UNLV

College of Engineering
College of Education

Smart Gardening System

Project Overview
This Arduino-based project will be focused on building a smart gardening system.

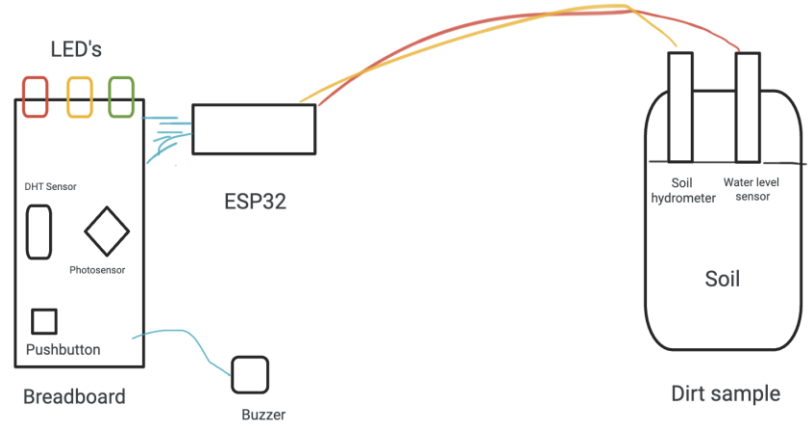
Course Connections	21st Century Skills	CTE Alignment
Middle School Science High School Physics High School Engineering	Communication, Critical Thinking, Collaboration	Energy, Environment, and Development Pathway

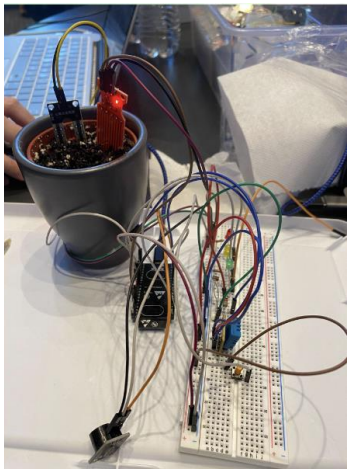
Objective
Students will build a smart gardening system to support a self-sustaining garden.

Materials

1. ESP32 Microcontroller
2. Soil Hydrometer
3. Water Level Sensor
4. Temperature & Humidity (DHT) Sensor
5. Buzzer
6. Photosensor
7. Push button
8. 3 LED's, 1 green, 1 yellow, and 1 red
9. Breadboard
10. Wires (Male to female)
11. USB Cable
12. Computer

Block Diagram





Application

Traditional gardening methods can be wasteful of resources, specifically water, and difficult to maintain. A smart gardening system notifies users of moisture levels, indicating too much or too little moisture. It increases efficiency and encourages a healthier garden. Climate change is a significant issue, so this system will also encourage a better ecosystem by conserving water.

Outline/Schedule

Part 1: Introduction to Smart Gardening System (complete by Jun. ??)

- Introduce the problem of traditional gardening methods and how it can be inefficient and wasteful.
- Introduce the solution of a smart gardening system and explain the benefits of using it.
- Discuss the materials needed for the project, draw the block diagram, explain how the system works.
- Discuss the smart garden showcase.
- Assign team tasks, distribute materials, and plan the schedule.
- Record video clips for important steps.

Part 2: Programming the ESP32 Microcontroller & DHT Sensor (complete by Jun. ??)

- Introduce the ESP32 Microcontroller and its capabilities.
- Review the basics of programming with Arduino and ESP32.
- Install the DHT Sensor Library library and show the connection between ESP32 and the DHT Sensor module.

- Demonstrate how to program the ESP32 to take input from the DHT sensor with the DHT module and output the results to the serial monitor.
- Allow team time to program their microcontrollers and test them with the given sensors.
- Record video clips for important steps.

Part 3: Adding Remaining Sensors (complete by Jun. ??)

- Introduce the Soil Hydrometer and how it can be used to detect moisture levels in the soil.
- Introduce the Water Level Sensor to detect water levels in the soil.
- Introduce the Photo Sensor to detect light/sun exposure.
- Demonstrate how to connect and control the sensors with the ESP32 Microcontroller.
- Allow teams time to add the sensors to their project and test it.
- Record video clips for important steps.

Part 4: Adding LEDs, Buzzer, and ThinkSpeak Integration (complete by Jun. ??)

- Introduce the use of LEDs to indicate the status of the moisture level.
- Demonstrate how to connect and control LEDs with the ESP32 Microcontroller.
- Install the Wifi and ThinkSpeak libraries and how they can be used to display the collected data from the sensors.
- Allow teams time to integrate LEDs, the buzzer, and Thinkspeak into their project and test it.
- Record video clips for important steps.

Part 5: Finalizing and Testing the Project (complete by Jun. ??)

- Design the testing plan based on the project objectives and requirements.
- Discuss the troubles that students may encounter and how to troubleshoot.
- Allow teams to prepare the showcase.
- Allow teams time to finalize their project and test it according to the testing plan provided.
- Have teams present their projects and demonstrate their functionality.
- Record video clips for important steps.
- Prepare the project presentation and make the video demo.

Reflection

- Have students reflect on their experience with the project and what they have learned about programming, robotics, and Arduino.
- Discuss how STEM skills and knowledge can be applied to real-world problems and applications.

- Allow time for students to provide feedback on the project and suggest improvements for future iterations.

Sample Code
Sample Code