Task 3 : Report

Project Name: Hydroponic IoT Monitoring System using ESP32 and Adafruit IO

A circuit board with wires and switches

AI-generated content may be incorrect.

Wokwi

A screenshot of a computer

AI-generated content may be incorrect.

Adafruit IO - Dashboard

Author: Hiten Shah

Student ID: 20078332

Simulation Platform: Wokwi / Adafruit IO

Contents

[Introduction and Project Vision 3](#_Toc202273034)

[Hardware Connections and Data Flow: 3](#_Toc202273035)

[Sensor Selection and Integration: 4](#_Toc202273036)

[Communication Protocol Details: 4](#_Toc202273037)

[Data Processing Method: 4](#_Toc202273038)

[Power Management Considerations: 5](#_Toc202273039)

[Regulatory Compliance: 5](#_Toc202273040)

[System Benefits: 5](#_Toc202273041)

[Error Handling Approach: 5](#_Toc202273042)

[Key Features Summary: 6](#_Toc202273043)

[Appendix A – Useful references 7](#_Toc202273044)

# Introduction and Project Vision

The goal of this project is to simulate and implement a hydroponic monitoring system that ensures reliable water circulation and prevents water related failures. The system automates monitoring of inflow and outflow water using flowmeter and detect conditions such as reservoir water level and trough pipe overflow using float switches. All data is transmitted to Adafruit IO for real-time visualisation and remote control of the water pump. This supports proactive maintenance, system automation and efficient water management.

# Hardware Connections and Data Flow:

* Component interconnections:
  + Flowmeters – Simulated by generating square wave pulses connected to GPIO pins D3 and D4
  + Float switches
    - D6 – Horizontal Float Switch for Reservoir warning level
    - D7 – Horizontal Float Switch for Reservoir critical level
    - D8 – Vertical Float Switch for Pipe overflow warning
  + Pump Control Switch: D5 for manual override
  + LEDs: D9 (Pump), D12 (Reservoir warning), D13 (Reservoir critical), D14 (Pipe overflow)
  + Potentiometers: Analog inputs A1 and A2 to simulate variable flow rates.
* Data Flow:
  + Interrupts capture pulses from flowmeters to measure water volume
  + Sensor readings update internal states
  + Data is published via MQTT to Adafruit IO every 30 seconds or when state changes. (We chose 30 seconds update as the Hydroponic system will be able to manage that level of delay, if any urgent action needs to be taken, the interrupts will handle them.)
* Signal conditioning/preprocessing:
  + Debouncing logic is implemented in software to prevent false triggers from the float switches (push buttons for simulation).
  + Edge detection via interrupts ensures responsive and efficient signal handling.
  + Flowmeter pulses are counted and converted to flow rate using a calibrated litres per pulse value (7.5L / pulse assumed).
  + Critical sensor states automatically disable the pump to prevent from system damage.

# Sensor Selection and Integration:

* Selected sensors:
  + Flowmeters simulators to measure the inflow and outflow rate of water via pulse frequency.
  + Float switches (push buttons) simulate digital HIGH / LOW levels to indicate warning, critical and overflow states.
* Integration approach:
  + All sensors are simulated via GPIO and processed using hardware interrupts.
  + Each float switch has a dedicated ISR to toggle states and trigger MQTT updates.
  + Pump control logic checks critical / overflow states and disables the pump to prevent flooding or system failure.
* Justification:
  + Simple digital sensors reduce complexity and cost.
  + Interrupt based design avoids continuous polling and is highly responsive.

# Communication Protocol Details:

* Protocol selection:
  + Sensor to Microcontroller: GPIO (digital input/output pins)
  + Microcontroller to Cloud: MQTT (Adafruit IO broker service)
* Implementation approach:
  + The ESP32 connects to Wi-Fi and publishes sensor data using MQTT every 30 seconds or on state change.
  + Subscriptions allow remote pump override from dashboard.

