotary knob, cycle through at least 3 different plant profiles at both slow and fast pace. 3) Monitor the DMM to see if the voltage stays at $0V \pm 0.1V$. <p><u>Presentation of Results:</u> A voltage log during user input interaction</p>

2.3.3 Control Subsystem

This is a dual-part system and will be explained further below.

2.3.3.1 Watering Can Controller

For the watering can, this subsystem stores plant-specific calibration data, and calculates the precise PWM/Logic timing for the pump based on the received sensor data. Since we need different amounts of water for different types of plants corresponding to their moisture levels, we need to use a dynamic watering logic. We will convert the calculated volume of water in a specific duration of pump time. Firstly, to dispense the needed amount of water for a plant, we will use the following formula:

$$\text{Pump time (min)} = \frac{\text{Volume needed (mL)}}{\text{Pump flow constant(mL/min)}}$$

Based on the datasheet, the flow rate of the pump is up to 100mL/min. However, to determine the actual pump flow constant, we will perform the following steps:

1. Place a beaker under the tubing spout.
2. Run the pump for exactly 30 seconds.
3. Multiply the calculated volume by 2 to get the actual flow rate in mL/min.

The units of the pump flow constant and pump time can also be adjusted to be in seconds or milliseconds.

To determine the volume needed, we will do a simple test as below:

1. Take a pot of soil at 0% moisture for instance.
2. Pour in exactly 50mL of water.
3. Wait 5 minutes for the water to soak in and check the moisture percentage. Let's say it increases to 20%.
4. Now, we know how much the moisture percentage increases with 50mL of water. We can then calculate the amount of water needed per moisture percentage. In this example $50\text{mL}/20\% = 2.5\text{mL}/\%$, so we know that this plant needs 2.5mL of water to replenish 1% of moisture percentage.

Then, we can compute the volume of water needed with the following formula:

$$\text{Volume(mL)} = (\text{Target \%} - \text{Current \%}) \times (\text{The mL/\% constant})$$

Requirements	Verification
<ul style="list-style-type: none"> The controller shall store and manage data for at least 3 independent sensor nodes. 	<p><u>Equipment:</u> Serial monitor</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Power on the watering can and all 3 sensor nodes. 2) Select each node one by one. 3) Verify on the display and serial monitor the unique ID and current moisture reading for each of the 3 nodes. 4) Restart the watering can and make sure that data for all 3 nodes persists. <p><u>Presentation of Results:</u> Table showing each node IDs and their moisture readings before and after restarting the watering can controller.</p>
<ul style="list-style-type: none"> Wireless reception shall reliably operate within 3m indoor range. 	<p><u>Equipment:</u> Measuring tape, Serial monitor</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Place the sensor node and the watering can's controller at a measured distance of 3 meters apart. 2) Monitor the serial output of the watering can's controller for incoming readings. 3) Confirm that 10 consecutive readings are received successfully without corruption. <p><u>Presentation of Results:</u> Table of success rate of readings at 3 meters</p>
<ul style="list-style-type: none"> If communication with a selected node fails, the pump shall remain disabled. 	<p><u>Equipment:</u> DMM</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Select a plant profile on the watering can. 2) Power off the corresponding sensor node to simulate a communication failure. 3) Attempt to trigger the watering action via user interface. 4) Measure the voltage at the pump's MOSFET Gate pin with DMM.

	<p>5) Confirm that the voltage remains at $0V \pm 0.1V$.</p> <p>6) Repeat for all plant profiles.</p> <p><u>Presentation of Results:</u> Confirmation of $0V \pm 0.1V$ output during the simulated link failures.</p>
--	---

2.3.3.2 Sensor Node Controller

For the sensor node, this subsystem receives data from multiple sensors on the pots and broadcasts it to the watering can’s control subsystem (the main control unit) via Bluetooth. We intend the sensor node controller to operate on a “Sleep-Wake” cycle to preserve the battery life. The responsibilities of this controller includes:

1. Interfacing with the capacitive moisture sensor to get an analog voltage.
2. Converting the voltage into a meaningful moisture percentage value.
3. Transmitting the data packets to the main watering can controller.
4. Going into deep-sleep mode until the next reading.

Requirements	Verification
<ul style="list-style-type: none"> • The sensor samples every 5 minutes for a duration of 1s. 	<p><u>Equipment:</u> Oscilloscope</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Connect an oscilloscope probe to the power pin of the moisture sensor. 2) Set the oscilloscope to “Single” trigger mode on a rising edge. 3) Measure the duration that the signal remains high (indicating sampling) and the duration it remains low (indicating idle). 4) Confirm that the high time is $1s \pm 0.1s$ and the low time is $300s \pm 5s$ <p><u>Presentation of Results:</u> Waveform showing periodic pulse with corresponding time</p>

<ul style="list-style-type: none"> • If communication fails, the node shall return to sleep. 	<p><u>Equipment:</u> DMM</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Power off the watering can controller. 2) Connect the DMM in series with the battery of the sensor node to monitor the current draw. 3) Observe the node wakes up (periodic behavior) and current should increase. 4) Verify that after the programmed timeout of 10 seconds, the current drops down to <1mA implying it went to sleep. <p><u>Presentation of Results:</u> Confirmation that the current of sensor node battery returned to sleep level after the connection failure</p>
<ul style="list-style-type: none"> • The node shall maintain periodic measurement operation without requiring external reset. 	<p><u>Equipment:</u> PC with Serial monitor</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Connect the sensor node to a PC and leave it running in a soil pot for one hour. 2) Monitor the timestamps of incoming data from the sensor on the serial monitor. 3) Verify that the node continues to send updates at the expected interval without crashing. <p><u>Presentation of Results:</u> A timestamp log over a one hour duration</p>

2.3.4 Water Dispensing Subsystem

This block is the physical actuator. It takes the low-power logic signal from the Control Subsystem and uses a power MOSFET to switch a 5V supply to the peristaltic pump, moving water from the reservoir to the spout.

We choose to use a peristaltic pump for a better precision level. A peristaltic pump works by using rollers to “squeeze” or compress a flexible tube, creating a vacuum which draws water through the tube. Since only the pump tubing will be in contact with the water, this eliminates the risk of the pump itself touching the water. Besides, the pump can dispense a fixed volume of

water because the rollers move a fixed volume of water per rotation. The diagram below shows the schematic of the pump motor driver.

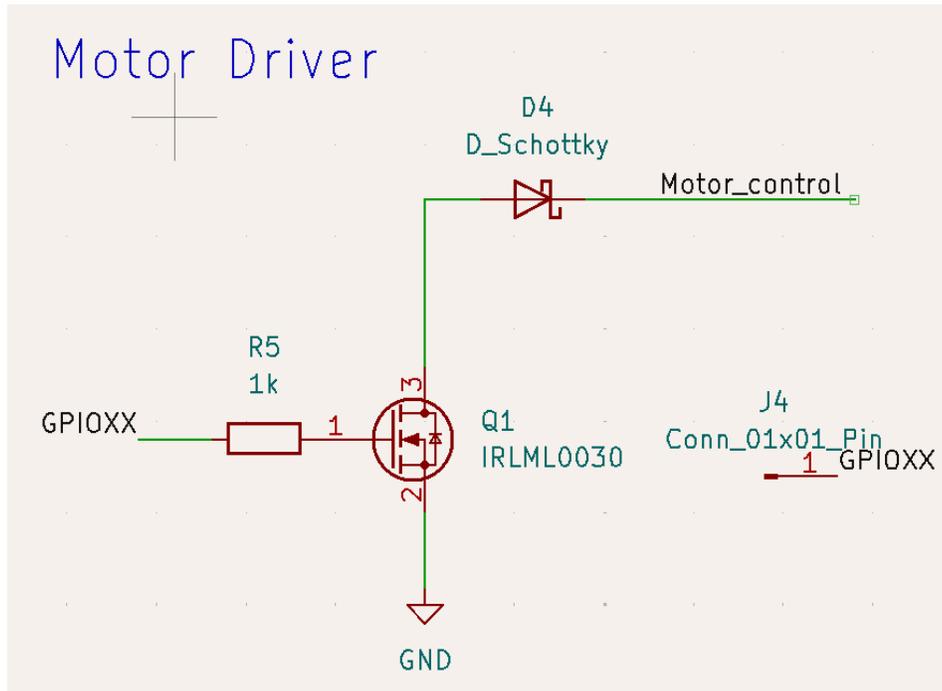


Figure 5: Schematic of motor driver

For electrical safety reasons, we incorporated a Schottky diode as a flyback diode. It is placed across the motor terminals. When the MOSFET turns off, the motor’s magnetic field collapses and creates a high-voltage spike that can fry the ESP32. The diode will then give this energy a safe path to dissipate and “recycle” this energy back into the motor instead of letting it hit the MOSFET.

Requirements	Verification
<ul style="list-style-type: none"> The dispensed volume should be stable around plus or minus 20 percent volume accuracy. 	<p><u>Equipment:</u> Beaker</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Program the system to dispense 100mL of water 2) Collect water from the pump in the beaker 3) Repeat for 5 times 4) Confirm that all trials fall within $\pm 20\%$ of 100mL. <p><u>Presentation of Results:</u></p>

	Table of volume of water dispensed across all 5 trials
<ul style="list-style-type: none"> Water shall begin flowing within 3s \pm1s of activation. 	<p><u>Equipment:</u> Stopwatch, beaker</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Position the pump tubing in a beaker. 2) Trigger the pump and start the stopwatch simultaneously. 3) Stop the stopwatch once the first drop of water enters the beaker. 4) Record the time taken on stopwatch and confirm that it is within 3s \pm 1s. 5) Repeat for 5 times. <p><u>Presentation of Results:</u> Table of time taken across 5 trials</p>
<ul style="list-style-type: none"> The subsystem shall operate continuously for up to 60 seconds without overheating or shutdown. 	<p><u>Equipment:</u> Infrared thermometer</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Run the pump for a continuous 60-second cycle. 2) Immediately measure the temperature of the MOSFET and the pump motor. 3) Confirm that the temperature of the pump is below 40 degrees Celsius. <p><u>Presentation of Results:</u> Table showing recorded temperature of components after 60s</p>

2.3.5 Power Subsystem

This is a dual-part system.

- The **Watering Can** gets ~5V(from LiPo step up) and ~3.3 V(LDO step down). NOTE the values aforementioned are approximations with margins specified further below.
- The **Sensor Node** uses a ~5V(from LiPo step up) with a LDO (Low Dropout) regulator to get 3.3V

Since the voltages required for several components are different, we will have to step-up and then step-down the ~3.7V from rechargeable batteries using the LD1117-3.3 (LDO) regulator and a boost converter. The diagram below shows the schematic of how these components come into play.

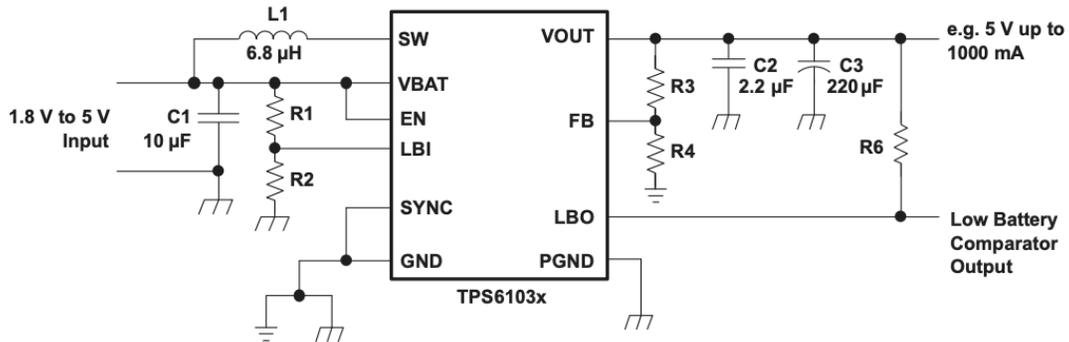


Figure 6: Schematic of boost converter

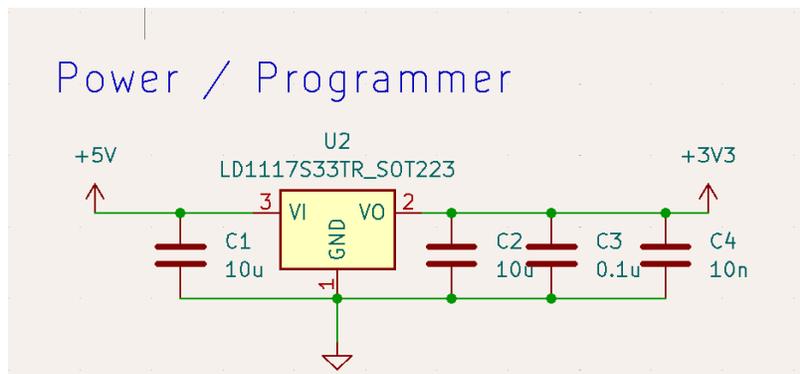


Figure 7: Schematic of LDO regulator

Since we are using rechargeable batteries, we will also incorporate a charging unit to charge the 3.7V batteries when connected to a standard 5V USB supply. Below is the schematic for the battery charger.

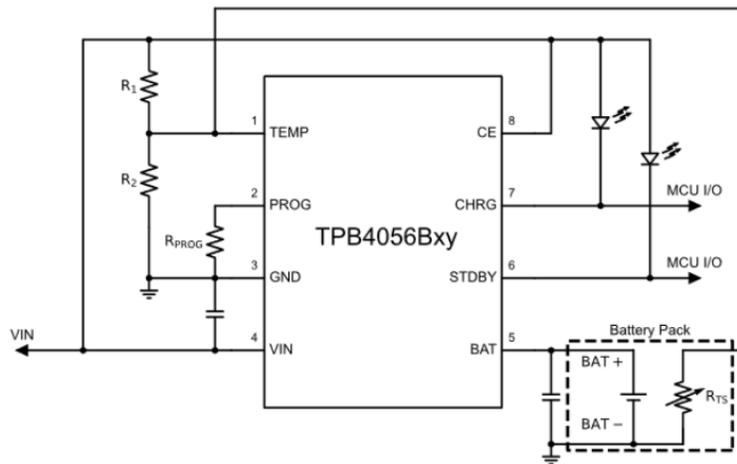


Figure 8: Schematic of battery charger

Requirements	Verification
<ul style="list-style-type: none"> The LDO regulator shall step down boosted voltage ~5V to a stable 3.3V \pm 5% for the microcontroller logic. 	<p><u>Equipment:</u> DMM, DC Bench Power Supply</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Connect the bench power supply set to 5V to the regulator's input. 2) Measure the LDO output pin with DMM and confirm that the reading is between 3.3V \pm 5%. 3) Swing the LDO input to 5.0V \pm 0.1V. 4) respectively, and verify the output remains within 3.3V \pm 5%. <p><u>Presentation of Results:</u> A table showing LDO output voltages at input voltages of 4.9 V, 5V, and 5.1V.</p>
<ul style="list-style-type: none"> The logic supply voltage shall remain within plus or minus 5 percent of nominal (3.3V) during pump activation. 	<p><u>Equipment:</u> Oscilloscope</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Connect the oscilloscope probe to the 3.3V logic rail. 2) Trigger the pump. 3) Monitor the oscilloscope for any sag or surge. 4) Verify the voltage stays within \pm 5% of 3.3V during pump activation. <p><u>Presentation of Results:</u> Oscilloscope trace showing 3.3V stability</p>

	under load
<ul style="list-style-type: none"> The boost converter shall step up the (3.7V - 4.2V) battery voltage to a stable 5.0V \pm 5% to ensure the peristaltic pump maintains a consistent flow state regardless of battery discharge level. 	<p><u>Equipment:</u> DMM, battery</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Connect the 3.7V battery to the input of the boost converter. 2) Measure the output voltage of the boost converter with DMM. 3) Record the readings and confirm it is within 5.0V \pm 0.1V. <p><u>Presentation of Results:</u> Table showing the input voltages and output voltages of the boost converter</p>
<ul style="list-style-type: none"> The sensor node supply shall maintain low standby current to preserve battery life. 	<p><u>Equipment:</u> DMM</p> <p><u>Test Procedures:</u></p> <ol style="list-style-type: none"> 1) Put the sensor node in deep sleep mode. 2) Measure the current draw at the battery terminal. 3) Confirm that the reading is below 100uA. <p><u>Presentation of Results:</u> Table showing current measurement during standby mode</p>

2.4 Tolerance Analysis

2.4.1 Pump Dispense Accuracy

A significant risk to project success is the accuracy of the water volume dispensed. Peristaltic pumps are generally consistent, but flow rate can vary based on battery voltage and tubing friction. To ensure the +/- 20% requirement, we must calculate the required timing required to dispense the required volume.

$$Q = 100 \text{ mL/min}$$

$$Q = \frac{100}{60} \text{ mL/s} \approx 1.66 \text{ mL/s}$$

$$t = \frac{V_{\text{target}}}{Q}$$

$$t = \frac{50 \text{ mL}}{1.66 \text{ mL/s}} \approx 30 \text{ s}$$

This shows that the pump dispenses 50mL in approximately 30s. So by fixing the pump time to 30s, we are dispensing 50mL.

We can then measure the accuracy using the following formula:

$$\text{Accuracy} = \frac{V_{\text{actual}}}{V_{\text{target}}} \times 100\%$$

This will allow us to see the current performance of the pump control and figure out if we need to improve motor output and other peripherals.

3. Cost and Schedule

3.1 Cost Analysis

- Labor:

Assuming an average hourly salary of an ECE graduate from University of Illinois to be \$45, the total hours spent on this project to be 7 hours per week with a total of 9 weeks, the labor cost will be $(\$45/hour) * 2.5 * 7 * 9 = \7087.50 per person.

Since we have two members, the total labor cost will be $\$7087.50 * 2 = \14175 .

- Parts:

Description	Manufacturer	Part #	Quantity	Unit Price	Extended Price	Link
RF TXRX MODULE BT PCB TRACE SMD	Espressif Systems	ESP32-S 3-WROO M-1-N16	4	\$5.92	\$23.68	ESP32-S3-W ROOM
Peristaltic Liquid Pump with Silicone Tubing - 5V to 6V DC Power	Adafruit Industries LLC	3910	1	\$24.95	\$24.95	Peristaltic Pump
Capacitive Soil Moisture Sensor	DFRobot	SEN0193	3	\$5.90	\$17.70	Soil Moisture Sensor
3.7V 1000mAh LiPo Battery Rechargeable 4 Pack	MakerHawk	102050	1	\$18.99	\$18.99	Battery Pack
1-A Single Cell Li-Ion Battery Linear Charger	3PEAK	TPB4056 B2X-ES1 R	4	\$0.50	\$2.00	Battery Charger
MOSFET N-CH 30V 5.3A SOT23	Infineon Technologies	IRLML00 30TRPBF	1	\$0.50	\$0.50	MOSFET

DIODE SCHOTTKY 40V 5A DO214AC	Comchip	CDBA54 0-HF	1	\$0.45	\$0.45	Schottky Diode
IC REG LINEAR 3.3V 1A SOT-223-3L	UMW	LD1117- 3.3	4	\$0.30	\$1.20	Voltage regulator 3.3V
Adafruit Grayscale 1.5" 128x128 OLED Graphic Display - STEMMA QT / Qwiic	Adafruit	4741	1	\$22.50	\$22.50	Display
ENCODER MECH QUAD VERT PC PIN	Adafruit	377	1	\$4.50	\$4.50	Rotary Encoder
IC REG BOOST ADJ 3.6A 16HTSSOP	Texas Instruments	TPS6103 0PWP	1	\$3.86	\$3.86	Boost Converter
CONN RCPT USB2.0 MICRO B SMD R/A	Amphenol	Micro USB-B	1	\$0.49	\$0.49	Connector Micro USB
CAP CER 10UF 50V X7R 0805	SAMSUNG	10uF	1	\$0.11	\$0.11	Capacitor 10uF
CAP CER 4.7UF 50V X7R 0805	YAGEO	4.7uF	2	\$0.38	\$0.76	Capacitor 4.7uF
CAP CER 0.1UF 50V X7R 0805	YAGEO	0.1uF	2	\$0.10	\$0.20	Capacitor 0.1uF
LED GREEN CLEAR CHIP SMD	LITE-ON	GREEN LTST-C17 1GKT	1	\$0.11	\$0.11	Green LED
LED RED CLEAR 1206 SMD	LITE-ON	RED LTST-C15 0CKT	1	\$0.15	\$0.15	Red LED
RES 5.1K OHM 5% 1/8W 0805	Stackpole Electronics	RMCF08 05JT5K1 0	3	\$0.10	\$0.30	Resistor 5.1k

SWITCH TACTILE SPST-NO 0.05A 12V	C&K	Tactile	2	\$0.36	\$0.72	Switch Tactile
Watering Can	Etarewilt	-	1	\$12.99	\$12.99	Watering Can
				Total	\$136.16	

The highlighted parts in the table are components that we have to purchase outside of the ECE E-shop and those purchases will be made mostly from DigiKey or Amazon. The total cost of the parts will be scaled according to the number of sensor nodes implemented; for now, we are doing three sensor nodes for three plants.

- Grand total:

Summing up \$14175 + \$136.16 from both parts, the total cost including labor and materials turns out to be **\$14311.16**.

3.2 Schedule

Assume worst case scenario PCB arrives 2 weeks after order

WEEK	HIGH LEVEL DESCRIPTION
1: 3/2 - 3/8	<p>Design Review - Monday</p> <ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Send in non E-shop components order form - Solder and test First PCB board (if arrived) - Update Control Unit PCB for second round order - Complete and Send in PCB design for sensor node - Software <ul style="list-style-type: none"> - Prototype BLE data communication using Dev board - Drive a simple display
2: 3/9 - 3/15	<p>Breadboard Demo- TBD</p> <ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Solder and test Both PCB board (if arrived) - Update Control Unit PCB Design for third round order <ul style="list-style-type: none"> - Update to include Rechargeable LiPo - Complete and Send in PCB design for both PCB board - Define the PCB dimensions for PCB housing and pass dimensions to machine shop - Software <ul style="list-style-type: none"> - Verify BLE data communication ESP-ESP <ul style="list-style-type: none"> - Try out Wi-Fi communication to see Power Tradeoff - Drive a simple display and display communicated data <p>BREADBOARD DEMO ONE Description</p> <ul style="list-style-type: none"> - Demo ESP-ESP communication - Demo Displaying communication data <p>Ideal Case - breakout PCB arrived and working Minimal Case - Functional requirements as above using dev board</p>
3: 3/16 - 3/22	<p>SPRING BREAK</p>
4: 3/23 - 3/29	<ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Solder and test Both PCB board (if arrived) - Update Both PCB board for FINAL round order <ul style="list-style-type: none"> - Mainly reducing breakouts for compactness - Verify power Unit and see if need to swap different boost converter - Complete and Send in PCB design for both PCB board - Software <ul style="list-style-type: none"> - Generate sensor reading and send to Main MCU - Display data on simple display
5: 3/30 - 4/5	<p>Individual progress reports</p> <ul style="list-style-type: none"> - Hardware

	<ul style="list-style-type: none"> - Solder and test Both PCB board (if arrived) - Debugging <ul style="list-style-type: none"> - Test Voltage Output over time - Test Water Volume being Output by peristaltic pump - Software <ul style="list-style-type: none"> - Port previously developed code and try running on PCB MCU <ul style="list-style-type: none"> - Develop communication pipeline for Moisture sensor (On Chip -> Wireless transmit) - Debugging <p>NOTE: At this point under the worst case assumption that the board takes 2 weeks to arrive the non E-shop orders should arrive sometime this week.</p>
6: 4/6 - 4/12	<p>Progress Demo</p> <ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Solder and test Both PCB board (if arrived) - Debugging - Software <ul style="list-style-type: none"> - Test code and try running on PCB MCU - Debugging - Integrated Testing <ul style="list-style-type: none"> - Compare Actual water volume outputted vs Ideal water outputted - Compare Sensor node reading vs Control Unit reading. <p>NOTE: At this point , under the worst case assumption that the board takes 2 weeks to arrive, the last round of PCB orders should arrive sometime this week.</p>
7: 4/13 - 04/19	<ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Solder and test Both PCB board (if arrived) - Debugging - Software <ul style="list-style-type: none"> - Test code and try running on PCB - Debugging - Refinement <ul style="list-style-type: none"> - Refine the code based on previous actual values. <ul style="list-style-type: none"> - ie non ideal buffering/ error correction
8: 4/20 - 4/26	<p>Mock demo & Mock Presentation</p> <ul style="list-style-type: none"> - Hardware <ul style="list-style-type: none"> - Debugging - Software <ul style="list-style-type: none"> - Debugging
9: 4/27 - 5/3	<p>Final demo & Final Presentation</p>

10: 5/4 - 5/7

Final papers

4. Ethics, safety, and societal impact

According to the IEEE Code of Ethics 1.1 [1], our design prioritizes the safety and health of the public. For safety concerns, we will apply protection to the components to provide isolation between water and electronics. We will use a waterproof enclosure to house all the circuitry on the watering can.

Aligning with IEEE Code of Ethics 1.4 [1] and ACM Code of Ethics 1.3 [2], we will avoid unlawful conduct and provide full disclosure of all systems capabilities and limitations while doing our project. We are committed to honest representation of our sensor data and avoid ethical breach of falsifying data or hard-coding success values. We should always report true soil moisture levels and this will be shown by implementing transparent calibration procedures.

Furthermore, one of the regulatory standards that is relevant to our project is the UL1642 standard [3] where the Li-Po batteries used will be equipped with a BMS (battery management system) to prevent overcharging and short circuits.

As for the societal impact, by encouraging water conservation, our project promotes environmental stewardship according to the National Oceanic and Atmospheric Administration (NOAA). Besides, by providing a solution that makes plant care easy, we are promoting indoor gardening which helps in an individual's mental well-being. From an economical perspective, this project provides a low-cost solution for plant hobbyists and can even extend to small-scale farmers who cannot afford industrial irrigation.

5. Citations

- [1] IEEE, "IEEE Code of Ethics," *IEEE.org*, 2024. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8>. [Accessed: Feb. 11, 2026].
- [2] Association for Computing Machinery, "ACM Code of Ethics and Professional Conduct," *ACM.org*, 2018. [Online]. Available: <https://www.acm.org/code-of-ethics>. [Accessed: Feb. 11, 2026].
- [3] Underwriters Laboratories, "UL 1642 Standard for Lithium Batteries," *LiPol Battery*, 2023. [Online]. Available: https://www.lipolbattery.com/UL1642_Certification_of_Lithium-ion_Battery.html. [Accessed: Feb. 11, 2026].
- [4] Adafruit Industries, "Peristaltic Liquid Pump with Silicone Tubing," *DigiKey Electronics*, 2024. [Online]. Available: <https://www.digikey.com/en/products/detail/adafruit-industries-llc/1150/5638299>. [Accessed: Feb. 13, 2026].
- [5] DFRobot, "Capacitive Soil Moisture Sensor - Corrosion Resistant," *DigiKey Electronics*, 2024. [Online]. Available: <https://www.digikey.com/en/products/detail/dfrobot/SEN0193/6588605>. [Accessed: Feb. 13, 2026].
- [6] DFRobot, "Capacitive Soil Moisture Sensor v1.0," *SEN0193 Datasheet*, 2024. [Online]. Available: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2107/SEN0193_Web.pdf. [Accessed: Feb. 13, 2026].
- [7] Espressif Systems, "ESP32-C3 Series Datasheet," *Espressif Documentation*, v1.0, 2024. [Online]. Available: https://documentation.espressif.com/esp32-c3_datasheet_en.pdf. [Accessed: Feb. 13, 2026].
- [8] Adafruit, "Monochrome 128x64 OLED Graphic Display Module Kit," *Adafruit Industries*, 2024. [Online]. Available: <https://www.adafruit.com/product/4741>. [Accessed: Feb. 13, 2026].
- [9] Missouri Botanical Garden, "Overwatering," *Help for the Home Gardener*, 2024. [Online]. Available: <https://www.missouribotanicalgarden.org/gardens-gardening/your-garden/help-for-the-home-gardener/advice-tips-resources/insects-pests-and-problems/environmental/overwatering>. [Accessed: Feb. 13, 2026].
- [10] Perennial Wellness, "The Therapeutic Power of Indoor Gardening for Mental Wellness," *Perennial Wellness Blog*, 2023. [Online]. Available: <https://www.perennialwellnesswny.com/post/the-therapeutic-power-of-indoor-gardening-for-mental-wellness>. [Accessed: Feb. 13, 2026].

[11] NOAA, "Environmental Stewardship Societal Impact: Common Measure Definitions," *NOAA Education Council*, 2023. [Online]. Available: <https://www.noaa.gov/office-education/noaa-education-council/monitoring-resources/common-measure-definitions/stewardship-definitions>. [Accessed: Feb. 13, 2026].

[12] The Grainger College of Engineering, "Computer Engineering," *University of Illinois Urbana-Champaign*, 2024. [Online]. Available: <https://grainger.illinois.edu/academics/undergraduate/majors-and-minors/computer-engineering>. [Accessed: Feb. 27, 2026].

[13] Watson-Marlow Fluid Technology Solutions, "Peristaltic pumps: how they work," *W-M Support: Pump Principles*, 2024. [Online]. Available: <https://www.wmfts.com/en-us/support/pump-principles/peristaltic-pumps-how-they-work/>. [Accessed: Feb. 27, 2026].

[14] Texas Instruments, "TPS6103x Efficient Boost Converter With 4-A Switch," *TPS61030 Datasheet*, SLVS454C, 2023. [Online]. Available: <https://www.ti.com/lit/ds/symlink/tps61030.pdf>. [Accessed: Feb. 27, 2026].