# Data Processing Method:

* Cloud platform: [Adafruit IO Dashboard](https://io.adafruit.com/hcshah26/dashboards/hydroponic-project)
  + Transmission method:
    - Sensor data and pump status are published to the Adafruit IO feeds.
    - Dashboard widgets display:
    - Pump inflow and return flow rates (numeric L/min)
    - Sensor alerts (toggle indicators)
    - Pump control switch (bi-directional control)
  + Visualisation:
    - Real-time flow rates
    - Digital indicators for overflow and warnings
    - Feed values updated with each trigger or scheduled refresh
* Local processing: ESP32
  + GPIO pins to get inputs from flowmeters, float switch sensors and pump override switch
  + LED lights to indicate states for flowmeters, float switches and pump override
  + No local storage, all data is published to Adafruit IO via feeds.

# Power Management Considerations:

* Power requirements:
  + Simulated in Wokwi using constant USB power supply
  + In real deployment, ESP32 and sensors would require 3.3 to 5V input and 250mA current.
* Efficiency measures:
  + Wi-Fi connection and MQTT updates optimised to only send data on state change
  + Future enhancements could include sleep modes for battery powered systems

# Regulatory Compliance:

* Applicable standards:
  + Electrical Safety: Compliance with Australia/New Zealand standards (AS/NZS 3000)
  + Data Privacy: Adafruit IO adheres to secure connection protocols and user credentials
  + EMC: Certified microcontrollers and sensor components to reduce interference
* Compliance measures:
  + Use of published secure MQTT protocol
  + Enclosure design and electrical insulation to prevent short circuits and hazards

# System Benefits:

|  |  |
| --- | --- |
| Feature | Benefit |
| Flow monitoring | Prevents water waste and detects flow anomalies |
| Float switches | Warns of low water or overflow before damage occurs |
| Remote dashboard | Allows monitoring and pump control from anywhere |
| Alerting system | Provides proactive notification for intervention |
| Simulation support | Enables safety testing without live water |

* Alignment with user stories:
  + Maintenance technician receives alerts for water level issues
  + System admin can remotely stop or start the pump and monitor usage trends

# Error Handling Approach:

* Detection methods:
  + Software debouncing for mechanical switch errors
  + MQTT connection loss detection and reconnection logic
  + Interrupt fallbacks to ensure pulse counts aren’t missed
* Response procedures:
  + If overflow or critical condition detected, pump is shutdown automatically
  + Failed MQTT publishes are logged to serial output and retried
  + Float switch and flowmeter toggles retrigger alert if stuck

# Key Features Summary:

The hydroponic monitoring system developed in this project offers a robust and scalable solution for managing water flow and reservoir safety. Key features include:

* **Real-time Monitoring:** The system continuously tracks the inflow and return flow rates using simulated flowmeters, providing accurate readings in litres per minute.
* **Cloud-Based Dashboard:** Integration with Adafruit IO enables remote monitoring and control, offering a user-friendly interface for viewing sensor data and managing system operations.
* **Automated Safety Controls:** Built-in logic disables the pump automatically when critical conditions are detected – such as very low reservoir levels or pipe overflow – helping to prevent equipment damage or water loss.
* **Responsive Sensor Handling:** Float switches are processed through hardware interrupts with software debouncing to ensure immediate and reliable state changes.
* **Efficient Data Communication:** MQTT protocol is used to transmit and receive sensor data and control commands, minimising bandwidth use while maintaining real-time responsiveness.
* **Manual Override Capability:** A local pump override switch and remote toggle via the dashboard allow users to intervene as needed for maintenance and emergency actions.
* **Error Handling and Recovery:** The system includes mechanism to detect sensor faults and reconnect to the cloud platform if the MQTT connection is lost, ensuring resilience and uptime.

These features collectively support the system’s goals of improving efficiency, minimising waste and enhancing the reliability of the hydroponic actions.

# Appendix A – Useful references

Wokwi simulation - <https://wokwi.com/projects/433543810637673473>

Adafruit Dashboard - <https://io.adafruit.com/hcshah26/dashboards/hydroponic-project>

ESP32 S2 Datasheet - <https://www.espressif.com/sites/default/files/documentation/esp32-s2_datasheet_en.pdf>

ESP32 S3 Datasheet - <https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf>